Proceedings of Machine Learning Research vol 242:1552–1564, 2024

A Deep Learning Approach for Distributed Aggregative Optimization with Users' Feedback

Riccardo Brumali RICCARDO.BRUMALI@UNIBO.IT
Department of Electrical, Electronic and Information Engineering, Università di Bologna, Italy.
Guido Carnevale GUIDO.CARNEVALE@UNIBO.IT
Department of Electrical, Electronic and Information Engineering, Università di Bologna, Bologna, Italy.
Giuseppe Notarstefano GIUSEPPE.NOTARSTEFANO@UNIBO.IT
Department of Electrical, Electronic and Information Engineering, Università di Bologna, Bologna, Italy.

Abstract

We propose a novel distributed data-driven scheme for online aggregative optimization, i.e., the framework in which agents in a network aim to cooperatively minimize the sum of local time-varying costs, each depending on a local decision variable and an aggregation of all of them. We consider a "personalized" setup in which each cost exhibits a term capturing the user's dissatisfaction and, thus, is unknown. We enhance an existing distributed optimization scheme by endowing it with a learning mechanism based on neural networks that estimate the missing part of the gradient via users' feedback about the cost. Our algorithm combines two loops with different timescales devoted to performing optimization and learning steps. In turn, the proposed scheme also embeds a distributed consensus mechanism aimed at locally reconstructing the unavailable global information due to the presence of the aggregative variable. We prove an upper bound for the dynamic regret related to (i) the initial conditions, (ii) the temporal variations of the functions, and (iii) the learning errors about the unknown cost. Finally, we test our method via numerical simulations.

Keywords: Distributed Optimization, Online Optimization, Deep Learning, Neural Network

1. Introduction

Distributed architectures are becoming more and more popular to control networks of computing and communicating devices. For this reason, distributed optimization is gaining increasing attention, see, e.g., the surveys Notarstefano et al. (2019); Giselsson and Rantzer (2018); Nedić and Liu (2018); Yang et al. (2019) for a comprehensive overview of existing setups and solution strategies. Distributed architectures are intrinsically suited for the so-called personalized optimization frameworks, i.e., the recent branch of research considering human satisfaction in the optimization process. Essentially, personalized optimization problems focus on cost functions given by the sum of a known part, named engineering function and related to measurable quantities (e.g., time or energy), and an unknown part representing the user's (dis)satisfaction with the current solution. Since synthetic models involving human preferences often work only in an average sense without ensuring effectiveness on the specific preferences of single users, in personalized optimization the key idea is to adopt data-driven strategies using feedback from specific users about the current solution.

Given the evolving synergy between teams of robots and teams of humans across various fields, personalized optimization approaches could yield significant benefits in the control of networks of cooperative robots. Differently from the well-known consensus optimization problems, the so-called aggregative setup allows for formulating many tasks arising in the context of cooperative robotics Testa et al. (2023). This novel framework has been introduced in Li et al. (2021a) and considers a network of agents that aim to minimize the sum of local costs depending on both local and global information (the so-called aggregative variable). The dependence on an aggregative variable has also been investigated in the context of aggregative games, see, e.g., Belgioioso et al. (2020); Gadjov and Pavel (2020); Cenedese et al. (2020); Carnevale et al. (2024a), where, however, the agents cooperate to find a Nash equilibrium rather than an optimal solution. In Chen and Liang (2022), the scheme proposed in Li et al. (2021a) has been extended to deal with quantized communication among the agents. In Wang and Yi (2023), the problem is addressed via a projection-free distributed method based on the Franke-Wolfe update. Online and constrained versions of the problem are considered in Li et al. (2021b); Carnevale et al. (2022a). The aggregative setup has been also studied from the perspective of feedback optimization Carnevale et al. (2022b, 2024b). The work Pichierri et al. (2023) adapts the scheme in Carnevale et al. (2022a) for multi-robot cooperative surveillance, while Carnevale and Notarstefano (2022) enhances it with a Recursive Least Square (RLS) method to estimate the unknown cost via feedback from the users.

In the field of centralized optimization, a first step towards the usage of a user's feedback for the definition of human discomfort has been done by Luo et al. (2020), in which a trajectory design problem has been approached considering a cost function depending on human complaints. In Simonetto et al. (2021), personalized optimization is addressed by combining a learning mechanism based on Gaussian Processes (GP) with an optimization method. In Fabiani et al. (2022), the personalized framework is addressed in the context of game theory. As for personalized distributed frameworks, the articles Ospina et al. (2022) and Notarnicola et al. (2022) respectively use GP combined with a primal-dual method and RLS merged with the gradient tracking algorithm.

The unknown terms in this work are tackled through a learning strategy based on Artificial Neural Networks (ANN). In the context of optimization, neural networks have also been used in Cothren et al. (2022, 2023), where the authors embed them into feedback controllers to estimate the system state from perceptual information. In Pirrone et al. (2023), neural networks are combined with a zeroth-order scheme. The work Camisa and Notarstefano (2022) uses generative adversarial networks in the context optimization for smart grids. Finally, in Fabiani and Goulart (2022), a Model Predictive Controller is equipped with neural networks to reduce the required computational burden.

This paper addresses online aggregative optimization with objective functions inherited from personalized optimization, i.e., composed of both a known engineering term and an unknown user-related one. Notably, to the best of the authors' knowledge, this work is the first one addressing the personalized aggregative framework with completely time-varying costs. In this setup, we propose ANN-PROJECTED AGGREGATIVE TRACKING (ANN-PAT), namely a distributed strategy that enhances an already existing algorithm by introducing ANNs combined with an automatic differentiation procedure to learn the missing part of the cost function gradients. In detail, at each iteration, agents use their feedback data to train locally a network in a different timescale to improve the gradient approximations. These estimates feed the distributed scheme PROJECTED AGGREGATIVE TRACKING (PAT) Carnevale et al. (2022a), which, in turn, is the interconnection between a distributed approximated projected gradient method and a consensus scheme due to locally reconstructing the unavailable global information related to the aggregative variable. We theoretically provide a bound on the achieved dynamic regret that depends on (i) the initial conditions, (ii) the temporal variations of the functions, and (iii) the learning errors about the unknown parts.

The paper is organized as follows. Section 2 presents the problem setup. Section 3.1 introduces ANN-PAT that is analyzed in Section 4. Section 5 tests ANN-PAT via numerical simulations.

Notation: The vertical concatenation of the vectors $x_1, x_2, \dots, x_n$ is $\text{col}(x_1, x_2, \dots, x_N)$. We use $\text{blkdiag}(M_1, \dots, M_N)$ to denote the block diagonal matrix where the i-th diagonal block is given by $M_i \in \mathbb{R}^{n_i \times m_i}$. The symbol 1_N denotes the vector of N ones, while I_m is the identity matrix in $\mathbb{R}^{m \times m}$. The symbol $\otimes$ refers to the Kronecker product, while $\mathbf{1}_{N,d} := 1_N \otimes I_d$. The Euclidean projection of the vector $y \in \mathbb{R}^n$ on a closed convex set $X \subset \mathbb{R}^n$ is denoted by $P_X[y]$. Given $g : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}^m$, $\nabla_1 g(\cdot, \cdot)$ and $\nabla_2 g(\cdot, \cdot)$, represent the gradients with respect to the first and the second argument, respectively. Given $M \in \mathbb{R}^{n \times n}$, we denote as $\rho_{\max}(M)$ its spectral radius.

2. Problem Formulation

We consider N agents that, for each $t \geq 0$, aim to cooperatively solve online personalized aggregative optimization problems in the form

$$\min_{(x_1,\dots,x_N) \in X} \quad \sum_{i=1}^{N} \left(V_{i,t}\left(x_i, \sigma_t(x)\right) + U_{i,t}\left(x_i, \sigma_t(x)\right) \right), \tag{1}$$

where $x := \text{col}(x_1, \dots, x_N) \in \mathbb{R}^n$ is the global decision vector with each $x_i \in \mathbb{R}^{n_i}$ such that $n = \sum_{i=1}^N n_i$. We consider a constraint set $X \subseteq \mathbb{R}^n$ given by $X := X_1 \times \cdots \times X_N$ with each $X_i \subseteq \mathbb{R}^{n_i}$. We identify for each agent the engineering cost function $V_{i,t} : \mathbb{R}^{n_i} \times \mathbb{R}^{n_\sigma} \longrightarrow \mathbb{R}$ and the unknown user's dissatisfaction term $U_{i,t} : \mathbb{R}^{n_i} \times \mathbb{R}^{n_\sigma} \longrightarrow \mathbb{R}$. Their sum defines the local cost $f_{i,t}(x_i, \sigma_t) := V_{i,t}(x_i, \sigma_t) + U_{i,t}(x_i, \sigma_t)$. The aggregation term $\sigma_t(x)$ reads as

$$\sigma_t(x) := \tfrac{1}{N} \sum_{i=1}^{N} \phi_{i,t}(x_i),$$

where $\phi_{i,t} : \mathbb{R}^{n_i} \longrightarrow \mathbb{R}^{n_\sigma}$ denotes the contribution of agent i at time t, for all $i \in \{1, \dots, N\}$ and $t \geq 0$. Furthermore, we introduce the global cost $f_t(x, \sigma_t(x)) := \sum_{i=1}^N f_{i,t}(x_i, \sigma_t(x))$, $\phi_t(x) := \text{col}(\phi_{1,t}(x_1), \dots, \phi_{N,t}(x_N))$, and $\nabla\phi_t(x) := \text{blkdiag}(\nabla\phi_{1,t}(x_1), \dots, \nabla\phi_N(x_N))$. As customary in online optimization, the new information $V_{i,t}$, $\phi_{i,t}$, and user's feedback data are revealed only once the new estimate $x_{i,t}$ has been computed. This implies that, in general, the optimal solution $x_t^\star$ cannot be reached and, thus, given $T > 0$, the performance of our distributed algorithm will be evaluated via dynamic regret, i.e., the metric R_T defined as

$$R_T = \sum_{t=1}^{T} f_t(x_t, \sigma_t(x_t)) - f_t(x_t^\star, \sigma_t(x_t^\star)). \tag{2}$$

We address problem (1) according to the distributed paradigm and, thus, agent i can only access its private information $V_{i,t}$, X_i, $\phi_{i,t}$, the user's feedback, and data received by its neighboring agents.

Indeed, we use an undirected graph $\mathcal{G} = (\{1, \dots, N\}, \mathcal{E}, \mathcal{W})$, to model the communication among the agents, where $\{1, \dots, N\}$ represents the set of agents, $\mathcal{E} \in \{1, \dots, N\} \times \{1, \dots, N\}$ is the edge set, and $\mathcal{W} \in \mathbb{R}^{N \times N}$ is the weighted adjacency matrix. Agent i and agent j can exchange information if and only if $j \in \mathcal{N}_i$, where $\mathcal{N}_i := \{j \mid (j, i) \in \mathcal{E}\}$ is the set of neighbors of agent i. Given the (i, j)-entry $w_{i,j}$ of $\mathcal{W}$, it holds $w_{i,j} \geq 0$ if and only if $(i, j) \in \mathcal{E}$, otherwise $w_{i,j} = 0$.

3. ANN-PAT: Algorithm Description and Convergence Properties

In this section, we present ANN-PROJECTED AGGREGATIVE TRACKING (ANN-PAT), a novel distributed optimization algorithm to address personalized aggregative problems of the form (1). ANN-PAT consists of two main blocks: (i) the distributed optimization algorithm named PAT Carnevale et al. (2022a) and (ii) a deep-learning procedure aimed at reconstructing the gradients of the unknown parts $U_{i,t}(x_i, \sigma_t(x))$. These two parts are briefly introduced in Section 3.1 and 3.2. Then, in Section 3.3, the novel algorithm is introduced and its convergence properties are stated.

3.1. PROJECTED AGGREGATIVE TRACKING

When running PAT, agent i maintains an estimate $x_{i,t} \in \mathbb{R}^{n_i}$ about the i-th portion $x^\star_{i,t} \in \mathbb{R}^{n_i}$ of the optimal solution $x^\star_t := \text{col}(x^\star_{1,t}, \dots, x^\star_{N,t})$ of problem (1) and updates it with an approximated projected gradient step. Indeed, the derivative of $f_t(\cdot, \sigma_t(\cdot))$ with respect to x_i, would read as

$$\frac{\partial f_t(x, \sigma_t(x))}{\partial x_i} = \nabla_1 f_{i,t}(x_i, \sigma_t(x)) + \frac{1}{N} \nabla \phi_{i,t}(x_i) \sum_{j=1}^{N} \nabla_2 f_{i,t}(x_i, \sigma_t(x)),$$

where, however, the local knowledge of the global terms $\sigma_t(x)$ and $\sum_{j=1}^N \nabla_2 f_{j,t}(x_{j,t}, \sigma_t(x))$ would be required thus violating the desired distributed paradigm. To compensate for this lack of knowledge, agent i maintains $s_i, r_i \in \mathbb{R}^{n_\sigma}$ providing proxies about $\sigma_t(x)$ and $\sum_{j=1}^N \nabla_2 f_{j,t}(x_j, \sigma_t(x))$, respectively, and replaces $\dfrac{\partial f_t(x, \sigma_t(x))}{\partial x_i}$ with $d_{i,t} : \mathbb{R}^{n_i} \times \mathbb{R}^{n_\sigma} \times \mathbb{R}^{n_\sigma} \to \mathbb{R}^{n_i}$ given by

$$d_{i,t}(x_i, s_i, r_i) := \nabla_1 f_{i,t}(x_i, s_i) + \tfrac{1}{N} \nabla \phi_{i,t}(x_i) r_i. \tag{3}$$

The variables s_i and r_i are updated via dynamic consensus Kia et al. (2019), namely

$$\begin{aligned} s_{i,t+1} &= \sum_{j \in \mathcal{N}_i} w_{i,j} s_{j,t} + \phi_{i,t+1}(x_{i,t+1}) - \phi_{i,t}(x_{i,t}) \\ r_{i,t+1} &= \sum_{j \in \mathcal{N}_i} w_{i,j} r_{j,t} + \nabla_2 f_{i,t+1}(x_{i,t+1}, s_{i,t+1}) - \nabla_2 f_{i,t}(x_{i,t}, s_{i,t}), \end{aligned}$$

in which we recall that each $w_{i,j}$ represents the (i,j)-entry of the weighted adjacency matrix $\mathcal{W}$.

3.2. Learning mechanism based on Neural Networks and Automatic Differentiation

The local costs of problem (1) are partially unknown and, thus, the direction $d_{i,t}(x_i, s_i, r_i)$ (cf. (3)) cannot be directly computed. Thus, we introduce a learning mechanism in which each agent exploits its user's feedback data to train a feedforward neural network that approximates the unknown user dissatisfaction $U_{i,t}(x_i, s_i)$. In detail, for each $t \geq 0$, a set of $\mathcal{Q}$ samples from the user are available and can be used to perform $\tau > 0$ training steps for the network considering the regression problem

$$\min_{\theta_i \in \mathbb{R}^p} \frac{1}{\mathcal{Q}} \sum_{q=1}^{\mathcal{Q}} \left| y_i(\theta_i, x_i^q, s_i^q) - U_{i,t}(x_i^q, s_i^q) \right|^2, \tag{4}$$

where $\theta_i \in \mathbb{R}^p$ is the vector containing the network trainable parameters and $y_i : \mathbb{R}^p \times \mathbb{R}^{n_i} \times \mathbb{R}^{n_\sigma} \longrightarrow \mathbb{R}$ is the inference dynamics of the network i. More in detail, along a different timescale with iteration index $k \in \mathbb{N}$, agent i maintains an estimate $\theta_{i,t,k} \in \mathbb{R}^p$ of a solution of problem (4) that is iteratively improved along the training phase. Then, at the end of such a training phase, agent i retrieves the final estimate $\theta_{i,t} := \theta_{i,t,\tau}$ and uses it to approximate $U_{i,t}(x_i, s_i)$ via $\hat{U}_{i,t}(x_i, s_i)$ defined as

$$\hat{U}_{i,t}(x_i, s_i) := y_i(\theta_{i,t}, x_i, s_i).$$

The gradients $\nabla_1 U_{i,t}(x_i, s_i)$ and $\nabla_2 U_{i,t}(x_i, s_i)$ are estimated by means of an automatic differentiation procedure running in backward mode Baydin et al. (2018). This method generalizes the well-known backpropagation for evaluating derivatives of numeric functions expressed as computer programs. By denoting with $L \in \mathbb{N}$ the number of network layers and with $P^l \in \mathbb{N}$ the neurons number of the l-th layer, the network inference dynamics is given by

$$v_i^{l+1} = \Phi_i^l(v_i^l, \theta_i^l), \tag{5}$$

for all $l \in \{1, \ldots, L\}$, where, given the activation function $\varphi_i^{l,p} : \mathbb{R} \to \mathbb{R}$ of neuron p of layer l, $\Phi_i^l(\cdot) := \text{col}\left(\varphi_i^{l,1}(\cdot), \ldots, \varphi_i^{l,P^l}(\cdot)\right)$. Let $v_i^l := \text{col}\left(v_i^{l,1}, \ldots, v_i^{l,P^l}\right)$ and $\theta_i^l \in \mathbb{R}^{P^l \cdot P^{l+1}}$ denote the stack of the neurons output and the subset of all the network weights of layer l, respectively. Note that, v_i^l can be evaluated for all $l \in \{0, \ldots, L\}$ by executing the network dynamics. Furthermore, we can consider

$$\bar{v}_i^{l,p} := \frac{\partial y_i(\theta_i, x_i, s_i)}{\partial v_i^{l,p}},$$

which can be always expressed recursively as a function of both $\bar{v}_i^{l+1} := \text{col}(\bar{v}_i^{l+1,1}, \ldots, \bar{v}_i^{l+1,P^{l+1}})$ and v_i^l by exploiting the derivation chain rule. Since by definition

$$\bar{v}_i^{L,1} = \frac{\partial y_i(\theta_i, x_i, s_i)}{\partial y_i} = 1,$$

we can evaluate $\bar{v}_i^l$ for all $l \in \{0, 1, \ldots, L\}$ starting from the network output. Note that the entries of $\bar{v}_i^0$ are the network inputs, i.e. $\bar{v}_i^0 := \text{col}(x_i, s_i)$. Thus, we can write

$$\begin{aligned}
\nabla_1 \hat{U}_{i,t}(x_i, s_i) &= \nabla_2 y_i(\theta_{i,t}, x_i, s_i) = \begin{bmatrix} \bar{v}_i^{0,1} & \ldots & \bar{v}_i^{0,n_i} \end{bmatrix} \\
\nabla_2 \hat{U}_{i,t}(x_i, s_i) &= \nabla_3 y_i(\theta_{i,t}, x_i, s_i) = \begin{bmatrix} \bar{v}_i^{0,n_i+1} & \ldots & \bar{v}_i^{0,n_i+n_\sigma} \end{bmatrix},
\end{aligned}$$

where $\nabla_3 y_i(\theta_{i,t}, x_i, s_i)$ is the gradient of $y_i(\theta_{i,t}, x_i, s_i)$ with respect to s_i. We formalize this procedure in Algorithm 1, where "forwardPass" denotes an execution of the network dynamics to evaluate $v_i := \text{col}(v_i^1, \ldots, v_i^L)$. Note that, each $\varphi_i^{l,p}(\cdot)$ is known from the network structure definition.

Algorithm 1: Backward-pass for Neural Networks

```
**Input:** $\theta_i$, $x_i$, $s_i$
$v_i = \text{forwardPass}(\theta_i, x_i, s_i)$
**for** $l = L-1, L-2, \ldots, 0$ **do**
    **for** $p = 1, 2, \ldots, P^l$ **do**
        $\bar{v}_i^{l,p} = \sum_{k=1}^{P^{l+1}} \bar{v}_i^{l+1,k} \frac{\partial \varphi_i^{l+1,k}(v_i^l)}{\partial v_i^{l,p}}$
    **end**
**end**
```

The next assumption states the learning capability of the mechanism considered in this work.

Assumption 1 (Neural Network) *For each time instant $t > 0$, consider the estimate $\hat{U}_{i,t}(x_i, s_i)$ given from the network of agent i. Then, there exist $b_{1,t}, b_{2,t} > 0$ such that*

$$\left\| \nabla_1 \hat{U}_{i,t}(x_i, s_i) - \nabla_1 U_{i,t}(x_i, s_i) \right\| \leq b_{1,t} \ , \qquad \left\| \nabla_2 \hat{U}_{i,t}(x_i, s_i) - \nabla_2 U_{i,t}(x_i, s_i) \right\| \leq b_{2,t},$$

for all $x_i \in X_i$, $s_i \in \mathbb{R}^{n_\sigma}$, and $i \in \{1, \dots, N\}$.

3.3. ANN-PROJECTED AGGREGATIVE TRACKING

The ANN-PAT scheme merges PAT and the described learning procedure. In particular, agent i approximates $\nabla_1 f_{i,t}(x_i, s_i)$ and $\nabla_2 f_{i,t}(x_i, s_i)$ via $\nabla_1 \hat{f}_{i,t}(x_i, s_i)$ and $\nabla_2 \hat{f}_{i,t}(x_i, s_i)$ defined as

$$\nabla_1 \hat{f}_{i,t}(x_i, s_i) = \nabla_1 V_{i,t}(x_i, s_i) + \nabla_1 \hat{U}_{i,t}(x_i, s_i), \quad \nabla_2 \hat{f}_{i,t}(x_i, s_i) = \nabla_2 V_{i,t}(x_i, s_i) + \nabla_2 \hat{U}_{i,t}(x_i, s_i).$$

A pseudo-code of ANN-PAT is reported in Algorithm 2 from the perspective of agent i. The parameters $\alpha > 0$ and $\delta \in (0, 1)$ are respectively the stepsize and the convex combination constant. We indicate with the function "TrainSteps" τ iteration of any optimization method (e.g. Adam) for the solution of Problem (4) to update the vector of network weights and with "GradLearning" the automatic differentiation procedure described in Algorithm 1. Moreover, $c_{i,t}^q \in \mathbb{R}^{n_i n_\sigma}$ represents a random sample picked in the interval $[c_{\text{min}}, c_{\text{max}}]$ considering a uniform probability distribution. It is worth mentioning that Algorithm 2 works on two independent timescales, one for the optimization steps and one for the network training. In addition, we recall that the learning mechanism is implemented locally by each agent, resulting in a fully distributed algorithm suitable for large-scale systems.

Algorithm 2: ANN-PROJECTED AGGREGATIVE TRACKING (for agent i)

```
Initialization:
$x_{i,0} \in X_i$, $s_{i,0} = \phi_{i,0}(x_{i,0})$, $r_{i,0} = \nabla_2 V_{i,0}(x_{i,0}, s_{i,0})$, $\theta_{i,0} \in \mathbb{R}^p$
for $t = 0, 1, 2, \dots$ do
    Optimization:
    $\tilde{x}_{i,t} = P_{X_i}[x_{i,t} - \alpha(\nabla_1 \hat{f}_{i,t}(x_{i,t}, s_{i,t}) + \nabla \phi_{i,t}(x_{i,t}) r_{i,t})]$
    $x_{i,t+1} = x_{i,t} + \delta(\tilde{x}_{i,t} - x_{i,t})$
    $s_{i,t+1} = \sum_{j \in \mathcal{N}_i} w_{i,j} s_{j,t} + \phi_{i,t+1}(x_{i,t+1}) - \phi_{i,t}(x_{i,t})$
    Measure:
    for $q = 1, \dots, \mathcal{Q}$ do
        $u_{i,t+1}^q = \text{col}(x_{i,t+1}, s_{i,t+1}) + c_{i,t+1}^q$
        $m_{i,t+1}^q = U_{i,t+1}(u_{i,t+1}^q)$
    end
    Learning:
    $\theta_{i,t+1} = \text{TrainSteps}(m_{i,t+1}^1, \dots, m_{i,t+1}^{\mathcal{Q}}, u_{i,t+1}^1, \dots, u_{i,t+1}^{\mathcal{Q}}, \tau)$
    $\left(\nabla_1 \hat{U}_{i,t+1}(x_{i,t+1}, s_{i,t+1}), \nabla_2 \hat{U}_{i,t+1}(x_{i,t+1}, s_{i,t+1})\right) = \text{GradLearning}(\theta_{i,t+1}, x_{i,t+1}, s_{i,t+1})$
    $r_{i,t+1} = \sum_{j \in \mathcal{N}_i} w_{i,j} r_{j,t} + \nabla_2 \hat{f}_{i,t+1}(x_{i,t+1}, s_{i,t+1}) - \nabla_2 \hat{f}_{i,t}(x_{i,t}, s_{i,t})$
end
```

To present the convergence properties of Algorithm 2, we collect all the local updates (omitting the learning steps) obtaining the following stacked vector formulation

$$x_{t+1} = x_t + \delta(P_X[x_t - \alpha(\hat{G}_{1,t}(x_t, s_t) + \nabla\phi_t(x_t) r_t)] - x_t) \tag{6a}$$
$$s_{t+1} = W s_t + \phi_{t+1}(x_{t+1}) - \phi_t(x_t) \tag{6b}$$
$$r_{t+1} = W r_t + \nabla_2 \hat{G}_{2,t+1}(x_{t+1}, s_{t+1}) - \nabla_2 \hat{G}_{2,t}(x_t, s_t), \tag{6c}$$

where we consider $W = \mathcal{W} \otimes I_N$, $\hat{G}_{1,t}(x_t, s_t) := \text{col}(\nabla_1 \hat{f}_{1,t}(x_{1,t}, s_{1,t}), \dots, \nabla_1 \hat{f}_{N,t}(x_{N,t}, s_{N,t}))$ and $\hat{G}_{2,t}(x_t, s_t) := \text{col}(\nabla_2 \hat{f}_{1,t}(x_{1,t}, s_{1,t}), \dots, \nabla_2 \hat{f}_{N,t}(x_{N,t}, s_{N,t}))$. With analogous notation, we also introduce $G_{1,t}(x_t, s_t) := \text{col}(\nabla_1 f_{1,t}(x_{1,t}, s_{1,t}), \dots, \nabla_1 f_{N,t}(x_{N,t}, s_{N,t}))$ and $G_{2,t}(x_t, s_t) := \text{col}(\nabla_2 f_{1,t}(x_{1,t}, s_{1,t}), \dots, \nabla_2 f_{N,t}(x_{N,t}, s_{N,t}))$. In the next, we will provide a bound for the dynamic regret of ANN-PAT. To this end, we introduce the error vector $z_t \in \mathbb{R}^3$ defined as

$$z_t := \begin{bmatrix} \|x_t - x_t^\star\| & \|s_t - \mathbf{1}\bar{s}_t\| & \|r_t - \mathbf{1}\bar{r}_t\| \end{bmatrix}^\top,$$

with $\bar{s}_t := \frac{1}{N}\sum_{i=1}^N s_{i,t}$ and $\bar{r}_t := \frac{1}{N}\sum_{i=1}^N r_{i,t}$. Then, we consider

$$\eta_t := \sup_{x \in R^n, s \in \mathbb{R}^{Nn_\sigma}} \|G_{2,t+1}(x,s) - G_{2,t}(x,s)\|, \qquad \omega_t := \sup_{x \in R^n} \|\phi_{t+1}(x) - \phi_t(x)\|$$

$$\zeta_t := \left\|x_{t+1}^\star - x_t^\star\right\|, \quad \nu_t := \text{col}(\zeta_t, \eta_t + \omega_t, \eta_t + L_2\omega_t), \quad H_t := \text{col}(\delta\alpha b_{1,t}, 0, (1+\sqrt{N}) b_{2,t+1} + b_{2,t}).$$

Now, we state the class of problems and graphs needed to provide our result.

Assumption 2 (Global Problem Convexity) *The set X_i is closed, convex, and nonempty for all $i \in \{1, \dots, N\}$, while the function $f_t(x, \sigma_t(x))$ is μ-strongly convex for all $t \geq 0$, with $\mu > 0$.*

Assumption 3 (Function Regularity) *For all $t \geq 0$, each function $f_{i,t}(x, \sigma_t(x))$ is differentiable and $f_t(x, \sigma_t(x))$ has L_1-Lipschitz continuous gradient, i.e.,*

$$\left\|\nabla f_t(x, \sigma_t(x)) - \nabla f_t(x', \sigma_t(x'))\right\| \leq L_1 \left\|x - x'\right\|, \qquad \forall x, x' \in \mathbb{R}^n.$$

Further, consider $G_t(x, s) := \nabla_1 f_t(x, s) + \nabla\phi(x)\mathbf{1}_N \otimes \frac{1}{N}\sum_{i=1}^N \nabla_2 f_{i,t}(x_i, s_i)$. Then, for all $t \geq 0$, $G_t(x, s)$ and $\nabla_2 f_t(x, s)$ are Lipschitz continuous, respectively with constant $L_1, L_2 > 0$. Moreover, for all $i \in \{1, \dots, N\}$ and $t \geq 0$, the function $\phi_{i,t}(x_i)$ is differentiable and L_3-Lipschitz continuous. Finally, assume both η_t and ω_t finite for all $t \geq 0$.

Assumption 4 (Communication Graph) *$\mathcal{G}$ is connected and $\mathcal{W}$ is doubly stochastic.*

With these assumptions at hand, we are ready to provide the main result of the paper.

Theorem 1 *Consider* ANN-PAT *as reported in Algorithm 2. Let Assumptions 1, 2, 3, and 4 hold. Then, there exist $\lambda, \bar{\delta} > 0$, and $\tilde{\rho} \in (0, 1)$, such that for all $T \geq 1$ it holds*

$$R_T \leq \frac{L_1\lambda^2}{2}\left(\frac{\|z_0\|^2}{1 - \tilde{\rho}^2} + 2\|z_0\| U_T + Q_T\right),$$

for any $\alpha \in \left(0, \frac{1}{L_1}\right)$ and $\delta \in \left(0, \bar{\delta}\right)$, where U_T and Q_T are given by

$$U_T := \sum_{t=1}^T \sum_{k=0}^{t-1} \tilde{\rho}^{t+1} \|\nu_{t-k-1} + H_{t-k-1}\|, \quad Q_T := \sum_{t=1}^T \left(\sum_{k=0}^{t-1} \tilde{\rho}^k \|\nu_{t-k-1} + H_{t-k-1}\|\right)^2. \tag{7}$$

The proof of Theorem 1 is provided in Section 4. The bound provided by Theorem 1 depends on (i) the initial conditions, (ii) the temporal variations of the functions, and (iii) the learning errors about the unknown part of the cost. As it will become clearer in the proof, Theorem 1 is stated by interpreting ANN-PAT as a perturbed version of PAT. More in detail, the perturbation is due to the learning error about the unknown terms $\nabla_1 U_{i,t}$ and $\nabla_2 U_{i,t}$. Indeed, in the case in which the learning process is perfect and/or the unknown terms are not present, Theorem 1 recovers the results provided by (Carnevale et al., 2022a, Th. 1), i.e., the dynamic regret bound achieved by PAT.

4. Convergence Analysis

To prove Theorem 1, the idea is to consider Algorithm 2 as a perturbed version of PAT. To shorten the notation, we introduce the quantities: $d_t(x,s,r) := G_{1,t}(x,s) + \nabla\phi_t(x)r$, $\hat{d}_t(x,s,r) := \hat{G}_{1,t}(x,s) + \nabla\phi_t(x)r$ and $e_{y,t}(x,s) := \hat{G}_{2,t}(x,s) - G_{2,t}(x,s)$. Let us introduce $n_{x,t} : \mathbb{R}^n \times \mathbb{R}^{N\cdot n_\sigma} \times \mathbb{R}^{N\cdot n_\sigma} \to \mathbb{R}^n$ and $n_{r,t} : \mathbb{R}^n \times \mathbb{R}^{N\cdot n_\sigma} \times \mathbb{R}^{N\cdot n_\sigma} \to \mathbb{R}^{N\cdot n_\sigma}$ defined as

$$\begin{aligned} n_{x,t}(x,s,r) &:= x + \delta\left(P_X\left[x - \alpha\left(G_{1,t}(x,s) + \nabla\phi_t(x)r\right)\right] - x\right) \\ n_{r,t}(x,s,r) &:= Wr + G_{2,t+1}\left(n_{x,t}(x,s,r), Ws + \phi_{t+1}(n_{x,t}(x,s,r)) - \phi_t(x)\right) - G_{2,t}(x,s). \end{aligned}$$

These functions describe the dynamics of x_t and r_t when both $G_{1,t}(x,s)$ and $G_{2,t}(x,s)$ are completely known and, thus, coincide with the update laws of PAT Carnevale et al. (2022a). Hence, by adding and subtracting the terms (i) $\delta(P_X[x_t - \alpha d_t(x_t,s_t,r_t)])$ and (ii) $G_{2,t+1}(x_{t+1},s_{t+1}) - G_{2,t}(x_t,s_t)$ to the right-hand side of (6a) and (6c), respectively, we rewrite them as

$$x_{t+1} = n_{x,t}(x_t,s_t,r_t) + \delta(P_X[x_t - \alpha\hat{d}_t(x_t,s_t,r_t)] - P_X[x_t - \alpha d_t(x_t,s_t,r_t)]) \tag{8a}$$

$$r_{t+1} = n_{r,t}(x_t,s_t,r_t) + e_{y,t+1}(x_{t+1},s_{t+1}) - e_{y,t}(x_t,s_t), \tag{8b}$$

These equations allow us to interpret (6) as a perturbed version of a nominal system coinciding with PAT. Let $n_{\bar{r},t}(x,s) := \dfrac{1}{N}\sum_{i=1}^N \nabla_2 f_{i,t}(x_i,s_i)$ and $\bar{e}_{y,t}(x,s) := \dfrac{1}{N}\sum_{i=1}^N e^i_{y,t}(x_i,s_i)$, with $e^i_{y,t}(x_i,s_i) := \dfrac{1}{N}\sum_{i=1}^N(\nabla_2\hat{f}_{i,t}(x_i,s_i) - \nabla_2 f_{i,t}(x_i,s_i))$. Then, it holds

$$\bar{r}_{t+1} = n_{\bar{r},t+1}(x_{t+1},s_{t+1}) + \bar{e}_{y,t+1}(x_{t+1},s_{t+1}). \tag{9}$$

Preparatory Lemmas

At this point, we can introduce three preparatory Lemmas, necessary for the proof of Theorem 1.

Lemma 1 *Let Assumptions 1, 2, 3, and 4 hold. If $\alpha \leq 1/L_1$, then for all $t \leq 0$*

$$\left\|x_{t+1} - x^\star_{t+1}\right\| \leq (1-\delta\mu\alpha)\left\|x_t - x^\star_t\right\| + \delta\alpha L_1\left\|s_t - \mathbf{1}\bar{s}_t\right\| + \delta\alpha L_3\left\|r_t - \mathbf{1}\bar{y}_t\right\| + \zeta_t\delta\alpha b_{1,t}.$$

Lemma 2 *[(Carnevale et al., 2022a, Lemma 3)] Let Assumptions 1, 2, 3, and 4 hold. Then*

$$\begin{aligned} \left\|s_{t+1} - \mathbf{1}\bar{s}_{t+1}\right\| \leq{}& (\Lambda + \delta\alpha L_1 L_3)\left\|s_t - \mathbf{1}\bar{s}_t\right\| + \delta(2L_3 + \alpha L_1 L_3 + \alpha L_1 L_3^2)\left\|x_t - x^\star_t\right\| \\ &+ \delta\alpha L_3^2\left\|r_t - \mathbf{1}\bar{r}_t\right\| + \omega_t, \end{aligned}$$

for all $t \leq 0$, where Λ is the maximum eigenvalue of the matrix $W - \mathbf{1}\mathbf{1}^\top/N$.

Lemma 3 *Let Assumptions 1, 2, 3, and 4 hold. Then for all $t \leq 0$*

$$\begin{aligned}\|r_{t+1} - \mathbf{1}\bar{r}_{t+1}\| \leq (\Lambda + \delta\alpha L_3(L_2 + L_2 L_3)) \|r_t - \bar{r}_t\| + \delta(2 + \alpha L_1 + \alpha L_1 L_3)(L_2 + L_2 L_3) \|x_t - x_t^\star\| \\ + (\delta\alpha L_1(L_2 + L_2 L_3) + 2L_2) \|s_t - \mathbf{1}\bar{s}_t\| + L_2\omega_t + \eta_t + (1 + \sqrt{N})b_{2,t+1} + b_{2,t},\end{aligned}$$

where Λ is the maximum eigenvalue of the matrix $W - \mathbf{1}\mathbf{1}^\top/N$.

The proofs for Lemma 1 and Lemma 3 are not included as they are straightforward extentions of (Carnevale et al., 2022a, Lemma 1, Lemma 4).

Proof of Theorem 1

Now, we prove Theorem 1 using Lemma 1, 2, and 3. These results allow us to write

$$z_{t+1} \leq M(\delta)z_t + \nu_t + H_t,$$

with $M(\delta) := M_0 + \delta E$, where the matrices $M_0, E \in \mathbb{R}^{3\times 3}$ read as

$$M_0 := \begin{bmatrix} 1 & 0 & 0 \\ 0 & \Lambda & 0 \\ 0 & 2L_2 & \Lambda \end{bmatrix}, \quad E := \begin{bmatrix} -\mu\alpha & \alpha L_1 & \alpha L_3 \\ E_{21} & \alpha L_1 L_3 & \alpha L_3^2 \\ E_{31} & E_{32} & E_{33} \end{bmatrix},$$

in which $E_{21} := 2L_3 + \alpha L_1 L_3 + \alpha L_1 L_3^2$, $E_{31} := (2 + \alpha L_1 + \alpha L_1 L_3)(L_2 + L_2 L_3)$, $E_{32} := \alpha L_1(L_2 + L_2 L_3)$ and $E_{33} := \alpha L_3(L_2 + L_2 L_3)$. Since z_t, $M(\delta)$, ν_t, and H_t contain all positive quantities, we exploit the Lagrange formula to write

$$z_t \leq M^t z_0 + \sum_{k=0}^{t-1} M^k(\nu_{t-k-1} + H_{t-k-1}). \tag{10}$$

From (Carnevale et al., 2022a, Appendix E), there exists $\bar{\delta} > 0$ such that $\rho_{\max}(M(\delta)) < 1$ for all $\delta \in (0, \bar{\delta})$. Let us arbitrarily choose $\tilde{\rho} \in (\rho_{\max}(M(\delta)), 1)$ and $\gamma = \rho_{\max}(M(\delta)) - \tilde{\rho}$. Then, we apply (Horn and Johnson, 2012, Lemma 5.7.13) to guarantee the existence of a matrix norm $\|\cdot\|_\gamma$ such that $\|M\|_\gamma \leq \tilde{\rho} < 1$. Furthermore, by applying (Horn and Johnson, 2012, Theorem 5.7.13), there exists a vector norm $\|\cdot\|_\gamma$ compatible with the previous matrix norm, such that $\|Mv\|_\gamma \leq \|M\|_\gamma \|v\|_\gamma$ for all $M \in \mathbb{R}^{3\times 3}$ and $v \in \mathbb{R}^3$. Because of the equivalence of all norms on finite-dimensional vector spaces, there exist always $\lambda_1, \lambda_2 > 0$ such that $\|\cdot\| \leq \lambda_1 \|\cdot\|_\gamma$ and $\|\cdot\|_\gamma \leq \lambda_2 \|\cdot\|$. Thus, by applying the norm $\|\cdot\|_\gamma$ on both sides of (10), we get

$$\begin{aligned}\|z_t\| \leq \lambda_1 \|z_t\|_\gamma &\leq \lambda_1 \left\| M^t z_0 + \sum_{k=0}^{t-1} M^k(\nu_{t-k-1} + H_{t-k-1}) \right\|_\gamma \\ &\overset{(a)}{\leq} \tilde{\rho}^t \lambda_1 \|z_0\|_\gamma + \lambda_1 \sum_{k=0}^{t-1} \tilde{\rho}^k \|\nu_{t-k-1} + H_{t-k-1}\|_\gamma,\end{aligned} \tag{11}$$

where in (a) we have combined the triangular inequality, the Cauchy-Schwarz inequality, and the definition of $\tilde{\rho}$. From Assumption 3, $\nabla f_t(x_t, s_t)$ is L_1-Lipschitz continuous. Then, it holds

$$f_t(x_t, \sigma_t) - f_t(x_t^\star, \sigma_t(x_t^\star)) \leq \frac{L_1}{2} \|x_t - x_t^\star\|^2 \overset{(a)}{\leq} \frac{L_1}{2} \|z_t\|^2,$$

$$\overset{(b)}{\leq} \frac{L_1\lambda_1^2}{2}\left(\tilde{\rho}^{2t}\left\|z_0\right\|_\gamma^2+2\tilde{\rho}^t\left\|z_0\right\|_\gamma\sum_{k=0}^{t-1}\tilde{\rho}^k\left\|\nu_{t-k-1}+H_{t-k-1}\right\|_\gamma+\left(\sum_{k=0}^{t-1}\tilde{\rho}^k\left\|\nu_{t-k-1}+H_{t-k-1}\right\|_\gamma\right)^2\right)$$

$$\overset{(c)}{\leq} \frac{L_1\lambda^2}{2}\left(\tilde{\rho}^{2t}\left\|z_0\right\|^2+2\tilde{\rho}^t\left\|z_0\right\|\sum_{k=0}^{t-1}\tilde{\rho}^k\left\|\nu_{t-k-1}+H_{t-k-1}\right\|+\left(\sum_{k=0}^{t-1}\tilde{\rho}^k\left\|\nu_{t-k-1}+H_{t-k-1}\right\|\right)^2\right),$$

where (a) uses the fact that $\|x_t - x_t^\star\|$ is a component of z_t, (b) uses (11), and (c) sets $\lambda = \lambda_1\lambda_2$. The proof follows from this inequality by using the fact that $\tilde{\rho} \in (0,1)$, the geometric series property, and the definitions of R_T, U_T, and Q_T.

5. Numerical Simulations

Consider $N = 10$ agents communicating according to an undirected, connected Erdős-Rényi graph with connectivity parameter 0.5. We fix $n_i = 2\ \forall i \in \{1,\dots,N\}$, $n_\sigma = 2$, and consider the costs

$$V_{i,t}(x_i,\sigma_t(x))=\frac{1}{2}\left\|x_i-p_{i,t}\right\|^2+\frac{\beta}{2}\left\|x_i-\sigma_t(x)\right\|^2,\ U_{i,t}(x_i,\sigma_t(x))=\frac{1}{2}\left\|x_i-u_{i,t}^x\right\|^2+\frac{1}{2}\|\sigma_t(x)-u_{i,t}^{\sigma_t}\|^2,$$

where $\beta = 0.4$, $p_{i,t} := p_{i,c} + \mathrm{col}(\cos(t/100), \sin(t/100))$ with $p_{i,c} \in [0,100] \times [0,100]$, $u_{i,t}^x := u_{i,c}^x+\mathrm{col}(\cos(t/100), \sin(t/100))$, and $u_{i,t}^{\sigma_t} := u_{i,c}^{\sigma_t}+\mathrm{col}(\cos(t/100), \sin(t/100))$. We select the aggregative variable as $\sigma_t(x) = \sum_{i=1}^N l_{i,t}x_i/N$ with $l_{i,t} > 0$ such that $l_{i,t} := l_{i,t-1} + 0.01\sin(t/100)\cdot(-1)^i$ for all $i \in \{1,\dots,N\}$ and $t \geq 0$. We choose $u_{i,c}^x$, $u_{i,c}^{\sigma_t}$, $p_{i,c}$, $l_{i,0}$, and $x_{i,0}$ randomly, considering a uniform probability distribution. As for the feasible sets, we consider $X_i := [0,100]\times[0,100]$ for all $i \in \{1,\dots,N\}$. Further, each agent uses a network with $L = 12$ and $P^l = 20$, the well-known Rectified Linear Unit activation function for all neurons, $\tau = 500$, $\mathcal{Q} = 100$, $c_{\min} = -100$, and $c_{\max} = 100$. As for the algorithm parameters, we set $\delta = 0.1$ and $\alpha = 0.1$.

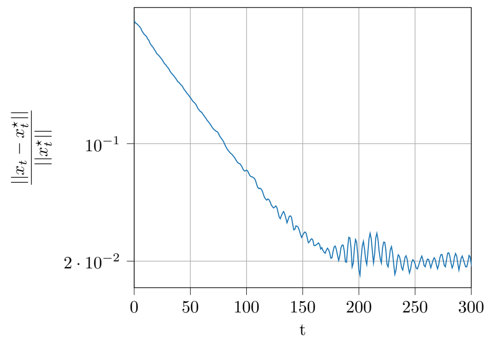

Figure 1: Relative solution error

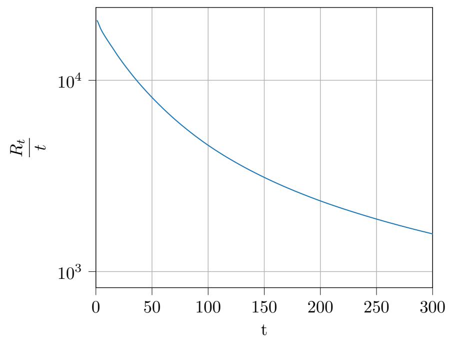

Figure 2: Average dynamic regret

Fig. 1 and 2 show the evolution of the relative error $\|x_t - x_t^\star\| \,/\, \|x_t^\star\|$ and the average dynamic regret R_T/T, respectively. Future works will provide simulations on more realistic datasets.

6. Conclusions

This paper proposed ANN-PAT, a novel distributed data-driven algorithm for personalized online aggregative optimization. This new scheme has been designed by interlacing an existing distributed optimization method with a deep-learning mechanism aimed at reconstructing the unknown terms due to the considered personalized setup. We analyzed the proposed distributed algorithm by providing an upper bound on the dynamic regret. Finally, we tested ANN-PAT via numerical simulations.

Acknowledgments

This work was supported in part by the Italian Ministry of Foreign Affairs and International Cooperation, grant number BR22GR01.

References

Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *Journal of Marchine Learning Research*, 18:1–43, 2018.

Giuseppe Belgioioso, Angelia Nedić, and Sergio Grammatico. Distributed generalized nash equilibrium seeking in aggregative games on time-varying networks. *IEEE Transactions on Automatic Control*, 66(5):2061–2075, 2020.

Andrea Camisa and Giuseppe Notarstefano. A distributed mixed-integer framework to stochastic optimal microgrid control. *IEEE Transactions on Control Systems Technology*, 31(1):208–220, 2022.

Guido Carnevale and Giuseppe Notarstefano. A learning-based distributed algorithm for personalized aggregative optimization. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 1576–1581. IEEE, 2022.

Guido Carnevale, Andrea Camisa, and Giuseppe Notarstefano. Distributed online aggregative optimization for dynamic multi-robot coordination. *IEEE Transactions on Automatic Control*, 2022a.

Guido Carnevale, Nicola Mimmo, and Giuseppe Notarstefano. Aggregative feedback optimization for distributed cooperative robotics. *IFAC-PapersOnLine*, 55(13):7–12, 2022b.

Guido Carnevale, Filippo Fabiani, Filiberto Fele, Kostas Margellos, and Giuseppe Notarstefano. Tracking-based distributed equilibrium seeking for aggregative games. *IEEE Transactions on Automatic Control*, 2024a.

Guido Carnevale, Nicola Mimmo, and Giuseppe Notarstefano. Nonconvex distributed feedback optimization for aggregative cooperative robotics. *Automatica*, 2024b. *To appear*.

Carlo Cenedese, Giuseppe Belgioioso, Yu Kawano, Sergio Grammatico, and Ming Cao. Asynchronous and time-varying proximal type dynamics in multiagent network games. *IEEE Transactions on Automatic Control*, 66(6):2861–2867, 2020.

Ziqin Chen and Shu Liang. Distributed aggregative optimization with quantized communication. *Kybernetika*, 58(1):123–144, 2022.

Liliaokeawawa Cothren, Gianluca Bianchin, and Emiliano Dall'Anese. Online optimization of dynamical systems with deep learning perception. *IEEE Open Journal of Control Systems*, 1: 306–321, 2022.

Liliaokeawawa Cothren, Gianluca Bianchin, Sarah Dean, and Emiliano Dall'Anese. Perception-based sampled-data optimization of dynamical systems. *IFAC-PapersOnLine*, 56(2):5083–5088, 2023.

Filippo Fabiani and Paul J Goulart. Reliably-stabilizing piecewise-affine neural network controllers. *IEEE Transactions on Automatic Control*, 2022.

Filippo Fabiani, Andrea Simonetto, and Paul J Goulart. Learning equilibria with personalized incentives in a class of nonmonotone games. In *2022 European Control Conference (ECC)*, pages 2179–2184. IEEE, 2022.

Dian Gadjov and Lacra Pavel. Single-timescale distributed gne seeking for aggregative games over networks via forward–backward operator splitting. *IEEE Transactions on Automatic Control*, 66 (7):3259–3266, 2020.

Pontus Giselsson and Anders Rantzer. *Large-scale and distributed optimization*, volume 2227. Springer, 2018.

Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

Solmaz S Kia, Bryan Van Scoy, Jorge Cortes, Randy A Freeman, Kevin M Lynch, and Sonia Martinez. Tutorial on dynamic average consensus: The problem, its applications, and the algorithms. *IEEE Control Systems Magazine*, 39(3):40–72, 2019.

Xiuxian Li, Lihua Xie, and Yiguang Hong. Distributed aggregative optimization over multi-agent networks. *IEEE Transactions on Automatic Control*, 67(6):3165–3171, 2021a.

Xiuxian Li, Xinlei Yi, and Lihua Xie. Distributed online convex optimization with an aggregative variable. *IEEE Transactions on Control of Network Systems*, 9(1):438–449, 2021b.

Xusheng Luo, Yan Zhang, and Michael M Zavlanos. Socially-aware robot planning via bandit human feedback. In *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS)*, pages 216–225. IEEE, 2020.

Angelia Nedić and Ji Liu. Distributed optimization for control. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:77–103, 2018.

Ivano Notarnicola, Andrea Simonetto, Francesco Farina, and Giuseppe Notarstefano. Distributed personalized gradient tracking with convex parametric models. *IEEE Transactions on Automatic Control*, 68(1):588–595, 2022.

Giuseppe Notarstefano, Ivano Notarnicola, and Andrea Camisa. Distributed optimization for smart cyber-physical networks. *Foundations and Trends® in Systems and Control*, 7(3):253–383, 2019.

Ana M Ospina, Andrea Simonetto, and Emiliano Dall'Anese. Time-varying optimization of networked systems with human preferences. *IEEE Transactions on Control of Network Systems*, 10 (1):503–515, 2022.

Lorenzo Pichierri, Guido Carnevale, Lorenzo Sforni, Andrea Testa, and Giuseppe Notarstefano. A distributed online optimization strategy for cooperative robotic surveillance. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5537–5543. IEEE, 2023.

Michelle Pirrone, Emiliano Dall'Anese, and Taylor W Barton. Data-driven optimization strategies for tunable rf systems. *IEEE Transactions on Microwave Theory and Techniques*, 2023.

Andrea Simonetto, Emiliano Dall'Anese, Julien Monteil, and Andrey Bernstein. Personalized optimization with user's feedback. *Automatica*, 131:109767, 2021.

Andrea Testa, Guido Carnevale, and Giuseppe Notarstefano. A tutorial on distributed optimization for cooperative robotics: from setups and algorithms to toolboxes and research directions. *arXiv preprint arXiv:2309.04257*, 2023.

Tongyu Wang and Peng Yi. Distributed projection-free algorithm for constrained aggregative optimization. *International Journal of Robust and Nonlinear Control*, 2023.

Tao Yang, Xinlei Yi, Junfeng Wu, Ye Yuan, Di Wu, Ziyang Meng, Yiguang Hong, Hong Wang, Zongli Lin, and Karl H Johansson. A survey of distributed optimization. *Annual Reviews in Control*, 47:278–305, 2019.