Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 14, No. 2, pp. 307–326, 2018

DISTRIBUTED ALGORITHMS FOR COMPUTING THE MINIMUM ERROR OF ε-OPTIMAL SOLUTIONS IN MULTI-OBJECTIVE OPTIMIZATION*

Youcheng Lou and Shouyang Wang

Abstract: Almost all of the existing works on multi-objective optimization assume that the given multiple objectives have no common optimal solution and then focus on how to design algorithms to find Pareto optimal solutions. In this paper, we consider the problem of how to check whether the given multiple objectives have common optimal solutions and the more general ε-optimal solutions in a fully distributed manner. We first equivalently convert the minimum error problem into a min-max problem and then employ one sum objective optimization problem to approximately approach this min-max problem. In this paper, we solve the sum objective optimization problem using a multi-agent network, where each objective is uniquely associated with an agent and agents communicate with their neighbors over an underlying directed graph. We propose a distributed algorithm to solve the sum objective optimization problem based on the well-known distributed subgradient algorithm and the distributed averaging algorithm. We first establish the optimal convergence of the sum objective function at agents' estimates, and then provide an estimate for the minimum error that guarantees the existence of ε-optimal solutions. Moreover, we also provide an estimate on the convergence of a class of weight-balanced adjacency matrix generating algorithms.

Key words: *multi-objective optimization, distributed algorithms, common optimal solutions, ε-optimal solutions*

Mathematics Subject Classification: *90C29, 68W15*

1 Introduction

In multi-objective optimization, serval objectives have to be minimized simultaneously. Since the objectives are usually conflicting, there is no feasible solution that is optimal for all objectives. Instead, the decision-maker needs to make a trade-off among these objectives within the well-known Pareto optimal solution (POS) set or the nondominated set in the outcome space [1, 4, 9, 32]. In the last decades, much attention has been paid to designing various algorithms to seek a POS, the entire POS set or a discrete representation of the POS set [2, 7, 10, 16, 17, 21–23, 29]. An extensive survey about the approximate methods for a discrete representation of the POS set or the nondominated set can be found in [28].

Almost all of the existing works on multi-objective optimization have a common presupposition: the objectives have no common optimal solution (COS). However, for given multiple objectives, how to check whether they have a COS? This is an important problem since if these objectives have a COS, this COS is good and desirable for the decision-maker,

*This work is supported by China Postdoctoral Science Foundation under Grant 2014M550098 and National Natural Science Foundation of China under Grant 71401163.

and if not, as the existing works suggest, the decision-maker needs to seek optimal solutions in other senses, for example, Pareto optimality. To the best of our knowledge, there are no papers to consider this problem. In this paper, we will investigate whether the given objectives have a COS. In fact, we solve this problem by considering a more general problem: how to compute the minimum error ε that guarantees that the objectives have at least one ε-optimal solution, noting that the objectives have a COS if and only if the minimum error guaranteeing the existence of ε-optimal solutions is zero. The aim of this paper is to provide an estimate for the minimum error that guarantees the existence of ε-optimal solutions in a fully distributed manner. We solve the minimum error estimate problem through a multi-agent network, where each agent uniquely corresponds to an objective and each agent only knows its own objective. In this network, agents can exchange information only with their neighbors via an underlying directed graph.

Distributed optimization and algorithms have been widely studied in the literature due to its broad applications in resource allocation and wireless sensor networks [3, 5, 6, 8, 12, 15, 18–20, 25–27, 30, 34]. In centralized algorithms, there is typically a fusion center that takes charge of collecting, analyzing and computing the data received from the agents. However, in contrast to this, in distributed algorithms there is no fusion center and the agents accomplish the task by mutual cooperation. We take the following widely studied resource allocation problem for example to illustrate the motivation of applying distributed algorithms to solve the considered problem. Consider a large factory consisting of multiple divisions producing different products (each division corresponds to an agent in the network). Each division manager only knows her own division's production plan and cost objective, but not other divisions' information. When the number of the divisions is very large, the workload of the central resource planner or the factory manager (playing the role of the fusion center in centralized algorithms) and the data transmitted from the divisions to the central resource planner are very huge. This centralized way may cause network congestion and even breakdown of this network. In this case, a possible approach is that all the division managers cooperate with each other to help the central resource planner accomplish the resource allocation task. Distributed algorithms enjoy the following advantages over centralized algorithms: scalability to the network size, robustness to communication failure, avoiding computation overload and network congestion.

We develop a fully distributed method to solve the minimum error estimation problem based on the proposed distributed subgradient algorithm and the well-known distributed averaging algorithm. The considered problem can be equivalently converted into a min-max optimization problem. The authors in [31] proposed two distributed methods (penalty-based method and primal-dual method) to solve general min-max problems. From a different viewpoint, in this paper we use a standard sum objective optimization problem (SOOP) to approximate the equivalently transformed min-max problem and propose a distributed subgradient algorithm to solve this standard SOOP. We first establish the optimal convergence of the distributed subgradient algorithm in terms of the sum objective at agents' estimates. Then we provide an estimate for the minimum error that guarantees the existence of ε-optimal solutions. The contribution of this paper is summarized as follows.

- To our knowledge, it is the first time that the problem whether the objectives in multi-objective optimization have COSs is investigated. Almost all of the works on multi-objective optimization assume that the given objectives have no COS. However, how to check whether the given objectives have a COS is an important problem in practice, since if they have a COS, this solution is desirable and good for all the objectives, and in this case it is no longer necessary to apply the existing methods to find POSs.

- In terms of the new concept of ε-optimal solutions, we consider a more general minimum error estimation problem in a fully distributed manner. Based on the proposed distributed subgradient algorithm and distributed averaging algorithm, we present an estimate for the minimum error that guarantees the existence of ε-optimal solutions.
- The weight-balance of the adjacency matrix plays an important role in the optimal convergence of distributed subgradient optimization algorithms [19]. The authors in [11] proposed a distributed mirror imbalance-correcting algorithm and established its finite-time convergence to a weight-balanced adjacency matrix and convergence time complexity analysis. In this paper, we provide an estimate on the convergence of a class of more general weight-balanced adjacency matrix generating algorithms which are the same as the mirror imbalance-correcting algorithm in [19] except without specifying a specific node selection rule.

This paper is organized as follows. In Section 2, we introduce the notation and the two concepts of common optimal solution and ε-optimal solution. In Section 3, we first review the distributed optimization algorithms in the literature and then introduce the proposed distributed subgradient algorithm following with an algorithm convergence result. In Section 4, we provide an estimate for the minimum error that guarantees the existence of ε-optimal solutions, while in Section 5, we give an estimate for the convergence of a class of weight-balanced matrix generating algorithms. Finally, some concluding remarks are given in Section 6.

2 Preliminaries and Problem Formulation

In this section, we formulate the minimum error estimation problem of interest. We first introduce the notation and basic terminology used in this paper.

2.1 Notation and Terminology

Throughout this paper, R is the real Euclidean space with the Euclidean norm $||\cdot||$. We view a vector as a column vector and write z' to denote the transpose of a vector or matrix z. We write $P_X(\cdot)$ to denote the projection operator onto closed convex set X. For a convex function $g : \mathrm{R}^m \to \mathrm{R}$, $v(\hat{z}) \in \mathrm{R}^m$ is a subgradient of g at vector $\hat{z}$ if $g(z) \geq g(\hat{z}) + (z-\hat{z})'v(\hat{z})$, $\forall z \in \mathrm{R}^m$. The set of all subgradients of g at $\hat{z}$ is denoted by $\partial g(\hat{z})$.

A vector is said to be a weighting vector if all its components are nonnegative and the sum of its components is one. A matrix is a weighting matrix if all of its row vectors are weighting vectors. Furthermore, if the transpose of a weighting matrix is also weighting, we call this weighting matrix a doubly weighting matrix. Clearly, for doubly weighting matrix A, we have $A\mathbf{1} = \mathbf{1}$, $\mathbf{1}'A = \mathbf{1}'$, where $\mathbf{1}$ is the vector with all ones. Let $A = (a_{ij}) \in \mathrm{R}^{n\times n}$ be a nonnegative matrix. Denote by $\mathcal{G}_A = \big(\mathcal{V}, \mathcal{E}_A\big)$ the induced graph of A, i.e., $(j,i) \in \mathcal{E}_A$ if and only if $a_{ij} > 0$, $\mathcal{V} = \{1, 2, ..., n\}$. Graph $\mathcal{G}_A$ is strongly connected if there is a path in $\mathcal{G}_A$ from i to j for each pair node i, j.

Let A be a weighting matrix with strongly connected induced graph $\mathcal{G}_A$. By Proposition 1 in [26], it holds that for any k,

$$\Big|\Big|A^k e_i - \frac{1}{n}\mathbf{1}\Big|\Big| \leq c_0\sigma^k, \tag{2.1}$$

where $c_0 = 2\sqrt{n}\frac{1+\zeta^{-n+1}}{(1-\zeta^{n-1})^{1+\frac{1}{n-1}}}$, $0 < \sigma = (1-\zeta^{n-1})^{\frac{1}{n-1}} < 1$, $\zeta = \min\{a_{ij}|a_{ij} > 0, i, j \in \mathcal{V}\}$, e_i is the vector with i-th component one and other components zero.

2.2 Common Optimal Solutions and ε-Optimal Solutions

A multi-objective optimization problem (MOOP) is usually defined as follows:

$$\begin{aligned} &\text{minimize } \ f(x) = \big(f_1(x), \ldots, f_n(x)\big) \\ &\text{subject to } \ x \in X. \end{aligned} \tag{2.2}$$

Here $X \subseteq \mathrm{R}^m$ is the convex constraint set, which is assumed to be bounded; $f_i : \mathrm{R}^m \to \mathrm{R}, i = 1, ..., n$ are the convex objective functions to be minimized simultaneously. By the boundedness of X and the convexity of f_i, $f_i^* := \min_{x\in X} f_i(x)$ is a finite number, $i = 1, ..., n$ and the minimum $\min_X \frac{1}{n}\sum_{i=1}^n f_i$ can be achieved.

When the objectives are conflicting, there exists no well-defined solution that can minimize all objectives simultaneously but some mathematically equally good solutions, which are called the POSs or nondominated points in the outcome space [1, 4, 9, 32]. Much attention has been paid to designing algorithms to seek the POSs. However, to our knowledge there is no paper investigating whether the given objectives are conflicting, or equivalently, whether they have COSs. We now introduce the formal definition of COSs.

Definition 2.1. (Common Optimal Solutions) We say that MOOP (2.2) has a COS if there exists a feasible point $x_0 \in X$ such that

$$f_i(x_0) = f_i^*, \ i = 1, ..., n.$$

In this paper, we are interested in the problem that how to check the given objectives have a COS. Although in most cases these objectives do not have COSs, the COS existence problem is still important in practice since if these objectives have a COS, this solution is desirable and it is no longer necessary to apply the existing methods to seek POSs. When the objectives have no COS, sometimes the feasible point with a certain error from all the optimal values f_i^* can also be viewed as an acceptable solution. Here is the definition of the newly proposed concept of ε-optimality.

Definition 2.2. (ε-Optimal Solutions) We say that MOOP (2.2) has an ε-optimal solution if there exists a feasible point $x_0 \in X$ such that

$$f_i(x_0) \le f_i^* + \varepsilon, \ i = 1, ..., n.$$

Here x_0 is said to be an ε-optimal solution of MOOP (2.2) and $\varepsilon \ge 0$ is referred to as the (approximate) error of x_0.

Clearly, the concept of ε-optimality is more general than the concept of common optimal solutions since a COS is a 0-optimal solution, and when the specified error ε is sufficiently large, there always exist ε-optimal solutions with ε as the approximate error.

2.3 ε-Pareto Optimal Solutions

Various concepts of ε-POSs (or ε-efficient solutions) have been proposed in the literature [13, 33]. We next introduce one of the most commonly used concepts and give some discussions on this concept and the ε-optimality proposed in this paper.

(ε-Pareto Optimal Solutions): Let $\varepsilon > 0$. Point $x_0 \in X$ is said to be an ε-POS of MOOP (2.2) if there is no $x \in X$ such that $f_i(x) \leq f_i(x_0) - \varepsilon$ for all i and with strict inequality holding for at least one index i.

We can see that ε-optimality is stronger than the ε-Pareto optimality. In fact, if $x_0 \in X$ is an ε-optimal solution, but not an ε-POS, then there exists $x \in X$ such that $f_i(x) \leq f_i(x_0) - \varepsilon$ for all i and with strict inequality holding for some index i_0. Combining this with the ε-optimality, we have $f_{i_0}(x) < f_{i_0}(x_0) - \varepsilon \leq f_{i_0}^*$, which yields a contradiction. An ε-optimal solution is thus also an ε-POS.

Now we give some discussions about the two concepts. First, since a POS is also an ε-POS, generally the ε-POS set is nonempty. We also note that a MOOP does not have ε-optimal solutions for all sufficiently small ε unless it has a COS. However, an advantage of ε-optimality over ε-Pareto optimality is that in some cases it excludes some solutions, which may be good for some objectives, but extremely bad for some other objectives, from the set of acceptable solutions. Here we take an example to illustrate this observation.

Example 2.3. We consider a bi-objective piecewise linear optimization problem with objectives:

$$f_1(x) = \begin{cases} -ax + 2 - 3a, & -\infty < x \leq -3; \\ -x - 1, & -3 < x \leq -1; \\ x + 1, & -1 < x \leq 1; \\ ax + 2 - a, & 1 < x < +\infty, \end{cases} \quad f_2(x) = \begin{cases} -ax + 2 - a, & -\infty < x \leq -1; \\ -x + 1, & -1 < x \leq 1; \\ x - 1, & 1 < x \leq 3; \\ ax + 2 - 3a, & 3 < x < +\infty, \end{cases}$$

where $a > 1$, the constraint set is $X = [-3, 3]$. We can see that the 2-optimal solution set is $[-1, 1]$ and 2-POS set is $(-3, 3)$. Although the points in $(-3, -1) \cup (1, 3)$ are acceptable in the sense of approximate Pareto optimality, intuitively they are not acceptable in the sense that though points in $(-3, -1)$ are relatively good for the first objective, they are extremely bad, especially those close to -3, for the second objective when $a > 1$ is sufficiently large. Similarly, points in $(1, 3)$ are approximately good for the second objective, but extremely bad for the first objective when $a > 1$ is sufficiently large.

2.4 Problem Formulation

In this paper we are interested in the following two problems:

Does MOOP (2.2) has a COS?

How to compute the minimum ε that can guarantee the existence of ε-optimal solutions?

In fact, we can give a positive answer to the first problem by solving the second more general one. The second problem can be described as the following optimization problem:

$$\begin{aligned} &\text{minimize } \ \varepsilon \\ &\text{subject to } \ x \in X, \ f_i(x) - f_i^* \leq \varepsilon, i = 1, ..., n, \end{aligned}$$

which can be further equivalently converted into

$$\min_{x \in X} \max_{1 \leq i \leq n} (f_i(x) - f_i^*) \triangleq M. \tag{2.3}$$

In this paper, M is referred to as the minimum error that guarantees the existence of ε-optimal solutions and we will develop a fully distributed method to give an approximate estimate for it. The following lemma is straightforward.

Lemma 2.4. *MOOP (2.2) has an ε-optimal solution if and only if $\varepsilon \geq M$. Particularly, MOOP (2.2) has a COS if and only if $M = 0$.*

3 Distributed Subgradient Algorithms

In this section, we will first review some existing distributed optimization algorithms and then propose a new distributed subgradient algorithm. Finally, we give a preliminary convergence result for the proposed algorithm.

3.1 Literature Review on Distributed Optimization Algorithms

In this subsection, we first review some distributed optimization algorithms and then present some comparisons between distributed algorithms and centralized algorithms.

Due to the broad applications in resource allocation, machine learning and wireless sensor networks, distributed optimization and distributed algorithms have been widely studied in the control, engineering and optimization community [3,5,6,8,12,15,18–20,25–27,30]. Most of attention has been paid to designing distributed algorithms for a multi-agent network to cooperatively minimize the SOOP

$$\min_{x\in\mathbb{R}^m} \sum_{i=1}^{n} g_i(x),$$

where each component g_i of the sum objective is only known by agent i. The network can achieve a global optimization goal via agents' local optimization and local information exchange among agents over a directed graph. The authors in [25] proposed a distributed subgradient algorithm

$$x_i(k+1) = \sum_{j\in\mathcal{N}_i(k)} a_{ij}(k)x_j(k) - \alpha d_i(k), i = 1, ..., n$$

to solve the above SOOP, where $x_i(k)$ is the estimate of agent i for the optimal solution of the sum objective at time k, $\mathcal{N}_i(k)$ is agent i's neighboring node set at time k in the time-varying network graphs, which are used to characterize the communication structure among agents, $d_i(k)$ is a subgradient of agent i's convex objective g_i at its current estimate $x_i(k)$. Under some basic assumptions of bounded subgradients and jointly connected network graph, the authors in [25] presented a convergence error between the sum objective value at an ergodic average of all past estimates and the optimal objective value in terms of some model parameters including the constant stepsize, the upper bound of subgradients and the number of agents. Then the authors in [26] considered a more general sum objective constrained optimization problem and proposed a similar projection subgradient algorithm with time-varying step-sizes: $x_i(k+1) = P_X\big(\sum_{j\in\mathcal{N}_i(k)} a_{ij}(k)x_j(k) - \alpha_k d_i(k)\big)$, where P_X denotes the convex projection operator. In addition to the bounded subgradients and jointly connected graph assumptions, the authors showed that all agents' estimates will converge to some common optimal point provided that the time-varying stepsizes $\{\alpha_k\}$ satisfies the classical stochastic approximation conditions $\sum_{k=0}^{\infty}\alpha_k = +\infty$ and $\sum_{k=0}^{\infty}\alpha_k^2 < +\infty$. Following this, many distributed subgradient algorithms and their variants emerge to solve the SOOP under different scenarios, for example, inexact subgradients with errors [27], random network graphs [18, 30], unbalanced network graphs [19], incremental subgradient methods [15].

Although the subgradient method is simple and easily implemented, it suffers from low convergence rate. Other distributed algorithms including distributed dual averaging algorithms [8] and ADMM [3] also appeared in the literature to solve the SOOP. Besides the

SOOP, distributed algorithms were also proposed in the literature to solve other problems, for instance, Nash equilibrium computation in games [12, 19], convex intersection problems in optimization theory [20], POS computation in multi-objective optimization [6]. In fact, most of the existing distributed algorithms focus on the SOOPs, except the work [31]. The authors in [31] proposed a distributed exact-penalty approach and a distributed primal-dual Lagrangian approach to solve the general min-max problem, respectively.

In centralized algorithms, typically there is a fusion center, which takes charge of collecting, analyzing and computing the data received from all the agents in the network. However, in practical situations a fusion center is usually not allowed and the communication cost to the fusion center is expensive and not economical. Moreover, the breakdown of the fusion center may cause the failure of the entire optimization task. In contrast to centralized algorithms, in distributed algorithms the fusion center is not required and agents autonomously exchange information with their neighbors at a low communication cost. The convergence results in [5, 25, 26] have shown that a global optimization can be achieved by agents in a cooperative way via agents' local information communication and local optimization for their own objective functions. Compared to centralized algorithms, distributed algorithms also have the advantages of scalability to the network size and robustness to communication failure. In addition, the distributed algorithms can also largely avoid the potential computation overload due to the absence of the fusion center.

3.2 A Distributed Subgradient Algorithm

By the norm relation $||z||_\infty \le ||z||_p \le n^{\frac{1}{p}}||z||_\infty$, $\forall z \in \mathbb{R}^n$ we have

$$0 \le \min_{x\in X}\Big(\sum_{i=1}^{n}(f_i(x)-f_i^*)^p\Big)^{\frac{1}{p}} - \min_{x\in X}\max_{1\le i\le n}(f_i(x)-f_i^*) \le (n^{\frac{1}{p}}-1)\max_{x\in X,1\le i\le n}(f_i(x)-f_i^*)$$

and then it follows from the boundedness of X and the convexity of f_i that

$$\lim_{p\to+\infty}\min_{x\in X}\Big(\sum_{i=1}^{n}(f_i(x)-f_i^*)^p\Big)^{\frac{1}{p}} = \min_{x\in X}\max_{1\le i\le n}(f_i(x)-f_i^*) = M.$$

In the sequel we will present an approximate estimate for M by solving the following SOOP

$$\min_{x\in X}\Big(\sum_{i=1}^{n}(f_i(x)-f_i^*)^p\Big)^{\frac{1}{p}}, \tag{3.1}$$

where p is sufficiently large.

We consider a network consisting of n agents with node set $\mathcal{V}$. Each agent i is associated with the convex objective $(f_i - \hat{f}_i^*)^p$ and agent i does not know other agents' objectives. In our setup, agents cannot obtain the minimum f_i^* of their own objectives and can only use approximate estimates $\hat{f}_i^*$. The goal of this network is to cooperatively solve optimization problem (3.1) by only local information exchange over a directed graph.

We propose the following distributed subgradient optimization algorithm:

$$\begin{cases} z_i(k+1) = P_X(z_i(k)-\alpha_k u_i(k)), \\ w_i(k+1) = f_i(\hat{z}_i(k+1)),\ \hat{z}_i(k+1) = \frac{1}{\sum_{r=0}^{k+1}\alpha_r}\sum_{r=0}^{k+1}\alpha_r z_i(r), \\ x_i(k+1) = P_X\big(\breve{x}_i(k)-\alpha_k p(f_i(\breve{x}_i(k))-w_i(k+1))^{p-1}v_i(k)\big), \\ y_i(k+1) = f_i(\hat{x}_i(k+1)),\ \hat{x}_i(k+1) = \frac{1}{\sum_{r=0}^{k+1}\alpha_r}\sum_{r=0}^{k+1}\alpha_r x_i(r),\ i=1,...,n, \end{cases} \tag{3.2}$$

where $\breve{x}_i(k) = \sum_{j\in\mathcal{N}_i} a_{ij}x_j(k)$, $u_i(k) \in \partial f_i(z_i(k))$, $v_i(k) \in \partial f_i(\breve{x}_i(k))$, $z_i(0) = x_i(0)$, $0 < \alpha_k \leq 1$ is the stepsize at time k. Here the adjacency matrix $A = (a_{ij}) \in \mathrm{R}^{n\times n}$ is used to characterize the communication structure among the agents, and $\mathcal{N}_i = \{j|(j,i) \in \mathcal{E}_A\}$ is the neighbor set of node i.

Remark 3.1. The authors in [31] solved the general min-max problem $\min_X \max_{1\leq i\leq n} g_i$ by first converting it into a SOOP that introduces some new decision variables and then proposed two distributed methods of penalty-based approach and primal-dual Lagrangian approach to solve it. Different from the two methods in [31], here we approximately convert the min-max problem into a standard SOOP involved only with the decision variable and fully employ the existing distributed algorithms to solve this standard SOOP. Note that agents generally cannot obtain the minimum f_i^* of their own objectives. In this paper, we propose a new distributed algorithm to solve the min-max problem (2.3) involved with individual minimum f_i^*, in which the well-known subgradient algorithm is used to generate an approximate estimate of the minimum f_i^*.

Remark 3.2. Here we make some comparisons between our algorithms and the two algorithms in [31]. First, our algorithm and the two algorithms in [31] are all gradient-based first-order methods and then can achieve the optimal convergence rate in terms of objective value by choosing appropriate stepsizes. Second, our distributed subgradient algorithm (3.2) requires only the decision variable x (with dimension m) to be iterated (the operation includes weighted average, subgradient computation and convex projection). However, besides the decision variable, the distributed penalty-based method proposed in [31] requires an additional auxiliary variable associated with the objective function value (with dimension one) to be iterated, and the distributed primal-dual method in [31] further requires both the auxiliary variable and the dual variable (with dimension n) to be iterated. Finally, our algorithm can only achieve an approximate optimality due to the approximate of the sum objective optimization to the min-max optimization while the algorithms in [31] can achieve the exact optimality.

Here we make two assumptions on the communication graph, which are basic in the multi-agent literature and assumed to be true throughout this paper.

Assumption 3.1. The adjacency matrix A is a doubly weighting matrix.

Assumption 3.2. The induced graph $\mathcal{G}_A$ of the adjacency matrix A is strongly connected with self-loops.

3.3 A Convergence Result

Theorem 3.3. *Consider distributed subgradient optimization algorithm (3.2) with $p \geq 2$. Then*

$$0 \leq \frac{1}{n}\sum_{i=1}^{n}(f_i(\hat{x}_i(k+1)) - f_i^*)^p - \min_X \frac{1}{n}\sum_{i=1}^{n}(f_i(x) - f_i^*)^p \leq \frac{1}{\sum_{r=0}^{k}\alpha_r}R_0, \tag{3.3}$$

where

$$R_0 = \frac{1}{2n}\sum_{i=1}^{n}||x_i(0)||^2_{\arg\min_X \frac{1}{n}\sum_{i=1}^{n}(f_i(x)-f_i^*)^p} + 2L_2\max_{r_1,r_2}||x_{r_1}(0) - x_{r_2}(0)|| + \frac{2L_2c_0\sigma}{1-\sigma}||x(0)||$$
$$+ \Big(\frac{L_1^2}{2} + 4L_1L_2 + \frac{2c_0L_1L_2\sqrt{n}}{1-\sigma}\Big)\sum_{r=0}^{k}\alpha_r^2 + \frac{L_3L_0^2}{2}\sum_{r=0}^{k}\frac{\alpha_r}{\sum_{s=0}^{r+1}\alpha_s}\sum_{s=0}^{r+1}\alpha_s^2$$

$$+\frac{L_3}{2}\sum_{r=0}^{k}\frac{\alpha_r}{\sum_{s=0}^{r+1}\alpha_s}\frac{1}{n}\sum_{i=1}^{n}||x_i(0)||^2_{\arg\min_X f_i},$$

and the numbers $L_i, 0 \le i \le 3$ are given in the proof.

Proof. Take $x^* \in \arg\min_X \frac{1}{n}\sum_{i=1}^{n}(f_i(x)-f_i^*)^p$. By the projection inequality $||P_X(y)-z|| \le ||y-z||$ for any $y \in \mathrm{R}^m$ and $z \in X$, we have

$$\begin{aligned}
&||x_i(k+1)-x^*||^2\\
&\le ||\breve{x}_i(k)-\alpha_k p\big(f_i(\breve{x}_i(k))-w_i(k+1)\big)^{p-1}v_i(k)-x^*||^2\\
&= ||\breve{x}_i(k)-x^*||^2+\alpha_k^2L_1^2-2\alpha_k(\breve{x}_i(k)-x^*)'p\big(f_i(\breve{x}_i(k))-f_i^*\big)^{p-1}v_i(k)\\
&\qquad +2\alpha_k(\breve{x}_i(k)-x^*)'v_i(k)p\big[(f_i(\breve{x}_i(k))-f_i^*)^{p-1}-(f_i(\breve{x}_i(k))-w_i(k+1))^{p-1}\big],
\end{aligned}\tag{3.4}$$

where $L_1 := p\sup_{i,k}|f_i(\breve{x}_i(k))-w_i(k+1)|^{p-1}L_0$, $L_0 = \sup_{z\in\bigcup_{i,x\in X}\partial f_i(x)}||z|| < \infty$, which are finite due to the boundedness of X and the convexity of f_i. It follows from the mean value theorem that there exists $\beta_{i,k} \in [0,1]$ such that

$$\begin{aligned}
(f_i(\breve{x}_i(k))-f_i^*)^{p-1} &= (f_i(\breve{x}_i(k))-w_i(k+1))^{p-1}\\
&\quad -(p-1)(f_i(\breve{x}_i(k))-(w_i(k+1)+\beta_{i,k}(f_i^*-w_i(k+1))))^{p-2}(f_i^*-w_i(k+1)).
\end{aligned}\tag{3.5}$$

Similarly, there exists $\gamma_{i,k} \in [0,1]$ such that

$$\begin{aligned}
(f_i(\bar{x}(k))-f_i^*)^p &= (f_i(\breve{x}_i(k))-f_i^*)^p\\
&\quad +p(f_i(\breve{x}_i(k))+\gamma_{i,k}(f_i(\bar{x}(k))-f_i(\breve{x}_i(k)))-f_i^*)^{p-1}(f_i(\bar{x}(k))-f_i(\breve{x}_i(k))),
\end{aligned}\tag{3.6}$$

where $\bar{x}(k) = \frac{1}{n}\sum_{i=1}^{n}x_i(k)$. Some simple calculations give

$$0 \le f_i(\hat{z}_i(k))-f_i^* \le \frac{1}{\sum_{r=0}^{k}\alpha_r}\Big(\frac{1}{2}||x_i(0)||^2_{\arg\min_X f_i}+\frac{L_0^2}{2}\sum_{r=0}^{k}\alpha_r^2\Big).\tag{3.7}$$

By (3.4), (3.5), (3.6), (3.7) and the convexity of function $||\cdot||^2$, we get

$$\begin{aligned}
||x_i(k+1)-x^*||^2 &\le \sum_{j\in\mathcal{N}_i}a_{ij}||x_j(k)-x^*||^2+\alpha_k^2L_1^2+2\alpha_k((f_i(x^*)-f_i^*)^p-(f_i(\bar{x}(k))-f_i^*)^p)\\
&\quad +2L_2\alpha_k||\bar{x}(k)-\breve{x}_i(k)||\\
&\quad +2L_3\frac{\alpha_k}{\sum_{r=0}^{k+1}\alpha_r}\Big(\frac{1}{2}||x_i(0)||^2_{\arg\min_X f_i}+\frac{L_0^2}{2}\sum_{r=0}^{k+1}\alpha_r^2\Big),
\end{aligned}\tag{3.8}$$

where $\bar{x}(k) = \frac{1}{n}\sum_{i=1}^{n}x_i(k)$,

$$L_2 := pL_0\sup_{i,x,y\in X,\gamma\in[0,1]}(f_i(x)+\gamma(f_i(y)-f_i(x))-f_i^*)^{p-1},$$

$$L_3 := p(p-1)L_0\sup_{i,x,y\in X,\beta\in[0,1]}|x-x^*|\big|f_i(x)-(f_i(y)+\beta(f_i^*-f_i(y)))\big|^{p-2},$$

which are also finite. Since $\mathbf{1}'A = \mathbf{1}'$, multiplying both sides of (3.8) by $\frac{1}{n}$ and then taking the sum over $i = 1, ..., n$ yield

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n}||x_i(k+1)-x^*||^2 \leq & \frac{1}{n}\sum_{i=1}^{n}||x_i(k)-x^*||^2 \\ & + 2\alpha_k\Big[\frac{1}{n}\sum_{i=1}^{n}((f_i(x^*)-f_i^*)^p-(f_i(\bar{x}(k))-f_i^*)^p)\Big] \\ & + 2L_2\alpha_k\frac{1}{n}\sum_{i=1}^{n}||\bar{x}(k)-x_i(k)|| + \alpha_k^2L_1^2 \\ & + L_3L_0^2\frac{\alpha_k\sum_{r=0}^{k+1}\alpha_r^2}{\sum_{r=0}^{k+1}\alpha_r} + L_3\frac{\alpha_k}{\sum_{r=0}^{k+1}\alpha_r}\frac{1}{n}\sum_{i=1}^{n}||x_i(0)||^2_{\arg\min_X f_i},\end{aligned} \tag{3.9}$$

where we use the relation $\sum_{i=1}^{n}||\bar{x}(k)-\breve{x}_i(k)|| \leq \sum_{i=1}^{n}||\bar{x}(k)-x_i(k)||$ following from $\mathbf{1}'A = \mathbf{1}'$. Rearranging the terms in (3.9) leads to

$$\begin{aligned}& 2\sum_{r=0}^{k}\alpha_r\frac{1}{n}\sum_{i=1}^{n}((f_i(\bar{x}(r))-f_i^*)^p-(f_i(x^*)-f_i^*)^p) \\ \leq & \frac{1}{n}\sum_{i=1}^{n}||x_i(0)-x^*||^2 + 2L_2\sum_{r=0}^{k}\alpha_r\frac{1}{n}\sum_{i=1}^{n}||\bar{x}(r)-x_i(r)|| + L_1^2\sum_{r=0}^{k}\alpha_r^2 \\ & + L_3L_0^2\sum_{r=0}^{k}\frac{\alpha_r}{\sum_{s=0}^{r+1}\alpha_s}\sum_{s=0}^{r+1}\alpha_s^2 + L_3\sum_{r=0}^{k}\frac{\alpha_r}{\sum_{s=0}^{r+1}\alpha_s}\frac{1}{n}\sum_{i=1}^{n}||x_i(0)||^2_{\arg\min_X f_i} \\ =: & \vartheta(k).\end{aligned}$$

Denote $\lambda_k = \sum_{r=0}^{k}\alpha_r$. Then

$$\frac{\sum_{r=0}^{k}\alpha_r\frac{1}{n}\sum_{i=1}^{n}((f_i(\bar{x}(r))-f_i^*)^p-(f_i(x^*)-f_i^*)^p)}{\lambda_k} \leq \frac{\vartheta(k)}{2\lambda_k}. \tag{3.10}$$

Now we give an estimate for the term $\sum_{r=0}^{k}\alpha_r\frac{1}{n}\sum_{i=1}^{n}||\bar{x}(r)-x_i(r)||$. Clearly,

$$x_i(k+1) = P_X\big(\breve{x}_i(k)-\alpha_k p(f_i(\breve{x}_i(k))-w_i(k+1))^{p-1}v_i(k)\big) =: \breve{x}_i(k)+\nu_i(k),$$

where $\nu_i(k) = P_X\big(\breve{x}_i(k)-\alpha_k p(f_i(\breve{x}_i(k))-w_i(k+1))^{p-1}v_i(k)\big)-\breve{x}_i(k)$, $||\nu_i(k)|| \leq \alpha_k p\big|f_i(\breve{x}_i(k))-w_i(k+1)\big|^{p-1}||v_i(k)|| \leq \alpha_k L_1$. The preceding algorithm can be written as a compact form

$$x(k+1) = (A\otimes I_m)x(k)+\nu(k),$$

where $x(k) = (x_1'(k), ..., x_n'(k))'$, $\nu(k) = (\nu_1'(k), ..., \nu_n'(k))'$. For national simplicity, without loss of generality we assume $m = 1$ in the subsequent proof of this theorem. It follows from the preceding equation that

$$x(k) = A^kx(0)+\sum_{r=0}^{k-2}A^{k-r-1}\nu(r)+\nu(k-1).$$

From $\mathbf{1}'A = \mathbf{1}'$ again, it can be seen that $\bar{x}(k) = \bar{x}(0) + \sum_{r=0}^{k-1} \frac{1}{n} \sum_{j=1}^{n} \nu_j(r)$. Therefore,

$$\begin{aligned}\big|\bar{x}(k) - x_i(k)\big| &= \Big|\bar{x}(k) - \sum_{j=1}^{n}(A^k)_{ij}x_j(0) - \sum_{r=0}^{k-2}\sum_{j=1}^{n}(A^{k-r-1})_{ij}\nu_j(r) - \nu_i(k-1)\Big| \\ &\le \big|\big|e_i'A^k - \frac{1}{n}\mathbf{1}'\big|\big|\,||x(0)|| + \sum_{r=0}^{k-2}\big|\big|e_i'A^{k-r-1} - \frac{1}{n}\mathbf{1}'\big|\big|\,||\nu(r)|| \\ &\quad + \Big|\frac{1}{n}\sum_{j=1}^{n}\nu_j(k-1) - \nu_i(k-1)\Big|. \end{aligned} \tag{3.11}$$

By the inequalities (3.11), (2.1) and the relation $||\nu_i(k)|| \le \alpha_k L_1$, we get

$$|\bar{x}(k) - x_i(k)| \le c_0\sigma^k||x(0)|| + c_0\sum_{s=0}^{k-2}\sigma^{k-s-1}\alpha_s\sqrt{n}L_1 + 2\alpha_{k-1}L_1.$$

As a consequence, for each j,

$$\begin{aligned}&\sum_{r=0}^{k}\alpha_r\frac{1}{n}\sum_{i=1}^{n}||\bar{x}(r) - x_i(r)|| \\ &\le \frac{1}{n}\sum_{i=1}^{n}||\bar{x}(0) - x_i(0)|| + c_0\sum_{r=1}^{k}\sigma^r||x(0)|| \\ &\quad + c_0\sum_{r=2}^{k}\alpha_r\sum_{s=0}^{r-2}\sigma^{r-s-1}\alpha_s\sqrt{n}L_1 + 2\sum_{r=1}^{k}\alpha_r\alpha_{r-1}L_1 \\ &\le \frac{1}{n}\sum_{i=1}^{n}||\bar{x}(0) - x_i(0)|| + \frac{c_0\sigma}{1-\sigma}||x(0)|| + \frac{c_0\sqrt{n}L_1}{1-\sigma}\sum_{r=0}^{\infty}\alpha_r^2 + 2L_1\sum_{r=0}^{\infty}\alpha_r^2. \end{aligned} \tag{3.12}$$

Then it follows from the convexity of f_i and $||\cdot||^p$ and (3.10) that

$$\begin{aligned}0 &\le \frac{1}{n}\sum_{i=1}^{n}\Big(f_i\Big(\frac{\sum_{r=0}^{k}\alpha_r x_i(r)}{\lambda_k}\Big) - f_i^*\Big)^p - \min_X\frac{1}{n}\sum_{i=1}^{n}(f_i(x^*) - f_i^*)^p \\ &= \frac{1}{n}\sum_{i=1}^{n}\Big(f_i\Big(\frac{\sum_{r=0}^{k}\alpha_r \bar{x}(r)}{\lambda_k}\Big) - f_i^*\Big)^p - \min_X\frac{1}{n}\sum_{i=1}^{n}(f_i(x^*) - f_i^*)^p \\ &\qquad + \frac{1}{n}\sum_{i=1}^{n}\Big(f_i\Big(\frac{\sum_{r=0}^{k}\alpha_r x_i(r)}{\lambda_k}\Big) - f_i^*\Big)^p - \frac{1}{n}\sum_{i=1}^{n}\Big(f_i\Big(\frac{\sum_{r=0}^{k}\alpha_r \bar{x}(r)}{\lambda_k}\Big) - f_i^*\Big)^p \\ &\le \frac{\vartheta(k)}{2\lambda_k} + \frac{L_2\sum_{r=0}^{k}\alpha_r\frac{1}{n}\sum_{i=1}^{n}||x_i(r) - \bar{x}(r)||}{\lambda_k}. \end{aligned}$$

Noting (3.12) and the relation $||\bar{x}(0) - x_i(0)|| \le \max_{r_1,r_2}||x_{r_1}(0) - x_{r_2}(0)||$, the conclusion follows by taking the infimum over all $x^* \in \arg\min_X \frac{1}{n}\sum_{i=1}^{n}(f_i(x) - f_i^*)^p$ on the right hand side of the preceding inequality. □

We also have

$$0 \le f_i(\hat{z}_i(k)) - f_i^* \le \frac{1}{\sum_{r=0}^{k}\alpha_r}\Big(\frac{1}{2}||z_i(0)||^2_{\arg\min_X f_i} + \frac{L_0^2}{2}\sum_{r=0}^{k}\alpha_r^2\Big). \tag{3.13}$$

By combining the estimate (3.13) with $x_i(0) = z_i(0)$, the relations $\sum_{r=0}^{k} \alpha_r \geq 2(\sqrt{k+2}-1)$, $\sum_{r=0}^{k} \alpha_r^2 \leq \ln(k+1)+1$, we get that for $\alpha_k = 1/\sqrt{k+1}$,

$$0 \leq f_i(\hat{z}_i(k)) - f_i^* \leq \frac{1}{4(\sqrt{k+2}-1)}\big(||x_i(0)||^2_{\arg\min_X f_i} + L_0^2(\ln(k+1)+1)\big)$$

and then it can be seen that when

$$\begin{aligned} &k \geq K_0 = K_0(\epsilon, p) \\ &= \max\Big\{\Big(\frac{L_4 \max_{1\leq i\leq n} ||x_i(0)||^2_{\arg\min_X f_i}}{2\epsilon} + 1\Big)^2,\ \min\Big\{k \geq 1 \Big| \frac{(\ln(k+1)+1)^2}{(k+2)^{\frac{1}{2}}-1} \leq \frac{2\epsilon}{L_4 L_0^2}\Big\}\Big\}, \end{aligned} \tag{3.14}$$

$0 \leq L_4(f_i(\hat{z}_i(k)) - f_i^*) \leq \epsilon$ for all i, where L_4 is a positive number to be determined later.

The following corollary is straightforward from the estimate (3.3).

Corollary 3.4. *Consider distributed subgradient optimization algorithm (3.2) with $p \geq 2$. Suppose $\sum_{k=0}^{\infty} \alpha_k = \infty$, $\lim_{k\to\infty} \frac{\sum_{r=0}^{k} \alpha_r^2}{\sum_{r=0}^{k} \alpha_r} = 0$ and $\lim_{k\to\infty} \frac{1}{\sum_{r=0}^{k} \alpha_r} \sum_{r=0}^{k} \frac{\alpha_r}{\sum_{s=0}^{r+1} \alpha_s} \sum_{s=0}^{r+1} \alpha_s^2 = 0$. Then*

$$\lim_{k\to\infty} \frac{1}{n} \sum_{i=1}^{n} \Big(f_i\Big(\frac{\sum_{r=0}^{k} \alpha_r x_i(r)}{\sum_{r=0}^{k} \alpha_r}\Big) - f_i^*\Big)^p = \min_{x\in X} \frac{1}{n} \sum_{i=1}^{n} (f_i(x) - f_i^*)^p.$$

Furthermore, let $\alpha_k = \frac{1}{\sqrt{k+1}}$, we have

$$\begin{aligned} 0 &\leq \frac{1}{n} \sum_{i=1}^{n} \Big(f_i\Big(\frac{\sum_{r=0}^{k} \alpha_r x_i(r)}{\sum_{r=0}^{k} \alpha_r}\Big) - f_i^*\Big)^p - \min_X \frac{1}{n} \sum_{i=1}^{n} \big(f_i(x) - f_i^*\big)^p \\ &\leq \frac{1}{\sqrt{k+2}-1}[c_1 + c_2(\ln(k+1)+1) + c_3(\ln(k+1)+1)^2], \end{aligned} \tag{3.15}$$

where

$$\begin{aligned} c_1 &= \frac{1}{4n} \sum_{i=1}^{n} ||x_i(0)||^2_{\arg\min_X \frac{1}{n}\sum_{i=1}^{n}(f_i(x)-f_i^*)^p} + L_2 \max_{r_1,r_2} ||x_{r_1}(0) - x_{r_2}(0)|| + \frac{L_2 c_0 \sigma}{1-\sigma} ||x(0)|| \\ &\quad + \frac{5L_3 L_0^2}{8} + \frac{L_3}{4n} \sum_{i=1}^{n} ||x_i(0)||^2_{\arg\min_X f_i}, \\ c_2 &= \frac{L_1^2}{4} + 2L_1 L_2 + \frac{c_0 L_1 L_2 \sqrt{n}}{1-\sigma} + \frac{L_3 L_0^2}{4} + \frac{L_3}{8n} \sum_{i=1}^{n} ||x_i(0)||^2_{\arg\min_X f_i}, \\ c_3 &= \frac{L_3 L_0^2}{4}. \end{aligned}$$

Consequently,

$$0 \leq \frac{1}{n} \sum_{i=1}^{n} \Big(f_i\Big(\frac{\sum_{r=0}^{k} \alpha_r x_i(r)}{\sum_{r=0}^{k} \alpha_r}\Big) - f_i^*\Big)^p - \min_X \frac{1}{n} \sum_{i=1}^{n} (f_i(x) - f_i^*)^p \leq \epsilon$$

when $k \geq K_1 = K_1(\epsilon, p) = \max\Big\{1, e^{\frac{c_1}{c_2}-1}, e^{\frac{c_2}{c_3}-1}, \min\big\{k \geq 1 \big| \frac{(\ln(k+1)+1)^2}{(k+2)^{\frac{1}{2}}-1} \leq \frac{\epsilon}{3c_3}\big\}, K_0\Big\}$.

Remark 3.5. A distributed subgradient algorithm with constant stepsize was first proposed in [25] to solve the unconstrained optimization problem $\min \sum_{i=1}^n g_i$, and the authors established an upper bound on the error between the sum objective value at agents' estimates and the optimal objective value. Following that, the authors in [26] considered the constrained optimization problem $\min_X \sum_{i=1}^n g_i$ by utilizing a distributed varying-stepsize algorithm. The authors in [31] employed the distributed subgradient algorithm to solve a SOOP, which is equivalent to the min-max problem $\min_X \max_i g_i$. The authors in [26, 31] showed the optimal convergence under the classical stochastic approximation stepsize conditions $\sum_{k=0}^\infty \alpha_k = \infty$, $\sum_{k=0}^\infty \alpha_k^2 < \infty$. Different from [26, 31], in order to guarantee the optimal convergence of objective value rather than the convergence of agents' estimates, here we do not require stepsize α_k to satisfy the condition $\sum_{k=0}^\infty \alpha_k^2 < \infty$. Note that in this paper we present an estimate of the upper bound on the approximate error between the optimal objective value at an ergodic average of agents' estimates and the optimal objective value, while only optimal convergence is established in [31].

Under the stepsize $\alpha_k = \frac{1}{\sqrt{k+1}}$, the proposed subgradient algorithm (3.2) can achieve the optimality with convergence rate $O\big(\frac{(\ln k)^2}{\sqrt{k}}\big)$. Note that the convergence rate $O\big(\frac{(\ln k)^2}{\sqrt{k}}\big)$ is slightly slower than the convergence rate $O\big(\frac{\ln k}{\sqrt{k}}\big)$ achieved by the standard subgradient method with stepsize $O(\frac{1}{\sqrt{k+1}})$ since in our algorithm agents do not know their own optimal objective values f_i^* and use iterated estimates to approximate their minimum. In fact, from the proof of the above theorem we can see that if agents know their own optimal objective values, the convergence rate $O\big(\frac{\ln k}{\sqrt{k}}\big)$ can also be achieved by our algorithm (in this case, L_3 can be taken as zero). Moreover, it is known that $O\big(\frac{1}{\sqrt{k}}\big)$ is the optimal convergence rate that the first-order subgradient method can achieve for general non-smooth convex optimization problems when the number of iterations is fixed in advance (see Chapter 3 in [24]).

Remark 3.6. Here we give some discussions on the coefficients c_1, c_2, c_3 in the upper bound in (3.15). If we take the initial condition $x_i(0) = 0, i = 1, ..., n$ and assume that agents know their own minimum f_i^*, then c_1, c_2, c_3 reduce to the following ones, respectively

$$
\begin{aligned}
c_1^* &= \frac{1}{4n}\sum_{i=1}^n ||x_i(0)||^2_{\arg\min_X \frac{1}{n}\sum_{i=1}^n (f_i(x)-f_i^*)^p},\\
c_2^* &= \frac{L_1^2}{4} + 2L_1L_2 + \frac{c_0L_1L_2\sqrt{n}}{1-\sigma},\\
c_3^* &= 0.
\end{aligned}
$$

Similar to the upper bound estimate (3.7) of centralized algorithms, we can find that c_1^*, c_2^* are essential for the convergence error of subgradient methods.

The obtained result presents an approximate estimate for $\min_X \frac{1}{n}\sum_{i=1}^n (f_i(x)-f_i^*)^p$ with approximate error characterized by some model parameters. Based on this result, in next section we will present an estimate for M.

4 Minimum Error Estimation

In this section, we will give an estimate for the minimum error M that guarantees the existence of ε-optimal solutions in a fully distributed manner. According to the estimate (3.15), agents need to cooperatively obtain the scalar $\frac{1}{n}\sum_{i=1}^n \Big(f_i\Big(\frac{\sum_{r=0}^k \alpha_r x_i(r)}{\sum_{r=0}^k \alpha_r}\Big) - f_i^*\Big)^p$,

however, which cannot be obtained by any particular agent. In fact, this global information can be obtained by employing the well-known distributed averaging algorithm [14]:

$$q_i(k+1) = \sum_{j\in\mathcal{N}_i} a_{ij}q_j(k), k \geq 0, q_i(0) = (f_i(\hat{x}_i(K_1)) - w_i(K_1+1))^p, i = 1, ..., n. \tag{4.1}$$

Lemma 4.1. *Consider distributed averaging algorithm (4.1). Then*

$$\Big|q_i(k) - \frac{1}{n}\sum_{j=1}^n q_j(0)\Big| \leq \epsilon \ \forall i \ \ \textit{when } k \geq K_2 = K_2(\epsilon) = \frac{\ln\left(\frac{\epsilon}{c_0||q(0)||}\right)}{\ln\sigma}.$$

Proof. Clearly, for each k, $\frac{1}{n}\sum_{j=1}^n q_j(k) = \frac{1}{n}\sum_{j=1}^n q_j(0) = \frac{1}{n}\sum_{j=1}^n \big(f_j(\hat{x}_j(K_1)) - w_j(K_1+1)\big)^p$ because the adjacency matrix A is doubly weighting. Therefore, from (2.1) we have that for each i,

$$\begin{aligned}\Big|q_i(k) - \frac{1}{n}\sum_{j=1}^n q_j(0)\Big| &= \Big|\sum_{j=1}^n (A^k)_{ij}q_j(0) - \frac{1}{n}\sum_{j=1}^n q_j(0)\Big| \\ &\leq c_0||q(0)||\sigma^k.\end{aligned}$$

Then the conclusion follows. □

Algorithm 1 Minimum Error Estimation Algorithm

Input: $0 < \epsilon < 1$, $K_\ell = K_\ell(\epsilon, p), \ell = 1, 2$;

Output: $q_i(K_2)$, $i = 1, ..., n$.

Step 1: Run distributed algorithm (3.2) (K_1+1) steps with $p \geq 2$, $\alpha_k = \frac{1}{\sqrt{k+1}}$ to get $f_i(\hat{x}_i(K_1))$, $w_i(K_1+1)$, $i = 1, ..., n$;

Step 2: Run distributed averaging algorithm (4.1) K_2 steps to get $q_i(K_2), i = 1, ..., n$.

Theorem 4.2. *Let $\epsilon > 0$ be any pre-specified positive number and $q_i(K_2)$, $i = 1, ..., n$ the outputs after running Algorithm 1. Then for each j,*

$$(q_j(K_2) - 3\epsilon)^{\frac{1}{p}} \leq M \leq n^{\frac{1}{p}}(q_j(K_2) + 3\epsilon)^{\frac{1}{p}}.$$

Furthermore, if $M = 0$, then $q_j(K_2) \leq 3\epsilon$, $j = 1, ..., n$.

Proof. Denote $N(p) = \min_X \frac{1}{n}\sum_{i=1}^n (f_i(x) - f_i^*)^p$. It follows from the fact $||z||_\infty \leq ||z||_p \leq n^{\frac{1}{p}}||z||_\infty$ that $\frac{1}{n}M^p \leq N(p) \leq M^p$ and further

$$(N(p))^{\frac{1}{p}} \leq M \leq n^{\frac{1}{p}}(N(p))^{\frac{1}{p}}. \tag{4.2}$$

First similar to (3.5) it is easy to see that for some $\beta_i \in [0, 1]$,

$$\begin{aligned}&\Big|\frac{1}{n}\sum_{i=1}^n \big(f_i(\hat{x}_i(K_0)) - w_i(K_0+1)\big)^p - \frac{1}{n}\sum_{i=1}^n \big(f_i(\hat{x}_i(K_0)) - f_i^*\big)^p\Big| \\ &= \Big|\frac{1}{n}\sum_{i=1}^n p\big[f_i(\hat{x}_i(K_0)) - (w_i(K_0+1) + \beta_i(f_i^* - w_i(K_0+1)))\big]^{p-1}(f_i^* - w_i(K_0+1))\Big|\end{aligned}$$

$$\leq L_4 \max_{1\leq i\leq n} \left|w_i(K_0+1) - f_i^*\right| \leq \epsilon,$$

where $L_4 := p\sup_{i,x,y\in X,\beta\in[0,1]} \left[f_i(x) - (f_i(y) + \beta(f_i^* - f_i(x)))\right]^{p-1} < \infty$. By Lemma 4.1, for any j,

$$\left|q_j(K_2) - \frac{1}{n}\sum_{i=1}^{n}\left(f_i(\hat{x}_i(K_1)) - w_i(K_1+1)\right)^p\right| = \left|q_j(K_2) - \frac{1}{n}\sum_{i=1}^{n} q_i(0)\right| \leq \epsilon.$$

Moreover, from Corollary 3.4 we have $0 \leq \frac{1}{n}\sum_{i=1}^{n}(f_i(\hat{x}_i(K_1)) - f_i^*)^p - N(p) \leq \epsilon$. Therefore, it follows from the preceding estimates that

$$\left|q_j(K_2) - N(p)\right| \leq 3\epsilon. \tag{4.3}$$

Thus, the first conclusion follows from (4.2) and (4.3), while the second one from the first conclusion and the fact that $M = 0$ if and only if $N(p) = 0$ for all $p > 0$. □

Remark 4.3. From Theorem 4.2 we can see that by letting p be sufficiently large such that $n^{\frac{1}{p}}$ approaches to 1 and then for this fixed p we execute Algorithm 1 with sufficiently small ϵ and sufficiently large K_1, K_2 to get $q_i(K_2)$, $i = 1, ..., n$. We can approximate M with arbitrarily small approximate error by the proposed distributed algorithm.

5 Doubly Weighting Adjacency Matrices

To apply the proposed algorithm, the adjacency matrix A of distributed algorithms (3.2) and (4.1) is required to be doubly weighting. In this section, we will provide an estimate on the convergence for a class of weight-balanced adjacency matrix generating algorithms, which are the same as the mirror imbalance-correcting algorithm proposed in [11] except that it does not any specific node selection rule. From the generated weight-balanced matrix, we can get a doubly weighting matrix by adjusting the weights of self-loops, as suggested by [11].

Define the in-degree, out-degree and unbalance degree of node i in the induced graph of a matrix $E = (e_{ij})$ as

$$d_i^- = \sum_{j=1}^{n} e_{ji}, \quad d_i^+ = \sum_{j=1}^{n} e_{ij}, \quad \nu_i = d_i^- - d_i^+,$$

respectively. We say that matrix E is weight-balanced if the in-degree and out-degree of each node are the same (i.e., $d_i^- = d_i^+$, or equivalently, $\nu_i = 0$ for all i).

Let $B(0) = (b_{ij}(0)) \in \mathbb{Z}^{n\times n}$ be any given nonnegative integer-valued matrix with strongly connected induced graph $\mathcal{G}_{B(0)} = (\mathcal{V}, \mathcal{E}_{B(0)})$. Let us consider the following distributed weight-balanced adjacency matrix generating algorithm:

For each $k \geq 0$ and the matrix $B(k) = (b_{ij}(k)) \in \mathbb{Z}^{n\times n}$ at time k, the matrix $B(k+1) = (b_{ij}(k+1)) \in \mathbb{Z}^{n\times n}$ at time $k+1$ is generated as follows: for each i,

$$\begin{aligned}
&\text{if } \nu_i(k) \leq 0, \text{ then}\\
&\qquad b_{ij}(k+1) = b_{ij}(k) \text{ for all } j;\\
&\text{if } \nu_i(k) > 0, \text{ then}\\
&\qquad b_{ij}(k+1) = \begin{cases} b_{ij}(k) + \nu_i(k) & \text{for } j = j_i(k);\\ b_{ij}(k) & \text{for } j \in \mathcal{V}/j_i(k),\end{cases}
\end{aligned} \tag{5.1}$$

where $j_i(k) \in \arg\min_{j\in\mathcal{N}_i/\{i\}} \nu_j(k)$, $\mathcal{N}_i = \{j|(j,i) \in \mathcal{E}_{B(0)}\}$ and $\nu_i(k)$ is the unbalance degree of node i corresponding to matrix $B(k)$.

Note that the only difference between the algorithm (5.1) and the mirror imbalance-correcting algorithm proposed in [11] is that algorithm (5.1) does not specify any specific selection rule of how to select node $j_i(k)$ from the set $\arg\min_{j\in\mathcal{N}_i/\{i\}} \nu_j(k)$ only requiring that $j_i(k)$ is taken from $\arg\min_{j\in\mathcal{N}_i/\{i\}} \nu_j(k)$.

Before executing each iteration step of algorithm (5.1), all nodes first tell their out-neighbors the weights they give to them and then compute their in-degrees based on the received information from their in-neighbors. Finally, all nodes report their individual unbalance degrees to their in-neighbors. Note that in this algorithm, nodes can just observe, but cannot be allowed to change, the weights that their in-neighbors give.

Define nonnegative function

$$h(k) = \sum_{i=1}^{n} |\nu_i(k)|,$$

which measures the imbalance of $B(k)$. Clearly, $B(k)$ is weight-balanced if and only if $h(k) = 0$.

Proposition 5.1. *If $B(k)$ is weight-balanced, algorithm (5.1) terminates at time k and otherwise,*

$$h(k+1) = h(k) - 2\Big(\sum_{i\in\mathcal{V}_1^+(k)} |\nu_i(k)| + \sum_{i\in\mathcal{V}_3^-(k)} \nu_i(k)\Big),$$

where the definitions of $\mathcal{V}_1^+(k)$ and $\mathcal{V}_3^-(k)$ are given in the proof. In particular, it holds that $h(k+1) = h(k)$ if $\mathcal{V}_1(k) \cap \mathcal{V}^-(k) = \emptyset$, and $h(k+1) \le h(k) - 2$ if $\mathcal{V}_1(k) \cap \mathcal{V}^-(k) \neq \emptyset$.

Proof. The first conclusion is obvious. We now show the second one. Suppose that $B(k)$ is not weight-balanced. This implies that $h(k) \ge 2$ because the matrix $B(k)$ is integer-valued. It follows from the weight-unbalance of $B(k)$ and $\sum_{i=1}^n d_i^- = \sum_{i=1}^n d_i^+$ that $\mathcal{V}^+(k) \neq \emptyset$ and $\mathcal{V}^-(k) \neq \emptyset$. Moreover, it is easy to see that the arc sets $\mathcal{E}_{B(k)}$, $k \ge 1$ are equal to $\mathcal{E}_{B(0)}$.

Define three node set sequences $\{\mathcal{V}^+(k)\}_{k\ge0}, \{\mathcal{V}^-(k)\}_{k\ge0}, \{\mathcal{V}^0(k)\}_{k\ge0}$, where

$$\mathcal{V}^+(k) = \{i|i \in \mathcal{V}, \nu_i(k) > 0\},\ \ \mathcal{V}^-(k) = \{i|i \in \mathcal{V}, \nu_i(k) < 0\},\ \ \mathcal{V}^0(k) = \{i|i \in \mathcal{V}, \nu_i(k) = 0\}.$$

Define node subsets

$$\mathcal{V}_1(k) = \big\{j|j \in \mathcal{V}^-(k) \cup \mathcal{V}^0(k), \text{there exists } i \in \mathcal{V}^+(k) \text{ such that } j = j_i(k)\big\}.$$

and

$$\begin{aligned}
\mathcal{V}_2(k) &= (\mathcal{V}^-(k) \cup \mathcal{V}^0(k))/\mathcal{V}_1(k),\\
\mathcal{V}_1^+(k) &= \mathcal{V}_1(k) \cap \mathcal{V}^+(k+1),\\
\mathcal{V}_1^-(k) &= \mathcal{V}_1(k) \cap \big(\mathcal{V}^-(k+1) \cup \mathcal{V}^0(k+1)\big),\\
\mathcal{V}_3^+(k) &= \{i|i \in \mathcal{V}^+(k), j_i(k) \in \mathcal{V}_1^+(k)\},\\
\mathcal{V}_3^-(k) &= \{i|i \in \mathcal{V}^+(k), j_i(k) \in \mathcal{V}_1^-(k)\}.
\end{aligned}$$

By the weight update rule (5.1), we have

$$\sum_{i\in\mathcal{V}^+(k)} d_i^+(k+1) = \sum_{i\in\mathcal{V}^+(k)} d_i^+(k) + \sum_{i\in\mathcal{V}^+(k)} \nu_i(k),$$

$$\sum_{i\in\mathcal{V}^+(k)} d_i^-(k+1) = \sum_{i\in\mathcal{V}^+(k)} d_i^-(k) + \Big(\sum_{i\in\mathcal{V}^+(k)} \nu_i(k) - \sum_{i\in\mathcal{V}_3(k)} \nu_i(k)\Big),$$
$$\sum_{i\in\mathcal{V}_1^+(k)} \nu_i(k+1) = \sum_{i\in\mathcal{V}_3^+(k)} \nu_i(k) - \sum_{i\in\mathcal{V}_1^+(k)} |\nu_i(k)|,$$
$$\sum_{i\in\mathcal{V}_1^-(k)} \nu_i(k+1) = \sum_{i\in\mathcal{V}_3^-(k)} \nu_i(k) - \sum_{i\in\mathcal{V}_1^-(k)} |\nu_i(k)|,$$
$$\nu_i(k+1) = \nu_i(k),\ i\in\mathcal{V}_2(k).$$

These equalities lead to

$$\begin{aligned}
h(k+1) &= \sum_{i\in\mathcal{V}^+(k)} |\nu_i(k+1)| + \sum_{i\in\mathcal{V}_1^+(k)} |\nu_i(k+1)| \\
&\qquad + \sum_{i\in\mathcal{V}_1^-(k)} |\nu_i(k+1)| + \sum_{i\in\mathcal{V}_2(k)} |\nu_i(k+1)| \\
&= \sum_{i\in\mathcal{V}^+(k)} \nu_i(k+1) + \sum_{i\in\mathcal{V}_1^+(k)} \nu_i(k+1) - \sum_{i\in\mathcal{V}_1^-(k)} \nu_i(k+1) + \sum_{i\in\mathcal{V}_2(k)} |\nu_i(k)| \\
&= \Big(\sum_{i\in\mathcal{V}^+(k)} d_i^-(k) - \sum_{i\in\mathcal{V}^+(k)} d_i^+(k) - \sum_{i\in\mathcal{V}_3(k)} \nu_i(k)\Big) + \sum_{i\in\mathcal{V}_2(k)} |\nu_i(k)| \\
&\qquad + \sum_{i\in\mathcal{V}_3^+(k)} \nu_i(k) - \sum_{i\in\mathcal{V}_1^+(k)} |\nu_i(k)| - \sum_{i\in\mathcal{V}_3^-(k)} \nu_i(k) + \sum_{i\in\mathcal{V}_1^-(k)} |\nu_i(k)| \\
&= \Big(\sum_{i\in\mathcal{V}^+(k)} \nu_i(k) + \sum_{i\in\mathcal{V}_1(k)} |\nu_i(k)| + \sum_{i\in\mathcal{V}_2(k)} |\nu_i(k)|\Big) \\
&\qquad - 2\Big(\sum_{i\in\mathcal{V}_1^+(k)} |\nu_i(k)| + \sum_{i\in\mathcal{V}_3^-(k)} \nu_i(k)\Big) \\
&= h(k) - 2\Big(\sum_{i\in\mathcal{V}_1^+(k)} |\nu_i(k)| + \sum_{i\in\mathcal{V}_3^-(k)} \nu_i(k)\Big),
\end{aligned}$$

where the first equality follows from the fact that $\mathcal{V}^+(k)\cup\mathcal{V}^0(k)\subseteq\mathcal{V}^+(k+1)\cup\mathcal{V}^0(k+1)$. Then the second conclusion follows.

The last conclusion follows from the following analysis. If $\mathcal{V}_1(k)\cap\mathcal{V}^-(k)=\emptyset$, then $\mathcal{V}_1(k)\subseteq\mathcal{V}^0(k)$ and $\mathcal{V}_1^-(k)=\emptyset$ by the update rule in (5.1). As a result, $\mathcal{V}_1^+(k)=\mathcal{V}_1(k)\subseteq\mathcal{V}^0(k)$ and $\mathcal{V}_3^-(k)=\emptyset$. This implies that $\sum_{i\in\mathcal{V}_1^+(k)}|\nu_i(k)|+\sum_{i\in\mathcal{V}_3^-(k)}\nu_i(k)=0$. If $\mathcal{V}_1(k)\cap\mathcal{V}^-(k)\neq\emptyset$, we can see that either $\mathcal{V}_1^-(k)\neq\emptyset$, implying $\mathcal{V}_3^-(k)\neq\emptyset$ and $\sum_{i\in\mathcal{V}_3^-(k)}\nu_i(k)\geq 1$, or $\mathcal{V}_1^-(k)=\emptyset$, implying $\mathcal{V}_1(k)\cap\mathcal{V}^-(k)\subseteq\mathcal{V}_1^+(k)$ and then $\sum_{i\in\mathcal{V}_1^+(k)}|\nu_i(k)|\geq 1$. □

The above proposition implies that the sequence $\{h(k)\}_k$ is nonincreasing and we can get a weight-balanced matrix for any selection rule choosing j_i that makes the condition $\mathcal{V}_1(k)\cap\mathcal{V}^-(k)\neq\emptyset$ holding at least $\lceil\frac{h(0)}{2}\rceil$ times, where $\lceil z\rceil$ denotes the smallest integer greater than or equal to z. Moreover, as suggested by [11], we can obtain a doubly weighting matrix based on one generated weight-balanced matrix $B=(b_{ij})$ by adjusting the weights of self-loops. In fact, it suffices to let

$$w = \max_{i=1,\ldots,n} \sum_{j=1}^{n} b_{ij} + 1$$

and $C = (c_{ij})$ with $c_{ii} = [b_{ii} + w - \sum_{j=1}^{n} b_{ij}]/w, i = 1, ..., n$; $c_{ij} = b_{ij}/w, \forall j \neq i$. It is easy to see that the matrix C is a doubly weighting matrix with positive diagonal elements.

Remark 5.2. The authors in [11] proposed a mirror imbalance-correcting algorithm, which is a special case of algorithm (5.1) with a specific selection rule of how to select $j_i(k)$ from $\arg\min_{j\in\mathcal{N}_i/\{i\}} \nu_j(k)$ (called fair-decision rule therein), to generate a weight-balanced adjacency matrix. The authors established its finite time convergence and convergence time complexity analysis. Here the algorithm (5.1) does not specify any specific selection rule and clearly, its convergence time depends on the specific selection rule. However, we should note that generally only the single selection condition $j_i(k) \in \arg\min_{j\in\mathcal{N}_i/\{i\}} \nu_j(k)$ without further selection rule cannot guarantee the convergence of algorithm (5.1), as pointed out in Example 4.8 in [11].

6 Conclusions

We developed a fully distributed method to consider the COS problem and the minimum error estimation problem of ε-optimal solutions in multi-objective optimization. We first proposed a distributed subgradient algorithm and then present an optimal convergence estimate. Based on this estimate, we established an approximate estimate on the minimum error that can guarantee the existence of ε-optimal solutions. We also obtained an estimate on the convergence of a class of weight-balanced adjacency matrix generating algorithms.

Acknowledgements

The authors are grateful to the associate editor, two anonymous referees and Prof. Fukushima - the editor of this journal for their helpful comments and suggestions.

References

[1] E. Ballestero and C. Romero, *Multiple Criteria Decision Making and its Applications to Economic Problems*, Kluwer Academic Publishers, Boston, 1998.

[2] H.P. Benson and S. Sayin, Toward finding global representations of the efficient set in multiple-objective mathematical programming, *Naval Res. Logist.* 44 (1997) 47–67.

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, *Found. Trends Mach. Learn.* 3 (2011) 1–122.

[4] G. Chen, X. Huang and X. Yang, *Vector Optimization: Set-valued and Variational Analysis*, Springer, 2005.

[5] J. Chen and A.H. Sayed, Diffusion adaptation strategies for distributed optimization and learning over networks, *IEEE Trans. Sig. Proc.* 60 (2012) 4289–4305.

[6] J. Chen and A.H. Sayed, Distributed Pareto optimization via diffusion strategies, *IEEE J. Sel. Top. Sig. Proc.* 7 (2013) 205–220.

[7] I. Das and J. Dennis, Normal boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM J. Optim.* 8 (1998) 631–657.

[8] J. Duchi, A. Agarwal and M. Wainwright, Dual averaging for distributed optimization: Convergence analysis and network scaling, *IEEE Trans. Automat. Control* 57 (2012) 592–606.

[9] M. Ehrgott, *Multicriteria Optimization*, Springer Verlag, 2000.

[10] J. Fliege, Gap-free computation of Pareto-points by quadratic scalarizations, *Math. Meth. Oper. Res.* 59 (2004) 69–89.

[11] B. Gharesifard and J. Cortés, Distributed strategies for generating weight-balanced and doubly stochastic digraphs, *European J. Control* 18 (2012) 539–557.

[12] B. Gharesifard and J. Cortés, Distributed convergence to Nash equilibria in two-network zero-sum games, *Automatica* 49 (2013) 1683–1692.

[13] C. Gutiérrez, B. Jiménez and V. Novo, On approximate efficiency in multiobjective programming, *Math. Meth. Oper. Res.* 64 (2006) 165–185.

[14] A. Jadbabaie, J. Lin and A.S. Morse, Coordination of groups of mobile agents using nearest neighbor rules, *IEEE Trans. Automat. Control* 48 (2003) 988–1001.

[15] B. Johansson, M. Rabi and M. Johansson, A randomized incremental subgradient method for distributed optimization in networked systems, *SIAM J. Optim.* 20 (2009) 1157–1170.

[16] E. Karasakal and M. Köksalan, Generating a representative subset of the nondominated frontier in multiple criteria decision making, *Oper. Res.* 59 (2009) 187–199.

[17] K. Klamroth and K. Miettinen, Integrating approximation and interactive decision making in multicriteria optimization, *Oper. Res.* 56 (2008) 222–234.

[18] I. Lobel and A. Ozdaglar, Distributed subgradient methods for convex optimization over random networks, *IEEE Trans. Automat. Control* 56 (2011) 1291–1306.

[19] Y. Lou, Y. Hong, L. Xie, G. Shi and K.H. Johansson, Nash equilibrium computation in subnetwork zero-sum games with switching communications, *IEEE Trans. Automat. Control* 61 (2016) 2920–2935.

[20] Y. Lou, G. Shi, K.H. Johansson and Y. Hong, Approximate projected consensus for convex intersection computation: Convergence analysis and critical error angle. *IEEE Trans. Automat. Control* 59 (2014) 1722–1736.

[21] Y. Lou and S. Wang, Approximate representation of the Pareto frontier in multiparty negotiations: Decentralized methods and privacy preservation, *European J. Oper. Res.* 254 (2016) 968–976.

[22] M. Masin and Y. Bukchin, Diversity maximization approach for multiobjective optimization, *Oper. Res.* 56 (2008) 411–424.

[23] V.N. Nefëdov, Approximation of a set of Pareto-optimal solutions, *Comput. Math. Math. Phys.* 26 (1986) 99–107.

[24] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Kluwer Academic Publishers, 2004.

[25] A. Nedić and A. Ozdaglar, Distributed subgradient methods for multi-agent optimization. *IEEE Trans. Automat. Control* 54 (2009) 48–61.

[26] A. Nedić, A. Ozdaglar and P. A. Parrilo, Constrained consensus and optimization in multi-agent networks, *IEEE Trans. Automat. Control* 55 (2010) 922–938.

[27] S.S. Ram, A. Nedić and V.V. Veeravalli, Distributed stochastic subgradient projection algorithms for convex optimization, *J. Optim. Theory Appl.* 147 (2010) 516–545.

[28] S. Ruzika and M.M. Wiecek, Approximation methods in multiobjective programming, *J. Optim. Theory Appl.* 126 (2005) 473–501.

[29] S. Sayin, A procedure to find discrete representations of the efficient set with specified coverage errors, *Oper. Res.* 51 (2003) 427–436.

[30] K. Srivastava and A. Nedić, Distributed asynchronous constrained stochastic optimization, *IEEE J. Sel. Top. Sig. Proc.* 5 (2011) 772–790.

[31] K. Srivastava, A. Nedić and D. Stipanovic, Distributed Bregman-distance algorithms for min-max optimization, a chapter in the book *Agent-based Optimization*, I. Czarnowski, P. Jedrzejowicz and J. Kacprzyk (Eds.), Springer Studies in Computational Intelligence, Springer, 2013, pp. 143–174.

[32] R.E. Steuer, *Multiple Criteria Optimization: Theory, Computation, and Application*, John Wiley, New York, 1986.

[33] D.J. White, Epsilon efficiency, *J. Optim. Theory Appl.* 49 (1986) 319–337.

[34] K. Yuan, Q. Ling and W. Yin, On the convergence of decentralized gradient descent, *SIAM J. Optim.,* 26 (2016) 1835–1854.

Manuscript received 26 September 2014
revised 9 Octoeber 2015, 2 September 2016
accepted for publication 14 February 2017

Youcheng Lou
Key Laboratory of Management Decision and Information Systems
Academy of Mathematics and Systems Science
Chinese Academy of Sciences
Beijing 100190, China
E-mail address: louyoucheng@amss.ac.cn

Shouyang Wang
Academy of Mathematics and Systems Science
Chinese Academy of Sciences
and School of Economics and Management
University of Chinese Academy of Sciences
Beijing 100190, China
E-mail address: sywang@amss.ac.cn