

# A Decentralized Multi-Objective Optimization Algorithm

Maude J. Blondin · Matthew Hale

Received: date / Accepted: date

**Abstract** During the past two decades, multi-agent optimization problems have drawn increased attention from the research community. When multiple objective functions are present among agents, many works optimize the sum of these objective functions. However, this formulation implies a decision regarding the relative importance of each objective function. In fact, optimizing the sum is a special case of a multi-objective problem in which all objectives are prioritized equally. In this paper, a distributed optimization algorithm that explores Pareto optimal solutions for non-homogeneously weighted sums

---

Maude J. Blondin, Corresponding author

Université de Sherbrooke

Sherbrooke, Canada

maude.josee.blondin@usherbrooke.ca

Matthew Hale

University of Florida

Gainesville, Florida

matthewhale@ufl.edu

of objective functions is proposed. This exploration is performed through a new rule based on agents' priorities that generates edge weights in agents' communication graph. These weights determine how agents update their decision variables with information received from other agents in the network. Agents initially disagree on the priorities of the objective functions though they are driven to agree upon them as they optimize. As a result, agents still reach a common solution. The network-level weight matrix is (non-doubly) stochastic, which contrasts with many works on the subject in which it is doubly-stochastic. New theoretical analyses are therefore developed to ensure convergence of the proposed algorithm. This paper provides a gradient-based optimization algorithm, proof of convergence to solutions, and convergence rates of the proposed algorithm. It is shown that agents' initial priorities influence the convergence rate of the proposed algorithm and that these initial choices affect its long-run behavior. Numerical results performed with different numbers of agents illustrate the performance and efficiency of the proposed algorithm.

**Keywords** Multi-agent systems · Distributed Optimization · Pareto Front · Multi-objective Optimization

## 1 Introduction

Over the last two decades, multi-agent systems have attracted significant interest [1]–[3]. In particular, the study of the consensus problem, where agents have to agree on a common value, has been motivated by emerging applica-

tions such as formation control [4]. The consensus problem has been extended to multi-agent optimization, i.e., agents collectively work towards minimizing a sum of objective functions by minimizing a local objective and repeatedly averaging their iterates to reach agreement on a final answer. One common approach in problems with many objectives is optimizing their sum with each agent independently optimizing only one of the objective functions [6]-[19]. However, optimizing the sum carries an implicit decision about the problem formulation, namely that all the objectives have the same priority and that all agents agree on these priorities.

Equal prioritization among functions represents a special case of a multi-objective problem, and applications in which objectives may have different importance are easy to envision. For instance, in a fleet of self-driving cars, agents may have different priorities in trajectory planning such as minimizing fuel usage vs. travel time, or in a collection of smart buildings, agents may have different preferences regarding the management of their energy [20].

A large body of work on multi-objective optimization to solve problems of this kind has emerged for centralized cases. The Tchebycheff method, the weighting method, and the  $\epsilon$ -Constraint method [21] are examples of algorithms for centralized multi-objective optimization problems. More algorithms of this category are surveyed in [21][22]. Such algorithms explore the Pareto optimal set using different prioritizations of the objective functions of the problem. With regard to these techniques, minimizing the sum of objective functions leads to a single element of the Pareto Front. Further exploring this

front can provide additional optimal solutions in different senses. For multi-agent systems, exploring the Pareto Front would provide a larger range of operating conditions for systems based on agents' needs, which can be encoded in heterogeneous weights on objectives. To the best of our knowledge, such methods remain largely unexplored in a multi-agent context.

This paper proposes a distributed algorithm for multi-agent multi-objective set-constrained problems, and the proposed algorithm enables the exploration of the Pareto Front. In particular, a team of  $m$  agents optimizes the weighted sum of convex cost functions  $f(x) = \sum_{i=1}^m w_i f_i(x)$ , where agent  $i$  minimizes  $f_i$ . A common convex set constrains the agents. At the beginning of the optimization process, agents have an initial vector of priorities encoded as weights and an initial vector of decision variables. The proposed algorithm performs four steps at each iteration: *i*) agent  $i$  updates its vector of priorities using those received from other agents in the network, *ii*) the vectors of priorities are used to generate the matrix of information weights for the decision variable update, *iii*) agent  $i$  updates its vector of decision variables with the generated matrix and the decision variables received from its neighbors, and *iv*) agent  $i$  takes a gradient descent step and projects its estimates on the constraint set.

The proposed algorithm belongs to a class of averaging-based distributed optimization algorithms, e.g., [6][8]-[13]. The existing literature considers predominantly problems with doubly-stochastic weights on agents' information exchanges. Indeed, many works rely on the doubly-stochasticity assumption in their model to provide convergence rates and proofs of convergence [6][7][8][14]-

[17]. Computing the infinite product of doubly-stochastic matrices simplifies the analysis of agents' computations, and there exist several rules that ensure the information matrix is doubly-stochastic, such as Metropolis-based weights [23] and the equal-neighbor model [24][25]. These rules restrict communication among agents and do not allow agents to individually prioritize information received from other agents in the network. In addition, these rules require coordination among agents to selected admissible information weights, which can be difficult to achieve if communicating is difficult or costly. The proposed algorithm addresses the limitations related to the doubly-stochasticity assumption in addition to giving agents increased flexibility in their choices. In particular, the following aspects distinguish our algorithm from the existing literature:

- Agents independently prioritize the information received from their neighbors. The sum of each agent's preferences must be 1. While individual agents can easily ensure that their preferences sums to 1, this implies that agents do not have know or consider other agents' preferences. Therefore, preferences of all agents for a particular objective function need not to sum to 1.
- This independence regarding prioritization of objective functions leads to a network-level information exchange matrix that is (non-doubly) stochastic.
- While the agents are reaching an agreement on preferences, they explore the Pareto Front of objective functions. This front exploration leads to optimal

solutions in different senses, which provides broader operating conditions for systems in conformity with agents' needs/preferences.

Because of these distinctions, new theoretical analysis is required to ensure algorithm convergence. In this paper, the proposed algorithm operates over an undirected graph with time-varying weights, and the constraint set is the same for all agents. Theoretical analysis shows that the proposed algorithm drives agents to a common solution. Agents simultaneously reach an agreement on their preferences and compute the optimum with respect to these agents' preferences. Also, we develop convergence rates for the proposed algorithm, which are shown to be significantly influenced by agents' preferences. Numerical simulations show the convergence of the proposed algorithm to the optimal solution along with its convergence rate. The agents' agreement on preferences is also illustrated. Simulations further show that agents' initial preferences directly influence the final results of their computations. This paper is an extension of [26] and it adds proof of convergence and convergence rates, in addition to new simulation results.

The rest of the paper is organized as follows. Section 2 presents background on graph theory and multi-agent interactions. The multi-agent optimization model and the proposed distributed optimization algorithm are provided in Section 3. Section 4 provides proofs of convergence and convergence rates of the proposed algorithm. Section 5 presents numerical results, and Section 6 concludes the paper.

## 2 Graph theory and multi-agent interactions

In this paper, agents' interactions are represented by a connected and undirected graph  $G = (V, E)$ , where  $V = [m] := \{1, 2, \dots, m\}$  is the set of agents and  $E \subset V \times V$  is the set of edges. An edge exists between agent  $i$  and  $j$ , i.e.,  $(i, j) \in E$ , if agent  $i$  communicates with agent  $j$ . By convention,  $(i, i) \notin E$  for all  $i$ . The degree of agent  $i$  is the total number of agents that agent  $i$  communicates with, denoted  $\deg(i)$ . The degree matrix, denoted  $\Delta(G)$ , is a diagonal  $n \times n$  matrix, with  $\deg(i)$  on its diagonal for  $i = 1, \dots, n$ . The maximum vertex degree of  $G$  is  $\Delta_{max} = \max_{i \in [m]} \deg(i)$ .

The adjacency matrix is an  $n \times n$  matrix denoted  $H(G)$ , where  $h_j^i$  is the entry in the  $j$ -th row and  $i$ -th column, defined as

$$h_j^i = \begin{cases} 1 & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases}.$$

Since  $G$  is an undirected graph without self-loops,  $H(G)$  is symmetric with zeros on its main diagonal. The Laplacian matrix associated with  $G$  is also symmetric, and is defined as

$$L(G) = \Delta(G) - H(G). \quad (1)$$

In this paper, we consider an arbitrary graph  $G$ , and, because  $G$  is unambiguous, we will simply write its Laplacian as  $L$ .

### 3 Multi-agent optimization model

In this section, we formally define the class of problems to be solved. Then we propose a multi-objective multi-agent update law for solving them.

#### 3.1 Problem Formulation

In this paper, problems in which agents minimize a prioritized sum of convex objective functions are considered. Agent  $i$  minimizes only the function  $f_i$ , about which we make the following assumption.

**Assumption 1** *For all  $i \in \{1, \dots, m\}$ , the function  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$  is continuously differentiable and convex.*  $\triangle$

All agents' decision variables are constrained to lie in the set  $X$ , about which we assume the following.

**Assumption 2**  *$X$  is non-empty, compact, and convex.*  $\triangle$

We next consider the following optimization problem.

**Problem 3.1** Given functions  $\{f_i\}_{i \in \{1, \dots, m\}}$  satisfying *Assumption 1*,

$$\text{minimize } \sum_{i=1}^m w_i f_i(x), \tag{2}$$

subject to  $x \in X$



where  $x$  is the vector of decision variables,  $w_i$  is a priority assigned to  $f_i$ ,  $\sum_{i=1}^m w_i = 1$ , and  $0 < w_i < 1$  for all  $i$ . Agent  $i$  knows only its objective function  $f_i$ . The constraint set  $X$  is identical for all agents.  $\diamond$

For centralized problems, the priorities  $\{w_i\}_{i \in \{1, \dots, m\}}$  are fixed. Therefore, a standard convex optimization method could solve Problem 1 in a centralized way. However, for decentralized cases, agents may choose different priorities. Agent  $i$  may choose  $\{w_l^i\}_{l \in \{1, \dots, m\}}$  while agent  $j$  chooses  $\{w_l^j\}_{l \in \{1, \dots, m\}}$ , with  $w_l^i \neq w_l^j$  for all  $l$ .

As this occurs, these priorities provide each agent with the flexibility to have preferences. For instance, mobile autonomous agents generating a trajectory may want to optimize fuel usage and travel time, and each agent can prioritize these two objectives according to their own needs. If agents' priorities differ, agents are solving different problems because they minimize different overall objective functions. As a result, reaching a common solution requires devising an optimization algorithm and driving agent priorities to a common value.

Changing agents' priorities from their initial values implies that no single agents' preferences are obeyed exactly. However, the net change across all agents can be done fairly. One such way is to drive all agent's priorities to their average value. While one could envision first computing the average priorities and then optimizing, this is undesirable because it requires solving two separate problems sequentially. Instead, we devise an update law that drives

agents to a common solution by interlacing optimization steps with priority averaging steps. Also, this interlacing enables agents to continuously modify their preferences based on the task at hand.

### 3.2 Proposed Update Law

At iteration  $k$ , agent  $i$  updates its priority vector  $w^i$ . Agent  $i$  assigns a priority to all agents (corresponding to the objective function updated by that agent), even though agent  $i$  does not communicate with all agents. This provides agent  $i$  with a way to influence all final priorities, and, as will be shown below, affect the final results agents attain. Agent  $i$  also updates its decision vector  $x^i$  by adding the weighted estimates received from its neighbors, then minimizing its objective function  $f_i$  through gradient step, and then projecting its new estimate on its constraint set  $X$ . We have

$$w^i(k+1) = w^i(k) + c \sum_{j=1}^n h_i^j(w^j(k) - w^i(k)) \quad (3)$$

$$a_j^i(k+1) = q_j^i w_j^i(k) + \sum_{\substack{j=i \\ j=1}}^m w_j^i \tilde{q}_j^i \quad (4)$$

$$v^i(k) = \sum_{j=1}^n a_j^i(k) x^j(k) \quad (5)$$

$$x^i(k+1) = P_X[v^i(k) - \alpha_k d_i(k)], \quad (6)$$

where  $0 < c < 1/\Delta_{max}$ ,  $h_j^i(k)$  is the  $j^{th}$   $i^{th}$  entry of  $H(G)$ ,  $a_j^i(k)$  is the weight that agent  $i$  assigns to the data provided by agent  $j$  at iteration  $k$ ,  $q_j^i$  is the  $j^{th}$   $i^{th}$  entry of  $Q$ , where  $Q = H + I$  and  $I$  is the identity matrix,  $\tilde{q}_j^i = 1 - q_j^i$ ,  $\alpha_k$  is the gradient step size for all agents at time  $k$ ,  $P_{X_i}$  is the projection operation, and  $d_i$  is the gradient vector of agent  $i$  at  $x^i(k)$ . Formally,  $d_i(k) = \nabla f_i(x^i(k))$ .

The equivalent network-level representation of (3)-(10) is

$$W(k+1) = PW(k), \quad (7)$$

where  $P = I - cL(G)$  and  $W(k)$  is the column matrix of agents' priorities, along with  $A(k)$  in which its column vectors are  $a^i(k)$  for  $i = \{1, \dots, m\}$ , and  $V(k)$  in which its column vectors are  $v^i(k)$  for  $i = \{1, \dots, m\}$ .

$$A(k) = Q \circ W(k) + (W(k) \circ \tilde{Q})J \circ I, \quad (8)$$

$$V(k) = A(k)X(k), \quad (9)$$

$$X(k+1) = P_{X_c}[V(k) - \alpha_k D(k)]. \quad (10)$$

where  $\circ$  denotes the Hadamard product,  $P_{X_c}$  is the projection operation that projects each column of  $V(k) - \alpha_k D(k)$  individually,  $D(k)$  is the column matrix of  $d^i(k)$  for  $i = \{1, \dots, m\}$ , and  $X(k)$  is the column matrix of  $x^i(k)$  for  $i = \{1, \dots, m\}$ . In line with the multi-objective optimization concept, our algorithm uses the priority vectors,  $w^i$  for  $i = \{1, \dots, m\}$ , to quantify the importance of information received to update  $x^i$  for  $i = \{1, \dots, m\}$ . Contrary to

most existing works, the  $A(k)$  matrix is a function of  $W(k)$  and  $A(k)$  matrix is stochastic, instead of doubly-stochastic. This occurs because an agent can ensure that its weights sum to 1, though different agents' weights for a particular objective need not to sum to 1. This implies that  $A(k)$ 's row sums need not to equal 1.

In (8),  $Q \circ W(k)$  computes the Hadamard product between  $Q$  and  $W(k)$ , where the resulting matrix contains  $w_j^i(k)$  for  $(i, j) \in E$ ,  $w_i^i(k)$  for all  $i$ , and the remaining terms are set to zero. Therefore, if agent  $i$  does not communicate with agent  $j$ , a zero is assigned to that agent. Regarding the second term of (8),  $(W(k) \circ \tilde{Q})J \circ I$  creates a diagonal matrix, where the diagonal terms are the sum of each row. The first term summed to the second term in (8) means that agent  $i$  assigns to itself the weights  $w_j^i$  if  $(i, j) \notin E$  and assigns a zero value to the entries of the  $i$ -th row and  $j$ -th column for  $(i, j) \notin E$ .

The next lemma pertains to the weights of the  $A$  matrix and the communication between agents.

**Lemma 3.1** *Since  $a_j^i(k)$  is obtained from (3) and (4), we have*

1.  $a_i^i(k) \geq \min_{j \in [n]} \min_{i \in [n]} w_j^i(0)$  for all  $k \geq 0$  and all  $i$ .
2.  $a_j^i(k) \geq \min_{j \in [n]} \min_{i \in [n]} w_j^i(0)$  for all  $k \geq 0$  and all  $(i, j) \in E$ .
3.  $a_j^i(k) = 0$  for all  $k$  if  $(i, j) \notin E$ .

*Proof* See Appendix.

To simplify the forthcoming development, Eq. (10) can be written as follows [5],

$$x^i(k+1) = v^i(k) - \alpha_k d_i(k) + \phi^i(k), \quad (11)$$

$$\phi^i(k) = P_X[v^i(k) - \alpha_k d_i(k)] - v^i(k) + \alpha_k d_i(k). \quad (12)$$

For all  $i$  and for all  $k$  and  $s$  where  $k > s$ , the above equivalent form allows us to express the decision variable update over time as:

$$\begin{aligned} x^i(k+1) = & \sum_{j=1}^m [\Phi(k, s)]_j^i x^j(s) - \sum_{r=s}^{k-1} \sum_{j=1}^m [\Phi(k, r+1)]_j^i \alpha_r d_j(r) \\ & - \alpha_k d_i(k) + \sum_{r=s}^{k-1} \sum_{j=1}^m [\Phi(k, r+1)]_j^i \phi^j(r) + \phi^i(k), \end{aligned} \quad (13)$$

where the transition matrix  $\Phi(k, s) = A(k)A(k-1), \dots, A(s)$  [6].

#### 4 Convergence of the proposed algorithm

This section provides the convergence analysis for the proposed algorithm (3)-(10).

The following well-known lemma confirms that the priority update (3) does indeed compute average priorities.

**Lemma 4.1**  $\lim_{k \rightarrow \infty} w^i(k) = \bar{w} = \sum_{j=1}^m \frac{w^j(0)}{m}$  for  $j = 1, \dots, m$ . At the network level,  $W(k) = \bar{W}$ , where  $\bar{W} = \mathbb{1}\bar{w}^\top$ .

*Proof* See [27][28].

From Assumption 1, the gradient is continuous and from Assumption 2  $X$  is compact. Therefore, we have  $\|d_i(k)\| \leq L$  for all  $i$ . From that statement, Lemma 4.2 follows.

**Lemma 4.2** *The errors  $\phi^i(k)$  satisfy  $\|\phi^i(k)\| \leq \alpha_k L$  for all  $i$  and  $k$ .*

*Proof* See [5].

The next Lemma describes the convergence behavior of  $\Phi(k, s)$ .

**Lemma 4.3** *From Lemma 3.1, the convergence of  $\Phi(k, s)$  is geometric according to*

$$|[\Phi(k, s)]_i^j - \gamma^j(s)| \leq C\beta^{k-s}, \quad (14)$$

where  $B_0 = m - 1$ ,  $m$  is the number of agents,  $\gamma^j(s) = \lim_{k \rightarrow \infty} [\Phi(k, s)]_i^j$ ,  
 $C = 2 \frac{(1 + \min_{j \in [m]} \min_{i \in [m]} w_i^j(0)^{-B_0})}{1 - \min_{j \in [m]} \min_{i \in [m]} w_i^j(0)^{B_0}}$ , and  $\beta = (1 - \min_{j \in [m]} \min_{i \in [m]} w_i^j(0)^{B_0})^{1/B_0}$ .

*Proof* See Lemma 3 and Lemma 4 in [6] and Lemma 3.1 above.

To prove the convergence results, we use the following lemmas [5].

**Lemma 4.4** *Assume that  $0 < \rho < 1$ ,  $\{\lambda_k\}_{k \in \mathbb{N}}$  be a positive scalar sequence,*

*and  $\lim_{k \rightarrow \infty} \lambda_k = 0$ . Then,*

$$\lim_{k \rightarrow \infty} \sum_{l=0}^k \rho^{k-l} \lambda_l = 0$$

*Moreover, if  $\sum_k \lambda_k < \infty$ , we have*

$$\sum_{k=1}^{\infty} \sum_{l=0}^k \rho^{k-l} \lambda_l < \infty$$

.

*Proof* See proof for Lemma 7 in [5].

**Lemma 4.5** Assume that  $X$  is a nonempty closed convex set in  $\mathbb{R}^n$ . Thus, we obtain for any  $x \in \mathbb{R}^n$ ,  $\|P_X[x] - y\|^2 \leq \|x - y\|^2 - \|P_X[x] - x\|^2$  for all  $y \in X$ .

*Proof* See proof for Lemma 1(b) in [5].

**Lemma 4.6** Let  $x^i(k)$  be generated by (9)-(10). We have for any  $z \in X$  and all  $k \geq 0$ ,

$$\begin{aligned} \sum_{i=1}^m \|x^i(k+1) - z\|^2 &\leq \sum_{i=1}^m \sum_{j=1}^m a_j^i(w^i(k)) \|x^j(k) - z\|^2 + \\ &\alpha_k^2 \sum_{i=1}^m \|d_i(k)\|^2 - 2\alpha_k \sum_{i=1}^m (f_i(v^i(k)) - f_i(z)) - \sum_{i=1}^m \|\phi^i(k)\|^2. \end{aligned}$$

*Proof* See Appendix.

The following lemma demonstrates that disagreements between agents go to 0, namely that  $\|x^i(k) - x^j(k)\|$  as  $k \rightarrow \infty$ . To assess agent disagreements, we consider agents' disagreements with the average of their decision variables,

$$y(k) = \frac{1}{m} \sum_{j=1}^m x^j(k). \quad (15)$$

In view of (9) and (11), we have

$$y(k+1) = \frac{1}{m} \sum_{i=1}^m \sum_{j=1}^m a_j^i(w^i(k)) x^j(k) - \frac{\alpha_k}{m} \sum_{i=1}^m d_i(k) + \frac{1}{m} \sum_{i=1}^m \phi^i(k). \quad (16)$$

**Lemma 4.7** *Let the algorithm generate iterates of  $x^i(k)$  by the algorithm (9)-(10) and consider  $\{y(k)\}_{k \in \mathbb{N}}$  defined in (16).*

(a) *If the stepsize is decreasing such as  $\lim_{k \rightarrow \infty} \alpha_k = 0$ , thus*

$$\lim_{k \rightarrow \infty} \|x^i(k) - y(k)\| = 0 \text{ for all } i.$$

(b) *If  $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$  therefore*

$$\sum_{k=1}^{\infty} \alpha_k \|x^i(k) - y(k)\| < \infty \text{ for all } i.$$

*Proof* See Appendix.

From Lemma 4.7(a), the following theorem is obtained regarding the convergence rate of  $\|x^i(k) - y(k)\|$ . As it has been demonstrated that agents' disagreements go to 0, as  $k \rightarrow \infty$  (Lemma 4.7a), this theorem shows the rate to reach agreement on agents' decision variable.

**Theorem 4.1** *Following Assumption 2, there is an  $M$  such that  $\sum_{j=1}^m \|x^j(0)\| \leq M$ . Let  $\epsilon > 0$  be given and let  $K$  be the first time that  $\alpha_k \leq \epsilon$ . Let  $C$  be defined as in Lemma 4.3. Then  $0 < \min_{j \in [m]} \min_{i \in [m]} w_i^j(0) < 1$ ,  $\alpha$  is decreasing, and  $\lim_{k \rightarrow \infty} \alpha_k = 0$ , and for all  $k \geq K + 3$ , we have*

$$\begin{aligned} \|x^i(k) - y(k)\| &\leq 2mCM \left( (1 - \min_{j \in [m]} \min_{i \in [m]} w_i^j(0)^{B_0})^{(k-1)/B_0} \right) \\ &\quad + 4mCL\alpha_0 \frac{((1 - \min_{j \in [m]} \min_{i \in [m]} w_i^j(0)^{B_0})^{(k-K)/B_0})}{1 - (1 - \min_{j \in [m]} \min_{i \in [m]} w_i^j(0)^{B_0})^{1/B_0}} + 4\alpha_{k-1}L + \frac{4mCL\alpha_0\epsilon}{1 - \beta}. \end{aligned}$$



*Proof* Recall (56) and  $\beta = ((1 - \min_{j \in [m]} \min_{i \in [m]} w_i^j(0)^{B_0})^{1/B_0})$ :

$$\|x^i(k) - y(k)\| \leq 2mC\beta^{k-1} \sum_{j=1}^m \|x^j(0)\| + 4mCL \sum_{r=0}^{k-2} \beta^{k-r} \alpha_r + 4\alpha_{k-1}L.$$

Then, we have

$$\|x^i(k) - y(k)\| \leq 2mCM\beta^{k-1} + 4mCL \sum_{r=0}^{k-2} \beta^{k-r} \alpha_r + 4\alpha_{k-1}L. \quad (17)$$

Suppose we have an arbitrary  $\epsilon > 0$  and let  $K$  be defined so that  $\alpha_r \leq \epsilon$  (since  $\alpha_r \rightarrow 0$ ) for all  $k \geq K + 3$ . We therefore have

$$\sum_{r=0}^{k-2} \beta^{k-r} \alpha_r \leq \sum_{r=0}^K \beta^{k-r} \alpha_r + \epsilon \sum_{r=K+1}^{k-2} \beta^{k-r} \leq \max_{0 \leq t \leq K} \alpha_t \sum_{r=0}^K \beta^{k-r} + \epsilon \sum_{r=K+1}^{k-2} \beta^{k-r}. \quad (18)$$

Because of  $\sum_{r=K+1}^{k-2} \beta^{k-r} \leq \frac{1}{1-\beta}$ , we obtain

$$\sum_{r=0}^{k-2} \beta^{k-r} \alpha_r \leq \max_{0 \leq t \leq K} \alpha_t \sum_{r=0}^K \beta^{k-r} + \frac{\epsilon}{1-\beta}. \quad (19)$$

Similarly, since  $\sum_{r=0}^K \beta^{k-r} \leq \frac{\beta^{k-K}}{1-\beta}$ , we obtain for all  $k \geq K + 3$ ,

$$\sum_{r=0}^{k-2} \beta^{k-r} \alpha_r \leq \max_{0 \leq t \leq K} \alpha_t \frac{\beta^{k-K}}{1-\beta} + \frac{\epsilon}{1-\beta}. \quad (20)$$

Inserting (20) into (17), we get for  $k \geq K + 3$ ,

$$\|x^i(k) - y(k)\| \leq 2mCM\beta^{k-1} + 4mCL \left[ \max_{0 \leq t \leq K} \alpha_t \frac{\beta^{k-K}}{1-\beta} + \frac{\epsilon}{1-\beta} \right] + 4\alpha_{k-1}L. \quad (21)$$

Because  $\alpha_t$  is decreasing, we obtain for all  $k \geq K + 3$ ,

$$\|x^i(k) - y(k)\| \leq 2mCM\beta^{k-1} + 4mCL\alpha_0 \left[ \frac{\beta^{k-K}}{1-\beta} + \frac{\epsilon}{1-\beta} \right] + 4\alpha_{k-1}L. \quad (22)$$

□

The convergence rate is affected by the value of  $\beta$ . Recall  $\beta = (1 - \min_{j \in [n]} \min_{i \in [n]} w_i^j(0)^{B_0})^{1/B_0}$ , meaning the value of  $\beta$  is a function of the minimum initial priority and the number of agents. The convergence rate slows down as the minimum initial agent weight decreases and the number of agents increases. Agents should therefore carefully choose their preferences. A small initial priority would make the convergence rate very slow, which can harm algorithm performance. This suggests that agents' priorities must be balanced with need for attaining a high-quality final result with a reasonable convergence rate. Along the same lines, an extremely large team of agents would increase the limit of the convergence rate; as the number of agents increases agents' preferences associated to objective functions tend to be smaller since agents' preferences sum to 1.

Based on Lemmas 4.6 and 4.7, the next theorem presents the asymptotic convergence of the proposed algorithm. In distinction to [5], it is shown that the iterates  $x^i(k)$  converge to an optimal solution for an information exchange matrix  $A$  that is (non-doubly) stochastic, which weights are obtained from agents' priorities (8).

**Theorem 4.2** *The iterates  $x^i(k)$  are generated by (7)-(10) with stepsize satisfying conditions of Lemma 4.7. Assume that the optimal solutions set  $X^*$  is nonempty. Therefore, an optimal point  $x^* \in X^*$  exists such that*

$$\lim_{k \rightarrow \infty} \|x^i(k) - x^*\| = 0 \text{ for all } i.$$

*Proof* From Lemma 4.6, we have

$$\begin{aligned} \sum_{i=1}^m \|x^i(k+1) - z\|^2 &\leq \sum_{i=1}^m \sum_{j=1}^m a_j^i(w^i(k)) \|x^j(k) - z\|^2 + \\ &\alpha_k^2 \sum_{i=1}^m \|d_i(k)\|^2 - 2\alpha_k \sum_{i=1}^m (f_i(v^i(k)) - f_i(z)) - \sum_{i=1}^m \|\phi^i(k)\|^2. \end{aligned}$$

Using the gradient bound and by removing the last nonpositive term on the right hand side, we get

$$\begin{aligned} \sum_{i=1}^m \|x^i(k+1) - z\|^2 &\leq \sum_{i=1}^m \sum_{j=1}^m a_j^i(w^i(k)) \|x^j(k) - z\|^2 + \alpha_k^2 m L^2 \\ &\quad - 2\alpha_k \sum_{i=1}^m (f_i(v^i(k)) - f_i(y(k))) - 2\alpha_k (f(y(k)) - f(z)). \end{aligned} \tag{23}$$

Considering the gradient boundedness and the stochasticity of weights, we have

$$|f_i(v^i(k)) - f_i(y(k))| \leq L \|v^i(k) - y(k)\| \leq L \sum_{j=1}^m a_j^i(w^i(k)) \|x^j(k) - y(k)\|. \tag{24}$$

Summing (24) over  $m$  and using it in (23), we obtain,

$$\begin{aligned}
\sum_{i=1}^m \|x^i(k+1) - z\|^2 &\leq \sum_{i=1}^m \sum_{j=1}^m a_j^i(w^i(k)) \|x^j(k) - z\|^2 + \alpha_k^2 m L^2 \\
&+ 2\alpha_k L \sum_{i=1}^m \sum_{j=1}^m a_j^i(k) \|x^j(k) - y(k)\| - 2\alpha_k (f(y(k)) - f(z)).
\end{aligned} \tag{25}$$

Considering  $z = x^* \in X^*$ , and by restructuring the terms we get,

$$\begin{aligned}
\sum_{i=1}^m \|x^i(k+1) - x^*\|^2 + 2\alpha_k (f(y(k)) - f(x^*)) &\leq \\
\sum_{i=1}^m \sum_{j=1}^m a_j^i(w^i(k)) \|x^j(k) - x^*\|^2 + \alpha_k^2 m L^2 + & \\
2\alpha_k L \sum_{i=1}^m \sum_{j=1}^m a_j^i(w^i(k)) \|x^j(k) - y(k)\|. &
\end{aligned} \tag{26}$$

By summing (26) over an arbitrary window from some positive integer  $K$  to  $N$  with  $K < N$ , we obtain,

$$\begin{aligned}
\sum_{i=1}^m \|x^i(N+1) - x^*\|^2 + 2 \sum_{k=K}^N \alpha_k (f(y(k)) - f(x^*)) &\leq \\
\sum_{i=1}^m \sum_{j=1}^m a_j^i(w^i(K)) \|x^j(K) - x^*\|^2 + & \\
m L^2 \sum_{k=K}^N \alpha_k^2 + 2L \sum_{k=K}^N \alpha_k \sum_{i=1}^m \sum_{j=1}^m a_j^i(w^i(k)) \|x^j(k) - y(k)\|. &
\end{aligned} \tag{27}$$

With  $K = 1$  and  $N \rightarrow \infty$  in (27), using  $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$  and  $\sum_{k=1}^{\infty} \alpha_k \sum_{j=1}^m \|x^j(k) - y(k)\| < \infty$ , which is a result of Lemma 4.7, we have

$$\sum_{k=1}^{\infty} \alpha_k (f(y(k)) - f(x^*)) < \infty.$$

Because  $x^j(k) \in X$  for all  $j$ ,  $y(k) \in X$  for all  $k$ . Given that  $x^* \in X^*$ ,  $f(y(k)) - f^* \geq 0$  for all  $k$ . As a result of this relation and the assumption that  $\sum_{k=1}^{\infty} \alpha_k = \infty$ , and  $\sum_{k=1}^{\infty} \alpha_k (f(y(k)) - f(x^*)) < \infty$ , we obtain,

$$\liminf_{k \rightarrow \infty} (f(y(k)) - f(x^*)) = 0. \quad (28)$$

The forthcoming development demonstrates that agents converge to the optimal point  $x^*$ . The nonnegative term in left side hand of (27) can be removed. Therefore, we have

$$\begin{aligned} \sum_{i=1}^m \|x^i(N+1) - x^*\|^2 &\leq \sum_{i=1}^m \sum_{j=1}^m a_j^i(w^i(K)) \|x^j(K) - x^*\|^2 + \\ &mL^2 \sum_{k=K}^N \alpha_k^2 + 2L \sum_{k=K}^N \alpha_k \sum_{i=1}^m \sum_{j=1}^m a_j^i(w^i(K)) \|x^j(k) - y(k)\|. \end{aligned} \quad (29)$$

Given that  $\sum_k \alpha_k^2 < \infty$  and  $\sum_{k=1}^{\infty} \alpha_k \sum_{i=1}^m \sum_{j=1}^m a_j^i(w^i(k)) \|x^i(k) - y(k)\| < \infty$ , it results that  $x^i(k)$  is bounded for each  $i$ , and

$$\limsup_{N \rightarrow \infty} \sum_{i=1}^m \|x^i(N+1) - x^*\|^2 \leq \liminf_{K \rightarrow \infty} \sum_{i=1}^m \sum_{j=1}^m a_j^i(w^i(K)) \|x^j(K) - x^*\|^2.$$

This implies that the scalar sequence  $\sum_{i=1}^m \|x^i(k) - x^*\|$  converges for every  $x^* \in X^*$ .

Given that  $\lim_{k \rightarrow \infty} \|x^i(k) - y(k)\| = 0$  (Lemma 4.7),  $\{y(k)\}_{k \in \mathbb{N}}$  is bounded and the scalar sequence  $\|y(k) - x^*\|$  is convergent for  $x^* \in X^*$ .

Because  $y(k)$  is bounded,  $y(k)$  has a limit point. From (28), we have  $\liminf_{k \rightarrow \infty} f(y(k)) = f^*$ . Considering the previous equality and the continuity of  $f$ , one of the limit points of  $\{y(k)\}$  must be in  $X^*$ , which is denoted by  $x^*$ . Therefore,  $\|y(k) - x^*\|$  is convergent. Thus,  $\lim_{k \rightarrow \infty} y(k) = x^*$  and  $\lim_{k \rightarrow \infty} \|x^i(k) - y(k)\| = 0$ , which implies that each sequence  $\{x^i(k)\}$  converges to the same  $x^* \in X^*$ .  $\square$

From Theorem 4.2 and Lemma 4, the following convergence rate is obtained.

**Theorem 4.3** *Let  $\epsilon > 0$  be given and let  $K$  be the first time that  $\alpha_k \leq \epsilon$ .*

*Using Lemma 4 and Theorem 4.1, we have for  $s \geq K + 3$ ,*

$$\begin{aligned} \sum_{i=1}^m \|x^i(k+1) - x^*\|^2 &\leq \sum_{i=1}^m \sum_{j=1}^m q_j^i \omega_{max}^{(k+1)} \|x^j(s) - x^*\|^2 \\ &+ \sum_{r=s}^k \sum_{i=1}^m \sum_{j=1}^m q_j^i \omega_{max}^{(k+1-r)} \alpha_r L \left[ \alpha_r L + 4mCM\beta^{r-1} \right. \\ &\quad \left. + 8mCL\alpha_0 \left[ \frac{\beta^{r-K}}{1-\beta} + \frac{\epsilon}{1-\beta} \right] + 8\alpha_{r-1}L \right]. \end{aligned} \quad (30)$$

*Proof* From (25), we have

$$\begin{aligned} \sum_{i=1}^m \|x^i(k+1) - z\|^2 &\leq \sum_{i=1}^m \sum_{j=1}^m a_j^i(w^i(k)) \|x^j(k) - z\|^2 + \alpha_k^2 m L^2 \\ &+ 2\alpha_k L \sum_{i=1}^m \sum_{j=1}^m a_j^i(k) \|x^j(k) - y(k)\| - 2\alpha_k (f(y(k)) - f(z)). \end{aligned} \quad (31)$$

Dropping the last negative term, we find

$$\begin{aligned}
\sum_{i=1}^m \|x^i(k+1) - x^*\|^2 &\leq \sum_{i=1}^m \sum_{j=1}^m a_j^i(w^i(k)) \|x^j(k) - x^*\|^2 + \alpha_k^2 m L^2 \\
&\quad + 2\alpha_k L \sum_{i=1}^m \sum_{j=1}^m a_j^i(k) \|x^j(k) - y(k)\|.
\end{aligned} \tag{32}$$

Re-arranging the terms, we have

$$\begin{aligned}
\sum_{i=1}^m \|x^i(k+1) - x^*\|^2 &\leq \sum_{i=1}^m \sum_{j=1}^m a_j^i(w^i(k)) \left[ \|x^j(k) - x^*\|^2 \right. \\
&\quad \left. + \alpha_k^2 L^2 + 2\alpha_k L \|x^j(k) - y(k)\| \right].
\end{aligned} \tag{33}$$

Define  $\omega(k) = \max_{j \in [n]} \max_{i \in [n]} a_j^i(k)$ . Therefore, the maximum value that  $\omega(k)$  can

take is  $\omega_{max} = 1 - \min_{j \in [n]} \min_{i \in [n]} w_i^j(0)$ . We therefore obtain

$$\begin{aligned}
\sum_{i=1}^m \|x^i(k+1) - x^*\|^2 &\leq \sum_{i=1}^m \sum_{j=1}^m q_j^i \omega_{max} \left[ \|x^j(k) - x^*\|^2 + \right. \\
&\quad \left. \alpha_k^2 L^2 + 2\alpha_k L \|x^j(k) - y(k)\| \right].
\end{aligned} \tag{34}$$

Using Theorem 4.1, we obtain

$$\begin{aligned}
\sum_{i=1}^m \|x^i(k+1) - x^*\|^2 &\leq \sum_{i=1}^m \sum_{j=1}^m q_j^i \omega_{max}^{(k+1)} \|x^j(s) - x^*\|^2 \\
&\quad + \sum_{r=s}^k \sum_{i=1}^m \sum_{j=1}^m q_j^i \omega_{max}^{(k+1-r)} \alpha_r L \left[ \alpha_r L + 4mCM\beta^{r-1} \right. \\
&\quad \left. + 8mCL\alpha_0 \left[ \frac{\beta^{r-K}}{1-\beta} + \frac{\epsilon}{1-\beta} \right] + 8\alpha_{r-1}L \right].
\end{aligned} \tag{35}$$

□

The convergence rate is determined by  $\omega_{max}$ . Since  $\omega_{max} = 1 - \min_{j \in [n]} \min_{i \in [n]} w_i^j(0)$ , the initial agents' weights influence the convergence rate. If

the smallest initial weight is extremely small, it could be detrimental for the algorithm performance as it would slow down significantly the convergence rate. Agents should consider balancing their need for reaching a high-quality final result and reasonable convergence rate. Agents should avoid extreme difference in their highest and lowest priorities.

## 5 Numerical results

Three simulation scenarios are run to illustrate the performance of the proposed algorithm. The numerical studies considers quadratic functions defined as,

$$f_i(x) = \frac{1}{2}x^T Q_i x + r_i^T x + c_i, \quad (36)$$

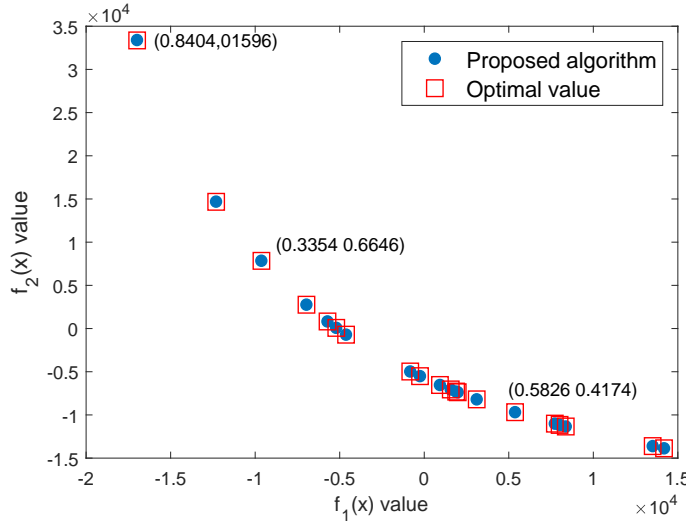
where  $x \in \mathbb{R}^n$  is the decision vector,  $Q \in \mathbb{R}^{n \times n}$  is a symmetric positive definite matrix,  $r \in \mathbb{R}^n$ , and  $i = 1, \dots, n$ . The matrix  $Q_i$  and the vector  $r_i$  and  $c_i$  are generated randomly and unique for each agent. An agent  $i$  knows exclusively the objective function  $f_i$ . The agents goal is to solve the following problem using (3)-(10):

$$\text{minimize } \sum_{i=1}^m w_i f_i(x). \quad (37)$$

subject to  $x \in [-1,000 \text{ to } 1,000]$ .

For all scenario, the initial gradient step size is  $\alpha_0 = 0.2$  and we let  $\alpha_k = \frac{\alpha_0}{k}$ .





**Fig. 1**  $f_2(x)$  in terms of  $f_1(x)$  for a team of 2 agents. It can be seen that the proposed algorithm reaches various optimal solutions. Some of agents' priorities are shown. As the priority of  $f_1$  diminishes, the value of  $f_1$  increases and the value of  $f_2$  decreases.

### 5.1 First simulation scenario

The first simulation aims to show the exploration of the Pareto Front by the algorithm. For illustrative purposes, the team has two agents and the number of decision variables is 10. Simulations with different initial agent priorities have been performed with identical initial states. Agents exchange information 100,000 times. Fig. 1 presents points on the Pareto Front obtained.

Optimal solutions in different senses are reached by the agents. The network controls the exploration of the Front through the assignment of priorities, providing a wider range of "optimal" solutions in different senses.

**Table 1** Initial priorities and priorities value for consensus

	Agent			
	1	2	3	$\bar{w}$
$w_1$	0.3495	0.2232	0.6315	0.4014
$w_2$	0.3027	0.3838	0.2494	0.3119
$w_3$	0.3478	0.3930	0.1191	0.2866

## 5.2 Second simulation scenario

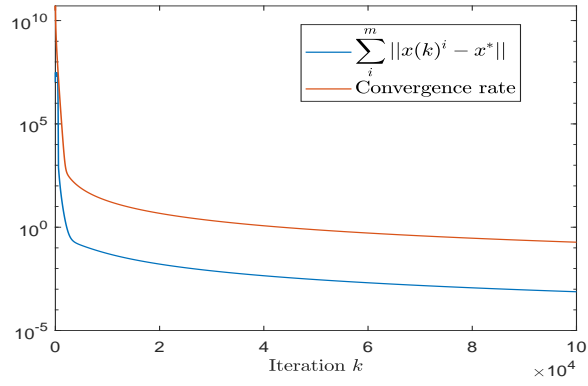
The second scenario aims to show graphically the convergence rate towards the optimal solution, i.e., (35). The team consists of three agents,  $m = 3$ , and the decision vector has 10 variables, i.e.,  $n = 10$ . The team minimizes 10 quadratic functions as defined by (36) to solve (37). The network exchanges information 100,000 times. Table 1 shows the initial agent preferences,  $w_i$ , and the convergence of the priority vector,  $\bar{w}$ . The sum of each agent priorities equals 1, i.e.,  $\sum_{i=1}^3 w_i^j = 1$ .

Table 2 presents the results obtained by the proposed algorithm. The first three columns correspond to the initial decision vector of each agent. The fourth column presents the final average estimate reached by the agents, i.e.,  $y(k)$  for  $k \rightarrow \infty$ . The last column shows the optimal solution.

The results obtained by the proposed algorithm closely approach the optimal value,  $x^*$ . Fig. 2 presents the algorithm's convergence rate calculated with (35) and  $K = 1$ .

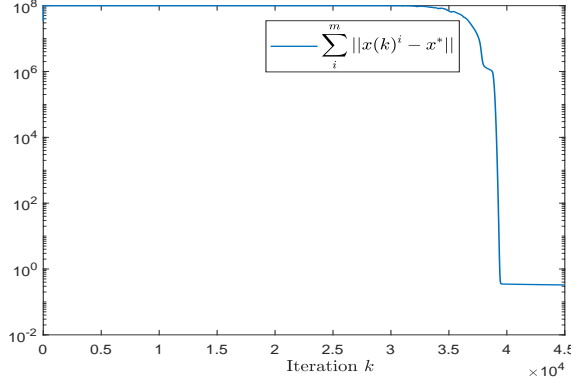
**Table 2** Result obtained by the proposed algorithm for Scenario 1

$x_1$	$x_2$	$x_3$	$\hat{x}$	$x^*$
-728.77	-284.03	-981.79	16.72	16.71
-94.90	406.18	951.03	-0.53	-0.53
429.65	792.26	-532.88	-9.03	-9.02
14.82	-360.08	73.41	5.52	5.52
846.91	-797.02	147.99	-4.74	-4.74
-789.88	986.15	-602.51	8.16	8.15
-285.74	723.87	-584.77	1.01	1.01
-820.97	39.10	-30.25	-5.07	-5.07
634.15	-431.47	888.04	-13.59	-13.58
-352.03	361.52	-10.59	-8.68	-8.67
$f(x)$			-6.1094e+03	-6.1094e+03

**Fig. 2** Convergence rate of the proposed algorithm - Team of 3 agents. The agents converge towards the optimal solution.

### 5.3 Third simulation scenario

The third simulation scenario objective is to demonstrate the proposed algorithm's efficiency on a larger team of agents and higher number of decision



**Fig. 3** Convergence rate of the proposed algorithm for a team of 100 agents. During the first iterations, agents' decision variables are close to the boundary of the constraint set, which explains the values  $\sum_{i=1}^m \|x^i(k+1) - x^*\|^2$  obtained during the first iterations. It takes several iterations to move agents away from the boundaries. However, once agents are far from the boundaries, agents converge quickly towards the optimal solution.

variables. The team consists of 100 agents with quadratic functions defined by (36) of 100 variables. Therefore, the agent teams solve Problem 3.1 where  $m = 100$ . The set of constraints is the same as scenario 1 and the quadratic functions are also created randomly. Fig. 3 displays  $\sum_{i=1}^m \|x^i(k+1) - x^*\|^2$  over the course of the algorithm. As  $k \rightarrow \infty$ , the  $\sum_{i=1}^m \|x^i(k+1) - x^*\|^2 \rightarrow 0$ , which means the agent team approximately reach the optimal solution. Indeed,  $f(x^*) = -50.11$  and  $f(\hat{x}) = -49.64$ .

A plateau followed by sharp drop is observed in the curve. The high bound for the gradient compared to the constraint set explains this phenomenon. Since the gradient can take large values, the decision variables may take large values too. However, the  $x_i$  is subject to the constraint set  $[-1,000 \text{ to } 1,000]$ . Therefore, during the first iterations, most of the decision variables are pro-

jected onto the limits of the constraint set. It takes several iterations before a significant number of agents move away from the boundary of the constraint set. However, once this number is reached, the algorithm converges quickly towards the optimal solution.

## 6 Conclusions

In this paper, a distributed algorithm to optimize a prioritized sum of convex objective functions was proposed. The proposed algorithm allows agents to have different priorities regarding other agents' objective functions. These agents' priorities enable the exploration of the Pareto Front, which provides optimal solutions in different senses. A rule based on agents' priority generates the information exchange matrix used to update agents' estimates. In the proposed algorithm, this matrix is stochastic, whereas, in most other distributed algorithms, the information exchange matrix is doubly-stochastic. Therefore, new theoretical analyses were needed because of the difference in the network-level set-up. It has been proved that the proposed algorithm converged towards the optimal solution. Also, convergence rates were obtained, which are influenced by agents' initial weights. Numerical results illustrated the performance of the proposed algorithm. Future works include time-varying topology and implementing the algorithm on a team of robots.

**Acknowledgements** Maude J. Blondin would like to thank the support of Fonds de recherche Nature et technologies postdoctoral fellowship.

## Appendix:

This appendix contains the proofs for some lemmas presented in the paper.

*Proof of Lemma 3.1 [26]*

Define  $\mu(k) := \min_{j \in [m]} \min_{i \in [m]} w_i^j(k)$ . Then,  $W(k+1) = PW(k)$  can be expressed as

$$\begin{aligned}
 & \begin{bmatrix} p_1^1 & \dots & p_1^m \\ \vdots & \vdots & \vdots \\ p_i^1 & \dots & p_i^m \\ \vdots & \vdots & \vdots \\ p_m^1 & \dots & p_m^m \end{bmatrix} \begin{bmatrix} \mu(k) + \delta_1^1(k) & \dots & \mu(k) + \delta_1^m(k) \\ \vdots & \vdots & \vdots \\ \mu(k) + \delta_i^1(k) & \dots & \mu(k) + \delta_i^m(k) \\ \vdots & \vdots & \vdots \\ \mu(k) + \delta_m^1(k) & \dots & \mu(k) + \delta_m^m(k) \end{bmatrix} \\
 &= \begin{bmatrix} w_1^1(k+1) & \dots & w_1^m(k+1) \\ \vdots & \vdots & \vdots \\ w_i^1(k+1) & \dots & w_i^m(k+1) \\ \vdots & \vdots & \vdots \\ w_m^1(k+1) & \dots & w_m^m(k+1) \end{bmatrix}, \tag{38}
 \end{aligned}$$

where  $\delta_i^j(k) = w_i^j(k) - \mu(k) \geq 0$  for  $i, j = \{1, \dots, n\}$ . Then, we have

$$\begin{aligned}
 w_i^j(k+1) &= \sum_{n=1}^m p_i^n [\mu(k) + \delta_n^j(k)] = \sum_{n=1}^m p_i^n \mu(k) + \sum_{n=1}^m p_i^n \delta_n^j(k) \\
 &= \mu(k) \sum_{n=1}^m p_i^n + \sum_{n=1}^m p_i^n \delta_n^j(k). \tag{39}
 \end{aligned}$$

By definition, we know that  $\sum_{n=1}^m p_i^n = 1$ . Therefore, we get

$$w_i^j(k+1) = \mu(k) + \sum_{n=1}^m p_i^n \delta_n^j(k). \tag{40}$$

Since  $\delta_n^j \geq 0$  and  $p_i^n \geq 0$  for  $i, j, n = \{1, \dots, m\}$ ,  $w_i^j(k+1) \geq \mu(k) = \min_{j \in [m]} \min_{i \in [m]} w_i^j(k)$  for  $i, j = \{1, \dots, m\}$  and all  $k$ . This establishes that the minimum of  $W(k)$  is non-decreasing and other agents cannot go below the previous minimum at the next time step.

Therefore, since (8) defines  $A(k)$ , the smallest non-zero element of  $A(k)$ , denoted  $\min_{i \in [m]} \min_{j \in [m]} [A(k)]_i^j$ , is at least  $\min_{i \in [m]} \min_{j \in [m]} w_i^j(k)$ . This directly implies that the lower bound can be set as  $\min_{j \in [m]} \min_{i \in [m]} w_i^j(0)$ .  $\square$

*Proof of Lemma 4.6*

From Lemma 4.5 and since  $x^i(k+1) = P_{X_i}[v^i(k) - \alpha_k d_i(k)]$ , we have

$$\begin{aligned} \|x^i(k+1) - z\|^2 &\leq \|v^i(k) - \alpha_k d_i(k) - z\|^2 - \\ &\quad \|P_{X_i}[v^i(k) - \alpha_k d_i(k)] - v^i(k) - \alpha_k d_i(k)\|^2. \end{aligned}$$

From the definition of  $\phi^i(k)$  in (12), the previous relation becomes,

$$\|x^i(k+1) - z\|^2 \leq \|v^i(k) - \alpha_k d_i(k) - z\|^2 - \|\phi^i(k)\|^2.$$

By expanding  $\|v^i(k) - \alpha_k d_i(k) - z\|^2$ , we have

$$\begin{aligned} \|v^i(k) - \alpha_k d_i(k) - z\|^2 &= \|v^i(k) - z\|^2 + \alpha_k^2 \|d_i(k)\|^2 \\ &\quad - 2\alpha_k d_i(k)'(v^i(k) - z). \end{aligned} \tag{41}$$

Because  $d_i(k)$  is the gradient of  $f_i(x)$  at  $x = v^i(k)$ , we obtain from convexity that

$$d_i(k)'(v^i(k) - z) \geq f_i(v^i(k)) - f_i(z). \tag{42}$$

By bringing together (41) and (42), we get

$$\begin{aligned} \|x^i(k+1) - z\|^2 &\leq \|v^i(k) - z\|^2 + \alpha_k^2 \|d_i(k)\|^2 \\ &\quad - 2\alpha_k [f_i(v^i(k)) - f_i(z)] - \|\phi^i(k)\|^2. \end{aligned} \tag{43}$$

Given the definition of  $v^i(k)$ , using the convexity of the norm squared function and the stochasticity of the  $a^i(w^i(k))$ , we find that

$$\|v^i(k) - z\|^2 \leq \sum_{j=1}^m a_j^i(w^i(k)) \|x^j(k) - z\|^2. \tag{44}$$

It then follows from (43) and (44) that

$$\begin{aligned} \|x^i(k+1) - z\|^2 &= \sum_{j=1}^m a_j^i(w^i(k)) \|x^j(k) - z\|^2 + \alpha_k^2 \|d_i(k)\|^2 \\ &\quad - 2\alpha_k [f_i(v^i(k)) - f_i(z)] - \|\phi^i(k)\|^2. \end{aligned} \tag{45}$$

By summing (45) over  $i = 1, \dots, m$ , we obtain the desired relation:

$$\begin{aligned} \sum_{i=1}^m \|x^i(k+1) - z\|^2 &\leq \sum_{i=1}^m \sum_{j=1}^m a_j^i(w^i(k)) \|x^j(k) - z\|^2 + \\ \alpha_k^2 \sum_{i=1}^m \|d_i(k)\|^2 - 2\alpha_k \sum_{i=1}^m [f_i(v^i(k)) - f_i(z)] - \sum_{i=1}^m \|\phi^i(k)\|^2. \end{aligned}$$

□

*Proof of Lemma 4.7*

(a) From (13), we have,

$$\begin{aligned} x^i(k) &= \sum_{j=1}^m [\Phi(k-1, s)]_j^i x^j(s) - \sum_{r=s}^{k-2} \sum_{j=1}^m [\Phi(k-1, r+1)]_j^i \alpha_r d_j(r) - \\ &\quad \alpha_{k-1} d_i(k-1) + \sum_{r=s}^{k-2} \sum_{j=1}^m [\Phi(k-1, r+1)]_j^i \phi^j(r) + \phi^i(k-1). \end{aligned} \quad (46)$$

Using the following transition matrices

$$\Phi(k, s) = A(W(k))A(W(k-1)), \dots, A(W(s)) \quad (47)$$

and following the same logic to obtain (13) [6], (16) can be re-written for all  $k$  and  $s$  with  $k > s$  as,

$$\begin{aligned} y(k) &= \frac{1}{m} \sum_{i=1}^m \sum_{j=1}^m [\Phi(k-1, s)]_j^i x^j(s) - \frac{1}{m} \sum_{i=1}^m \sum_{r=s}^{k-2} \sum_{j=1}^m [\Phi(k-1, r+1)]_j^i \alpha_r d_i(r) + \\ &\quad \frac{1}{m} \sum_{i=1}^m \sum_{r=s}^{k-2} \sum_{j=1}^m [\Phi(k-1, r+1)]_j^i \phi^i(r) - \frac{\alpha_k}{m} \sum_{i=1}^m d_i(k-1) + \frac{1}{m} \sum_{i=1}^m \phi^i(k-1). \end{aligned} \quad (48)$$

By subtracting (48) from (46), we obtain,

$$\begin{aligned} x^i(k) - y(k) &= \sum_{j=1}^m [\Phi(k-1, s)]_j^i x^j(s) - \frac{1}{m} \sum_{i=1}^m \sum_{j=1}^m [\Phi(k-1, s)]_j^i x^j(s) \\ &\quad - \sum_{r=s}^{k-2} \sum_{j=1}^m [\Phi(k-1, r+1)]_j^i \alpha_r d_j(r) + \frac{1}{m} \sum_{i=1}^m \sum_{r=s}^{k-2} \sum_{j=1}^m [\Phi(k-1, r+1)]_j^i \alpha_r d_j(r) \\ &\quad - \alpha_{k-1} d_i(k-1) + \frac{\alpha_{k-1}}{m} \sum_{i=1}^m d_i(k-1) + \sum_{r=s}^{k-2} \sum_{j=1}^m [\Phi(k-1, r+1)]_j^i \phi^j(r) \\ &\quad - \frac{1}{m} \sum_{i=1}^m \sum_{r=s}^{k-2} \sum_{j=1}^m [\Phi(k-1, r+1)]_j^i \phi^j(r) + \phi^i(k-1) - \frac{1}{m} \sum_{i=1}^m \phi^i(k-1). \end{aligned} \quad (49)$$



Taking the norm of (49), we get

$$\begin{aligned}
\|x^i(k) - y(k)\| &\leq \sum_{j=1}^m \left\| [\Phi(k-1, s)]_j^i - \frac{1}{m} \sum_{i=1}^m [\Phi(k-1, s)]_j^i \right\| \|x^j(s)\| \\
&+ \sum_{r=s}^{k-2} \sum_{j=1}^m \left\| [\Phi(k-1, r+1)]_j^i - \frac{1}{m} \sum_{i=1}^m [\Phi(k-1, r+1)]_j^i \right\| \alpha_r \|d_j(r)\| \\
&\quad + \alpha_{k-1} \|d_i(k-1)\| + \frac{\alpha_{k-1}}{m} \sum_{i=1}^m \|d_i(k-1)\| \\
&+ \sum_{r=s}^{k-2} \sum_{j=1}^m \left\| [\Phi(k-1, r+1)]_j^i - \frac{1}{m} \sum_{i=1}^m [\Phi(k-1, r+1)]_j^i \right\| \|\phi^j(r)\| \\
&\quad + \|\phi^i(k-1)\| + \frac{1}{m} \sum_{i=1}^m \|\phi^i(k-1)\|.
\end{aligned} \tag{50}$$

Using Lemma 4.3 and for  $s = 0$ , and  $k \rightarrow \infty$ , the first right-hand term of (50) is

$$\begin{aligned}
&\|x^i(k) - y(k)\| \leq \\
&\sum_{j=1}^m \left\| [\Phi(k-1, 0)]_j^i - \gamma_j(0) \right\| + \left\| \frac{1}{m} \sum_{i=1}^m [\Phi(k-1, 0)]_j^i - \gamma_j(0) \right\| \|x^j(0)\| \\
&+ \sum_{r=0}^{k-2} \sum_{j=1}^m \left\| [\Phi(k-1, r+1)]_j^i - \frac{1}{m} \sum_{i=1}^m [\Phi(k-1, r+1)]_j^i \right\| \alpha_r \|d_j(r)\| \\
&\quad + \alpha_{k-1} \|d_i(k-1)\| + \frac{\alpha_{k-1}}{m} \sum_{i=1}^m \|d_i(k-1)\| \\
&+ \sum_{r=0}^{k-2} \sum_{j=1}^m \left\| [\Phi(k-1, r+1)]_j^i - \frac{1}{m} \sum_{i=1}^m [\Phi(k-1, r+1)]_j^i \right\| \|\phi^j(r)\| \\
&\quad + \|\phi^i(k-1)\| + \frac{1}{m} \sum_{i=1}^m \|\phi^i(k-1)\|,
\end{aligned} \tag{51}$$

which can be simplified as,

$$\begin{aligned}
& \|x^i(k) - y(k)\| \leq 2mC\beta^{k-1} \sum_{j=1}^m \|x^j(0)\| \\
& + \sum_{r=0}^{k-2} \sum_{j=1}^m \left[ \left| [\Phi(k-1, r+1)]_j^i - \frac{1}{m} \sum_{i=1}^m [\Phi(k-1, r+1)]_j^i \right| \alpha_r \|d_j(r)\| \right] \\
& + \alpha_{k-1} \|d_i(k-1)\| + \frac{\alpha_{k-1}}{m} \sum_{i=1}^m \|d_i(k-1)\| \\
& + \sum_{r=0}^{k-2} \sum_{j=1}^m \left[ \left| [\Phi(k-1, r+1)]_j^i - \frac{1}{m} \sum_{i=1}^m [\Phi(k-1, r+1)]_j^i \right| \|\phi^j(r)\| \right] \\
& + \|\phi^i(k-1)\| + \frac{1}{m} \sum_{i=1}^m \|\phi^i(k-1)\|.
\end{aligned} \tag{52}$$

Similarly, using Lemma 4.3, the second right-hand term is

$$\begin{aligned}
& \|x^i(k) - y(k)\| \leq 2mC\beta^{k-1} \sum_{j=1}^m \|x^j(0)\| + 2mCL \sum_{r=0}^{k-2} \beta^{k-r} \alpha_r \\
& + \alpha_{k-1} \|d_i(k-1)\| + \frac{\alpha_{k-1}}{m} \sum_{i=1}^m \|d_i(k-1)\| \\
& + \sum_{r=0}^{k-2} \sum_{j=1}^m \left[ \left| [\Phi(k-1, r+1)]_j^i - \frac{1}{m} \sum_{i=1}^m [\Phi(k-1, r+1)]_j^i \right| \|\phi^j(r)\| \right] \\
& + \|\phi^i(k-1)\| + \frac{1}{m} \sum_{i=1}^m \|\phi^i(k-1)\|.
\end{aligned} \tag{53}$$

Using Lemma 4.2 and the gradient bound, the third-hand right term is

$$\begin{aligned}
& \|x^i(k) - y(k)\| \leq 2mC\beta^{k-1} \sum_{j=1}^m \|x^j(0)\| + 2mCL \sum_{r=0}^{k-2} \beta^{k-r} \alpha_r + 2\alpha_{k-1}L \\
& + \sum_{r=0}^{k-2} \sum_{j=1}^m \left[ \left| [\Phi(k-1, r+1)]_j^i - \frac{1}{m} \sum_{i=1}^m [\Phi(k-1, r+1)]_j^i \right| \|\phi^j(r)\| \right] \\
& + \|\phi^i(k-1)\| + \frac{1}{m} \sum_{i=1}^m \|\phi^i(k-1)\|.
\end{aligned} \tag{54}$$

Using again Lemma 4.2 and 4.3, we obtain for the last two terms,

$$\begin{aligned}
& \|x^i(k) - y(k)\| \leq 2mC\beta^{k-1} \sum_{j=1}^m \|x^j(0)\| + 2mCL \sum_{r=0}^{k-2} \beta^{k-r} \alpha_r + 2\alpha_{k-1}L \\
& + 2mCL \sum_{r=0}^{k-2} \beta^{k-r} \alpha_r + 2\alpha_{k-1}L.
\end{aligned} \tag{55}$$

We therefore obtain,

$$||x^i(k) - y(k)|| \leq 2mC\beta^{k-1} \sum_{j=1}^m ||x^j(0)|| + 4mCL \sum_{r=0}^{k-2} \beta^{k-r} \alpha_r + 4\alpha_{k-1}L. \quad (56)$$

Since  $0 < \beta < 1$ ,  $\beta^k \rightarrow 0$  as  $k \rightarrow \infty$ . Assuming that  $\alpha_k \rightarrow 0$  and taking the limit superior, we have for all  $i$ ,

$$\limsup_{k \rightarrow \infty} ||x^i(k) - y(k)|| \leq 4mCL \limsup_{k \rightarrow \infty} \sum_{r=0}^{k-2} \beta^{k-r} \alpha_r. \quad (57)$$

By Lemma 4.4, we have

$$\lim_{k \rightarrow \infty} \sum_{r=0}^{k-2} \beta^{k-r} \alpha_r = 0.$$

Therefore,  $\lim_{k \rightarrow \infty} ||x^i(k) - y(k)|| = 0$  for all  $i$ .

(b) By multiplying (56) with  $\alpha_k$ , we get

$$\alpha_k ||x^i(k) - y(k)|| \leq 2mC\alpha_k\beta^{k-1} \sum_{j=1}^m ||x^j(0)|| + 4mCL \sum_{r=0}^{k-2} \beta^{k-r} \alpha_k \alpha_r + 4\alpha_k \alpha_{k-1}L.$$

Using  $2\alpha_k \alpha_r \leq \alpha_k^2 + \alpha_r^2$  and  $\alpha_k \beta^{k-1} \leq \alpha_k^2 + \beta^{2(k-1)}$  for any  $k$  and  $r$ , we obtain

$$\begin{aligned} \alpha_k ||x^i(k) - y(k)|| &\leq 2mC\beta^{2(k-1)} \sum_{j=1}^m ||x^j(0)|| + 2mC\alpha_k^2 \sum_{j=1}^m ||x^j(0)|| \\ &\quad + 2mCL\alpha_k^2 \sum_{r=0}^{k-2} \beta^{k-r} + 2mCL \sum_{r=0}^{k-2} \beta^{k-r} \alpha_r^2 + 2L(\alpha_k^2 + \alpha_{k-1}^2). \end{aligned}$$

Since  $\sum_{r=0}^{k-2} \beta^{k-r} \leq \frac{1}{1-\beta}$ , we have

$$\begin{aligned} \alpha_k ||x^i(k) - y(k)|| &\leq 2mC\beta^{2(k-1)} \sum_{j=1}^m ||x^j(0)|| + 2mC\alpha_k^2 \sum_{j=1}^m ||x^j(0)|| \\ &\quad + 2mCL\alpha_k^2 \frac{1}{1-\beta} + 2mCL \sum_{r=0}^{k-2} \beta^{k-r} \alpha_r^2 + 2L(\alpha_k^2 + \alpha_{k-1}^2). \end{aligned}$$

By summing from  $k = 1$  to  $k = \infty$ , we obtain

$$\begin{aligned}
\sum_{k=1}^{\infty} \alpha_k \|x^i(k) - y(k)\| &\leq 2mC \sum_{k=1}^{\infty} \beta^{2(k-1)} \sum_{j=1}^m \|x^j(0)\| + \\
2mC \sum_{k=1}^{\infty} \alpha_k^2 \sum_{j=1}^m \|x^j(0)\| &+ 2mCL \frac{1}{1-\beta} \sum_{k=1}^{\infty} \alpha_k^2 + \\
2mCL \sum_{k=1}^{\infty} \sum_{r=0}^{k-2} \beta^{k-r} \alpha_r^2 &+ 2L \sum_{k=1}^{\infty} (\alpha_k^2 + \alpha_{k-1}^2).
\end{aligned} \tag{58}$$

In (58), the first term is summable since  $0 < \beta < 1$ . The second and third, and fifth terms are also summable since  $\sum_{k \rightarrow \infty} \alpha_k^2 < \infty$ . By Lemma 4.4, the fourth term is summable. Thus,  $\sum_{k=1}^{\infty} \alpha_k \|x^i(k) - y(k)\| < \infty$  for all  $i$ .  $\square$

## References

1. J. Qin, Q. Ma, Y Shi and L. Wang, *Recent advances in consensus of multi-agent systems: A brief survey*. IEEE Transactions on Industrial Electronics. 2016 Dec 7;64(6):4972-83.
2. A. Filotheou, A. Nikou and D. V. Dimarogonas, *Decentralized Control of Uncertain Multi-Agent Systems with Connectivity Maintenance and Collision Avoidance*, 2018 European Control Conference (ECC), Limassol, 2018, pp. 8-13.
3. X. Wang, H. Su, X. Wang, and G. Chen, *An overview of coordinated control for multi-agent systems subject to input saturation*, Perspectives in Science. 2016 Mar 1;7:133-9.
4. Oh KK, Park MC, Ahn HS. A survey of multi-agent formation control. Automatica. 2015 Mar 1;53:424-40.
5. Nedić, Angelia, Asuman Ozdaglar, and Pablo A. Parrilo. *Constrained consensus*, arXiv preprint arXiv:0802.3922 (2008).
6. A. Nedić and A. Ozdaglar, *Distributed subgradient methods for multi-agent optimization*, IEEE Transactions on Automatic Control, 54(1), 48, 2009.
7. A. Nedic and A. Ozdaglar, 2010. *10 cooperative distributed multi-agent*, Convex Optimization in Signal Processing and Communications, 340.
8. A. Nedic, A. Ozdaglar and P.A. Parrilo, *Constrained consensus and optimization in multi-agent networks*. IEEE Transactions on Automatic Control, 55(4), 922-938, 2010.

9. K.I. Tsianos, S. Lawlor and M.G. Rabbat, Consensus-based distributed optimization: Practical issues and applications in large-scale machine learning. In 2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton) 2012 Oct 1 (pp. 1543-1550). IEEE.
10. J.C. Duchi, A. Agarwal and M.J. Wainwright, Dual averaging for distributed optimization: Convergence analysis and network scaling. IEEE Transactions on Automatic control. 2011 Jun 30;57(3):592-606.
11. Wang J, Elia N. Control approach to distributed optimization. In 2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton) 2010 Sep (pp. 557-561). IEEE.
12. Agarwal A, Duchi JC. Distributed delayed stochastic optimization. In Advances in Neural Information Processing Systems 2011 (pp. 873-881).
13. Liu Q, Wang J. A second-order multi-agent network for bound-constrained distributed optimization. IEEE Transactions on Automatic Control. 2015 Mar 27;60(12):3310-5.
14. Y. Zhang, Y. Lou and Y. Hong, An approximate gradient algorithm for constrained distributed convex optimization. IEEE/CAA Journal of Automatica Sinica 1.1, 61-67, 2014.
15. B. Touri, and A. Nedic, On backward product of stochastic matrices. Automatica 48.8: 1477-1488, 2018.
16. S.S. Ram, A. Nedić and V.V. Veeravalli, 2010. Distributed stochastic subgradient projection algorithms for convex optimization. Journal of optimization theory and applications, 147(3), pp.516-545.
17. P. Bianchi, G. Fort, W. Hachem and J. Jakubowicz, "Performance analysis of a distributed Robbins-Monro algorithm for sensor networks." In 2011 19th European Signal Processing Conference, pp. 1030-1034. IEEE, 2011.
18. I. Lobel, A. Ozdaglar and D. Feijer, (2011). *Distributed multi-agent optimization with state-dependent communication*, Mathematical programming, 129(2), 255-284.
19. A. Nedic and D.P. Bertsekas, *Incremental subgradient methods for nondifferentiable optimization*. SIAM Journal on Optimization, 2001;12(1):109-38.

- 
20. Byungchul Kim and O. Lavrova, "Optimal power flow and energy-sharing among multi-agent smart buildings in the smart grid," 2013 IEEE Energytech, Cleveland, OH, 2013, pp. 1-5, doi: 10.1109/EnergyTech.2013.6645336.
  21. K. M. Miettinen, *Nonlinear Multiobjective Optimization*, Kluwer Academic Publishers, 1999.
  22. Y. Collette and P. Siarry, *Multiobjective Optimization: Principles and Case Studies*, Springer, 2004.
  23. L. Xiao, S. Boyd and S.J. Kim, *Distributed average consensus with least-mean-square deviation*, Journal of parallel and distributed computing, 67(1), 33-46, 2007.
  24. A. Olshevsky and J.N. Tsitsiklis, *Convergence speed in distributed consensus and averaging*, SIAM review 53.4: 747-772, 2011.
  25. V.D. Blondel, J.M. Hendrickx, A. Olshevsky and J.N. Tsitsiklis, *Convergence in multi-agent coordination, consensus, and flocking*, In Proceedings of the 44th IEEE Conference on Decision and Control, pp. 2996-3000, 2005.
  26. Blondin, Maude J., and Matthew Hale. *An Algorithm for Multi-Objective Multi-Agent Optimization*, American Control Conference, 2020.
  27. R. Olfati-Saber, J. A. Fax and R. M. Murray, "Consensus and Cooperation in Networked Multi-Agent Systems," in Proceedings of the IEEE, vol. 95(1), pp. 215-233, 2007.
  28. S. Khim, "The Frobenius-Perron theorem," 2007.