Hindawi
Journal of Mathematics
Volume 2021, Article ID 5540262, 19 pages
https://doi.org/10.1155/2021/5540262

Research Article

A Study on Distributed Optimization over Large-Scale Networked Systems

Hansi K. Abeynanda [1] **and G. H. J. Lanel** [2]

[1]*Mathematics Unit, Faculty of Humanities and Sciences, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka*
[2]*Department of Mathematics, Faculty of Applied Sciences, University of Sri Jayewardenepura, Nugegoda, Sri Lanka*

Correspondence should be addressed to Hansi K. Abeynanda; kavindika.a@sliit.lk

Received 8 February 2021; Revised 2 April 2021; Accepted 7 April 2021; Published 29 April 2021

Academic Editor: Efthymios G. Tsionas

Distributed optimization is a very important concept with applications in control theory and many related fields, as it is high fault-tolerant and extremely scalable compared with centralized optimization. Centralized solution methods are not suitable for many application domains that consist of large number of networked systems. In general, these large-scale networked systems cooperatively find an optimal solution to a common global objective during the optimization process. Thus, it gives us an opportunity to analyze distributed optimization techniques that is demanded in most distributed optimization settings. This paper presents an analysis that provides an overview of decomposition methods as well as currently existing distributed methods and techniques that are employed in large-scale networked systems. A detailed analysis on gradient like methods, subgradient methods, and methods of multipliers including the alternating direction method of multipliers is presented. These methods are analyzed empirically by using numerical examples. Moreover, an example highlighting the fact that the gradient method fails to solve distributed problems in some circumstances is discussed under numerical results. A numerical implementation is used to demonstrate that the alternating direction method of multipliers can solve this particular problem, by revealing its robustness compared with the gradient method. Finally, we conclude the paper with possible future research directions.

1. Introduction

Optimization is a mathematical discipline which determines the best possible solution corresponding to the optimum performance of a quantitatively well-defined system. The theory of optimization has been established as a desirable tool that is used in a wide range of disciplines, such as automatic control systems, estimation and signal processing, communications and networks, electronic circuit design, data analysis and modeling, statistics, and finance [1–3]. In the recent study [4], the novelty search, a tool that is used in evolutionary and swarm robotics was developed for the use of global optimization. Formally, a mathematical optimization problem can be posed as follows:

$$\begin{aligned} &\text{minimize } f_0(x) \\ &\text{subject to } x \in \mathscr{C}, \end{aligned} \tag{1}$$

where f_0 is a real-valued objective function of the decision variables $x \in \mathbb{R}^n$ and $\mathscr{C} \subseteq \mathbb{R}^n$.

However, in reality, it may be difficult or not possible to find analytic solutions to certain optimization problems. As a result, iterative methods that provide approximate solutions have been introduced by researchers. Algorithms that are used to solve optimization problems have been extensively analyzed mainly under centralized and decentralized architectures [5, 6]. Centralized solution methods are not suitable for many communication networking problems such as large-scale and data-intensive problems that demand distributed solutions. Consequently, the application of distributed optimization techniques where subsystems coordinate to find a solution to the original problem is of utmost importance. Intranets, the Internet, telecommunication networks, aircraft control systems, sensor networks, and electronic banking are some important examples for

distributed systems. These systems consist of a large number of smaller subsystems, and they integrate together to reach an optimal status of the process. This optimal status of process in large-scale networked systems needs to be achieved without incurring errors and exceeding already set time limits for expected outcomes. Therefore, the study of well-established theoretical concepts together with empirical implementations on distributed optimization is critical. This gives us an opportunity to analyze currently existing distributed techniques and methods. In general, we may have many subsystems in a distributed optimization setting. We consider the following optimization problem with five subsystems as an example to provide a deeper explanation of distributed optimization:

$$\begin{aligned} &\text{minimize} \quad f_1(x_1, y) + f_2(x_2, y, r) \\ &\quad + f_3(x_3, y) + f_4(x_4, y, z) + f_5(x_5, r, z) \\ &\text{subject to} \quad x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2, x_3 \in \mathcal{X}_3, x_4 \in \mathcal{X}_4, \\ &\quad x_5 \in \mathcal{X}_5, y \in \mathcal{Y}, r \in \mathcal{R}, z \in \mathcal{Z}, \end{aligned} \tag{2}$$

where $\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3, \mathcal{X}_4, \mathcal{X}_5, \mathcal{R}$, and $\mathcal{Z}$ are subsets of $\mathbb{R}^n$. In this problem, we can observe that there are three complicating variables y, r, and z. The variable y is shared by subsystems $1, 2, 3$, and 4, the variable r is shared by subsystems 2 and 5, while the variable z is shared among subsystems 4 and 5. Figure 1 shows the associated decomposition structure of (2), and the related distributed problem can be stated as follows:

$$\begin{aligned} &\text{minimize} \quad f_1(x_1, y_1) + f_2(x_2, y_2, r_1) + f_3(x_3, y_3) \\ &\quad + f_4(x_4, y_4, z_1) + f_5(x_5, r_2, z_2) \\ &\text{subject to } y_1 = y_2 = y_3 = y_4, r_1 = r_2, z_1 = z_2. \end{aligned} \tag{3}$$

Here, we can observe that problem (3) is minimized by multiple users cooperatively. Hence, a distributed method is required to find a solution.

Many networked systems cannot communicate exact information between subsystems due to unavoidable errors that may occur as a result of limited communication bandwidths and sometimes due to measurement errors [7, 8]. Therein lies the importance of analysing quantized distributed methods in real life situations [9–13]. Although many quantized distributed methods have been analyzed, deeper investigation of quantization methods is still required.

We present the outline of our paper as follows. In Section 2, we discuss the preliminaries related to distributed optimization and primal and dual decomposition. Section 3 provides a general literature review on currently existing well-known distributed optimization methods. Next, in Sections 4, 5, and 6, we discuss the gradient method, the subgradient method, and the alternating direction method of multipliers (ADMM), respectively. In those sections, we discuss the theoretical concepts of the relevant methods as well as previous studies performed on them. In Section 7, we continue our discussion on distributed optimization with noise to emphasize the importance of involvement of error in distributed optimization methods. In Section 8, we provide our numerical results to discuss the convergence of aforementioned distributed methods. Finally, in Section 9, we conclude our paper with possible future research directions.

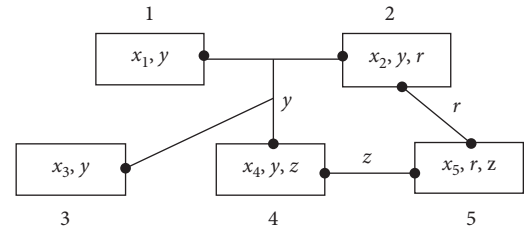

Figure 1: Decomposition structure with five subsystems and three coupling variables.

2. Preliminaries

In this section, we discuss the concept of distributed optimization and we introduce primal decomposition and dual decomposition, which play an important role in distributed optimization. Our introduction on primal decomposition and dual decomposition is mainly inspired by the lecture notes on decomposition methods by boyd et al. [14]. Throughout the paper, we will use following notations.

Notation. We let $\mathbb{R}, \mathbb{R}^n$, and $\mathbb{R}^n_+$ represent the set of real numbers, n-dimensional Euclidean space, and positive orthant in n-dimensional Euclidean space, respectively. For $x \in \mathbb{R}^n$, $\|x\|$ denotes the Euclidean norm and $[x]_{\mathcal{X}}$ denotes the projection of x on to the set $\mathcal{X} \subseteq \mathbb{R}^n$. The set of $n \times m$ matrices is denoted by $\mathbb{R}^{n \times m}$. The transpose of a matrix A is given by A^T. ∇f represents the gradient of a scalar valued function f.

2.1. Distributed Optimization. Distributed optimization is an optimization process that is used in networked systems with a large number of users. This process enables the system to solve a global problem cooperatively even if there is no central controller available in the system. When compared with centralized techniques, distributed optimization has many considerable advantages. In distributed algorithms, nodes or users in the network share information only with necessary parties. This fact improves cyber security and reduces communication cost. Furthermore, distributed techniques have the ability to handle problems even if the problem size is very large. These techniques also have the potential to increase the solution speed [15].

Distributed optimization algorithms solve large-scale and data-intensive problems in a wide range of application areas such as communications [16–19], electricity grid [20, 21], large-scale multiagent systems [22, 23], smart grids, wireless sensor networks [24], and statistical learning. Zhang and Sahraei-Ardakani have developed a fully distributed DC optimal power flow method that incorporates flexible transmission and discussed the effect of communication limitations on the convergence properties [25, 26]. In [27],

authors have presented a study on finite-time consensus opinion dynamics and studied an application to distributed optimization over digraph.

Many distributed optimization algorithms are built on decomposition methods. Decomposition is an interesting approach to solving a global problem by breaking it up into smaller subproblems and solving each of them separately. These subproblems get solved either in parallel or sequentially [6, 14, 28–30]. Decomposition in optimization appears in early work on large-scale linear programs from 1960s [31]. The simplest decomposition structure is available in block separable problems. For an example, a block separable problem can be given as follows:

$$\begin{aligned} &\text{minimize} \quad f_1(x_1) + f_2(x_2) \\ &\text{subject to } x_1 \in \mathscr{X}_1, x_2 \in \mathscr{X}_2. \end{aligned} \tag{4}$$

In this form, we can minimize $f_1(x)$ and $f_2(x)$ separately in parallel and obtain the optimal value and optimal solution. However, this method seems to be trivial and does not seem to be an interesting task as many real life problems appear in a more complex form than this [14]. This problem becomes more complicated and creates more interest when the subvectors x_1 and x_2 are coupled. This situation can be handled by primal decomposition and dual decomposition, which are the most well-known decomposition methods currently available.

2.2. Primal Decomposition. Primal decomposition deals with complicating variables. Here, we consider a constrained minimization problem that consists of m number of users as follows:

$$\begin{aligned} &\text{minimize} \quad f(x) = \sum_{i=1}^{m} f_i(x_i, y) \\ &\text{subject to } x_i \in \mathscr{C}_i, i = 1, 2, \ldots, m, y \in \mathscr{Y}, \end{aligned} \tag{5}$$

where $x = (x_1, x_2, \ldots, x_m, y)$, $\mathscr{C}_i \subseteq \mathbb{R}^n$, $\mathscr{Y} \subseteq \mathbb{R}^n$, and $f_i s$ represent real-valued objective functions of individual users. Here, the variable y is called the complicating variable, which complicates the system. When y is fixed, problem (5) decomposes in to m smaller subproblems.

Subproblems are as follows:

$$S_i(y) = \underset{x_i \in \mathscr{C}_i}{\text{minimize}}\, f_i(x_i, y). \tag{6}$$

Then, the original problem (5) is equivalent to the problem

$$\underset{y \in \mathscr{Y}}{\text{minimize}}\, S(y) = \sum_{i=1}^{m} S_i(y), \tag{7}$$

and this is called the master problem in primal decomposition [14]. Next, the original problem (5) can be solved by solving the master problem (7), using a distributed algorithm under some well-defined assumptions on individual primal objective functions $f_i s$.

2.3. Dual Decomposition. Here, we consider the same problem (5) discussed under primal decomposition only with two users. Then, we have the objective function as $f(x) = f_1(x_1, y) + f_2(x_2, y)$. Next, the problem can be rearranged by introducing new variables y_1 and y_2 as follows [14]:

$$\begin{aligned} &\text{minimize } f(x_1, x_2, y_1, y_2) = f_1(x_1, y_1) + f_2(x_2, y_2) \\ &\text{subject to } y_1 = y_2, x_1 \in \mathscr{C}_1, x_2 \in \mathscr{C}_2, y_1, y_2 \in \mathscr{Y}. \end{aligned} \tag{8}$$

According to this new arrangement, the objective function f is separable. Next, we can apply the decomposition with its dual problem. The Lagrangian of (8) is given by

$$L(x_1, x_2, y_1, y_2, \lambda) = f_1(x_1, y_1) + f_2(x_2, y_2) + \lambda^T (y_1 - y_2). \tag{9}$$

Next, the related dual function is given by

$$g(\lambda) = \inf_{\substack{x_1 \in \mathscr{C}_1 \\ x_2 \in \mathscr{C}_2 \\ y_1, y_2 \in \mathscr{Y}}} L(x_1, x_2, y_1, y_2, \lambda), \tag{10}$$

which is accompanied with subproblems

$$\begin{aligned} g_1(\lambda) &= \inf_{x_1 \in \mathscr{C}_1, y_1 \in \mathscr{Y}} f_1(x_1, y_1) + \lambda^T y_1, \\ g_2(\lambda) &= \inf_{x_2 \in \mathscr{C}_2, y_2 \in \mathscr{Y}} f_2(x_2, y_2) - \lambda^T y_2. \end{aligned} \tag{11}$$

Then, the dual problem of (8) is given by

$$\underset{\lambda}{\text{maximize}}\, g(\lambda) = g_1(\lambda) + g_2(\lambda). \tag{12}$$

This is called the master problem in dual decomposition. This problem can be solved by using an iterative method such as subgradient method, which will be discussed under Section 5. Although we are able to solve the dual problem and find dual optimal measures, we still cannot guarantee that we can find primal optimal measures without introducing some acceptable conditions on the primal objective function. For an example, if f_1 and f_2 are strictly convex, then the primal variables x_1, x_2, y_1, and y_2 found by solving two subproblems g_1 and g_2 are guaranteed to converge to the optimal solution of the primal problem (8) [14].

3. A General Literature Review on Distributed Methods for Solving Optimization Problems

In this section, we provide a general overview of currently existing distributed optimization methods. A detailed analysis will be given in later sections with more technical details. Most of the existing studies done on distributed optimization problems have been analyzed and related solution methods have been discussed when the optimization problem is convex. Convex optimization problems can be solved very reliably and efficiently using interior-point methods, and most of the theories related to convex optimization have been already developed. Therefore, recognizing or formulating a problem as a convex optimization

problem gives us a great advantage. In the texts [5, 6], authors have provided the readers with a very good background to develop a working knowledge on convex optimization to recognize, formulate, and solve convex optimization problems. For example, if we consider a nonconvex constrained optimization problem, the associated negative dual problem is always convex. Hence, in some situations, the original problem can be solved by using the dual problem which provides an easy environment to work with because of the convexity.

We have observed that currently available state-of-the-art distributed methods of solving optimization problems are gradient-based algorithms, subgradient-based algorithms, and their variants, such as ADMM [30, 32–38]. The gradient method is generally applied on unconstrained optimization problems. In 1970, Ramsay had studied gradient methods for optimizing nonlinear functions of several variables that cause difficulties when second derivative approaches are used [39]. In the recent study [40], Nedić et al. have focused on solving a distributed convex optimization problem using "push-pull gradient methods." They have given this interesting name as the agents in the problem network push the gradient information to the neighbors and the decision variable information is pulled by neighbors throughout the method. In [41], Calamai and Moré have studied the convergence properties of the projected gradient method for linearly constrained problems which are useful in large-scale problems. The projected gradient method is a variant of the gradient method which is used in constrained optimization.

The subgradient method can be considered as a generalization of the gradient method and is useful in optimizing nondifferentiable functions. In [9–12, 22, 42], subgradient methods are used to solve large-scaled distributed problems that deals with the sum of a large number of convex local objective functions. References [24, 43–45] are some studies that have been focused on effects of constraints, and they have presented projected subgradient algorithms to solve constrained optimization problems. In [44], Amini and Yousefian have studied a very important class of bilevel convex optimization problems that are often used for large-scale data processing in machine learning and neural networks. The authors in [45] have studied the binary iterative hard thresholding algorithm, a state-of-the-art recovery algorithm in one-bit compressive sensing which makes use of the projected subgradient method.

ADMM is also a well-suited method used in distributed convex optimization over large-scale networked systems arising in statistics and machine learning. The ADMM was first proposed by Gabay, Mercier, Glowinski, and Marrocco [46] in the mid-1970s. In the recent study [47], Xiao et al. have presented a distributed and scalable algorithm for managing the residential demand response programs using ADMM. They have shown through their simulation studies that the proposed method can reduce customers' electricity bills and peak load. Authors in [48] have presented a distributed ADMM for solving the direct current dynamic optimal power flow with carbon emission trading problem. In [49], Hajinezhad and Shi proposed an algorithm related to ADMM to study a class of nonconvex nonsmooth optimization problems with bilinear constraints which are widely used in machine learning and signal processing application domains. The study [50] has presented a modified distributed ADMM to handle nonconvex optimization problems with discrete control variables.

4. The Gradient Method

Let us consider an unconstrained minimization problem as follows:

$$\text{minimize } f(x), \tag{13}$$

where $f: \mathbb{R}^n \longrightarrow \mathbb{R}$ is differentiable and $x \in \mathbb{R}^n$. Then, the gradient method to solve optimization problems of form (13) can be expressed by following iterative process, which starts from some initial point x^0:

$$x^{k+1} = x^k - \alpha_k \nabla f\left(x^k\right), \tag{14}$$

where $\alpha_k \geq 0$ is known to be the step size. The convergence of method (14) can be discussed under various considerations, using the theorems presented in [51].

Theorem 1 (see [51]). *Suppose that $\alpha_k = \alpha$ (a constant step size) in (14). Let $f(x)$ be differentiable on $\mathbb{R}^n$, ∇f is Lipschitz continuous with constant L, and let $f(x)$ be a strongly convex with constant l. Then, method (14) converges to a unique global minimum point x^* with the rate of geometric progression when $0 < \alpha < 2/L$:*

$$\left\|x^k - x^*\right\| \leq cq^k; \quad 0 \leq q < 1. \tag{15}$$

Next, the following theorem shows the convergence of (14) for an even smaller class of functions.

Theorem 2 (see [51]). *Let $f(x)$ be strongly convex and twice differentiable. Suppose that*

$$lI \leq \nabla^2 f(x) \leq LI; \quad l > 0, \forall x. \tag{16}$$

Then, for $0 < \alpha < 2/L$,

$$\begin{aligned} \left\|x^k - x^*\right\| &\leq \left\|x^0 - x^*\right\| q^k, \\ q &= \max\{1 - \alpha l, 1 - \alpha L\} < 1. \end{aligned} \tag{17}$$

Moreover, when $\alpha = 2/(L + l)$, q is minimal and equal to $q^* = (L - l)/(L + l)$. The proofs of Theorem 1 and 2 are given in [51], and the convergence to a local minimum point of $f(x)$ is also discussed in the same text under Theorem 4 of Section 1.4. We discuss the convergence of the gradient method using a numerical example in the numerical results section (Section 8). In Section 8.1, our focus of discussion is the convergence results with the use of primal decomposition.

There are many early studies done on gradient methods [39, 41, 52, 53]. Authors in [53] had combined gradient methods with back propagation methods for neural networks to discuss the optimization of weights of multilayer

neural networks. In the study [52], authors have proposed two new step sizes for the classical-steepest descent method, where α_k in method (14) is used as $\alpha_k = \text{argmin}_\alpha f(x^k - \alpha \nabla f(x^k))$. The most interesting fact regarding these new step sizes is that they require less computational effort than the classical-steepest descent method. However, these studies have not given enough attention and emphasis on distributed optimization techniques, which have become crucial to be analyzed in many application domains.

Some recent work that relies on gradient methods can be found in [8, 40, 54, 55]. In these studies, the gradient method has been applied with the use of distributed techniques. In [8], the authors have investigated fundamental properties of distributed optimization based on gradient methods, where gradient information is communicated using a limited number of bits. It is a well-known fact that message exchange between subsystems is a common phenomenon in distributed optimization settings. However, perfect message exchange between subsystems is not possible due to limited communication bandwidths between subsystems. Therefore, quantized information tends to be exchanged between users in networked systems, which led to the exploration of new findings on quantized distributed techniques. The study [8] is a very good initiative in this regard. This piece of work has studied a general class of quantized gradient methods where the gradient direction is approximated by a finite quantization set, to optimize a constrained convex optimization problem. Here, they have considered optimization problems of the form as follows:

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } x \in \mathcal{X}, \end{aligned} \tag{18}$$

where f is convex and differentiable with L-Lipschitz continuous gradient, $x \in \mathbb{R}^n$, $\mathcal{X}$ is closed and convex set, and the optimal solution set $\mathcal{X}^*$ is nonempty and bounded.

To solve problem (18), they have used the projected gradient method as follows:

$$x^{k+1} = \left[x^k - \alpha_k d^k\right]_{\mathcal{X}}, \tag{19}$$

where d^k is quantized gradient information coded using limited number of bits. In this paper, authors have proposed two types of quantization schemes, namely, binary quantization and proper quantization.

(a) *Binary Quantization*. In this quantization scheme, the quantization set is taken as $\mathcal{D} = \{1/\sqrt{n}\,(e_1, e_2, \ldots, e_n) | e_i \in \{-1, 1\}\}$, where $d^k = \text{sign}(\nabla f(x^k))$. A convergence proof of method (19) was given under this binary quantization when $\mathcal{X} = \mathbb{R}^n$ and $\mathcal{X} = \mathbb{R}^n_+$. These convergence results are very important as they can deal with a dual problem of form (18) associated with equality and inequality constrained primal problems.

(b) *Proper Quantization*. When the above discussed binary quantization is used to solve TCP problems, the related quantized gradients are transmitted using n bits. There are many applications, where the dual problem is maintained by an individual coordinator [18, 19]. Therefore, it is worth seeking to analyze whether it is possible to use less number of bits than n when an individual coordinator exerts the problem. This fact motivates authors in [8] to discuss about the proper quantization. Here, we like to highlight the following two definitions they have used to establish their results.

Definition 1 (see [8]). A finite set $\mathcal{D}$ is a proper quantization for problem (18); if for every initialization $x^0 \in \mathcal{X}$ in iterates (19), we can choose $d^k \in \mathcal{D}$ and $\alpha_k > 0, \forall k \in \mathbb{N}$, s.t $\lim_{k \longrightarrow \infty} (\inf_{x^* \in \mathcal{X}} \|x^* - x^k\|) = 0$.

Definition 2 (see [8]). The finite set $\mathcal{D}$ is a $\theta-$cover if $\theta \in [0, \pi/2)$ and $\forall g \in \mathcal{S}^{n-1}$, $\exists d \in \mathcal{D}$ s.t $\text{ang}(g, d) = \cos^{-1}(\langle g, d \rangle) \leq \theta$, where $\mathcal{S}^{n-1}$ represents the unit sphere in $\mathbb{R}^n$. It has been proved that $\theta-\text{cover} \subseteq \mathcal{S}^{n-1}$ is a proper quantization for the problem class (18), and the minimal proper quantization is $n + 1$ [8].

Authors in [54] have introduced two measures of communication complexity of dual decomposition, which help to identify the communication overhead required by limited communication networks. The first measure determines the smallest number of bits needed to find a solution within a given accuracy, while the second measure quantifies the best possible solution accuracy when a fixed amount of bits were communicated. Furthermore, in this same work, the authors have studied a quantization scheme (introduced as Primal-Feasible quantization scheme) which guaranteed primal feasibility at each iteration in their method.

5. The Subgradient Method

Subgradient method is basically used to minimize nondifferentiable convex problems. Nondifferentiable or nonsmooth functions are one important class of problems that arise in many applications of mathematical programming, such as game theory, multicriteria models, nonlinear programming problems, optimal control problems with continuous or discrete time, and integer and mixed integer programming problems [56]. Subgradient methods are first-order methods. Their performance highly depends on problem scaling and conditioning, whereas Newton's method and interior-point methods are not dependent on problem scaling [57].

Before entering into the topic of subgradient methods, we would like to discuss about subgradients, which can be introduced as a generalized concept of gradients. When a function is nondifferentiable, the gradient of the function at nondifferentiable points cannot be found uniquely. Therefore, a well-defined way to express the slope of the function at those nondifferentiable points is required, mainly in optimization theory. Thus, getting a better understanding of subgradients is essential in the field of optimization theory. Reference [56] gives a very good exposition of the concept of subgradients, and it provides many important theoretical

aspects related to subgradients. Polyak's text [51] and the text [6] of Bertsekas are two other good references that discuss subgradients and subgradient methods. Next, we will define a subgradient of a convex function.

Definition 3. A vector $g \in \mathbb{R}^n$ is a subgradient of $f: \mathbb{R}^n \longrightarrow \mathbb{R}$ at $x \in \mathrm{dom} f$ if for all $y \in \mathrm{dom} f$,

$$f(y) \geq f(x) + g^T(y - x). \tag{20}$$

The set of all subgradients of f at x is called the subdifferential of f at x and denoted by ∂f. If f is differentiable, then its subgradient at x is unique and it is the gradient of f at x.

5.1. The Basic Subgradient Method. We consider the same form of the unconstrained optimization problem (13) considered in Section 4. The objective function $f(x)$ is still convex but not necessarily differentiable. Then, the subgradient method used to solve this problem can be given by the following iterative sequence starting at some initial point x^0:

$$x^{k+1} = x^k - \alpha_k g^k, \tag{21}$$

where x^k is the kth iterate, g^k is an any subgradient of f at x^k, and $\alpha_k > 0$ is the step size related to kth iteration. The subgradient method (21) can be considered as an extension of the gradient method (14). The difference is that, in each iteration, we use a subgradient g^k of the function $f(x)$ at x^k instead of $\nabla f(x^k)$ in (14). Moreover, the step size selection in the subgradient method is much different to the gradient method. In [57], Boyd has given five basic step size rules, namely, constant step size, constant step length, square summable but not summable, nonsummable diminishing, and nonsummable diminishing step length. From theses five step size rules, we present three common ones as follows:

(1) A constant step size, $\alpha_k = \alpha$ is a positive constant and independent of k.

(2) Square summable but not summable: the step sizes satisfy

$$\begin{aligned} \alpha_k &\geq 0, \\ \sum_{k=1}^{\infty} \alpha_k^2 &< \infty, \\ \sum_{k=1}^{\infty} \alpha_k &= \infty. \end{aligned} \tag{22}$$

For example, $\alpha_k = 1/k$.

(3) Nonsummable diminishing: the step sizes satisfy

$$\begin{aligned} \alpha_k &\geq 0, \\ \lim_{k \longrightarrow \infty} \alpha_k &= 0, \\ \sum_{k=1}^{\infty} \alpha_k &= \infty. \end{aligned} \tag{23}$$

For example, $\alpha_k = 1/\sqrt{k}$.

Above choices for the step size α_k do not depend on details computed during the subgradient algorithm. This fact differs from the step size rules found in standard descent methods, which uses current point and search direction. Good discussions on descent methods can be found in chapter 9 of [5] and chapter 8 of [58]. We can find many other choices for step size α_k in addition to the choices mentioned above. In [51], Polyak has shown that the subgradient method (21) cannot converge rapidly under diminishing nonsummable step size rule. Therefore, the author has described another variant of the subgradient method, by introducing a different step size rule that depends on f^*, the optimal value of $f(x)$. We introduce this step size in Theorem 4.

Next, we discuss the convergence of the subgradient method (21) that relies on Boyd's step size rules mentioned above. We use the following assumptions to discuss the convergence:

Assumption 1. Optimal set $\mathcal{X}^*$, the set of minimizers of problem (13) is nonempty

Assumption 2. $\|g^k\|$ is bounded

Assumption 3. The number K s.t $\|x^0 - x^*\| \leq K$ is known, where $x^* \in \mathcal{X}^*$ and x^0 is the initial point of the algorithm

Theorem 3 (see [57]). *Let Assumptions 1, 2, and 3 hold and let $f_{best}^k = \min_{i=1,\ldots,k} f(x^i)$. Then, in method (21), the following inequality holds:*

$$f_{\text{best}}^k - f^* \leq \frac{R^2 + G^2 \sum_{i=1}^k \alpha_i^2}{2 \sum_{i=1}^k \alpha_i}, \tag{24}$$

where R is s.t $\|x^0 - x^*\| \leq R$ and G is s.t $\|g^k\| \leq G$ for all k.

The proof of Theorem 3 can be found in Section 3.2 of [57]. Using this theorem, one can show that the subgradient method converges within some range of the optimal value f^*, for constant step size and constant step length. For other variants of the step size, square summable but not summable, nonsummable diminishing, and nonsummable diminishing step lengths, the subgradient method converges exactly to the optimal value without incurring any error. We discuss the convergence of the basic subgradient method empirically, in the numerical results section with the above presented three step size rules. In Section 8.2, we use a constrained optimization problem, and we dedicate our attention to discussing the convergence using dual decomposition. Next, we state the following theorem which gives the convergence of the subgradient method using Polyak's step length.

Theorem 4 (see [51]). *Let the set of minimizers $\mathcal{X}^*$ of problem (13) (with nondifferentiable f) is nonempty and $\alpha_k = (f(x^k) - f^*)/\|\partial f(x^k)\|^2$. Then, in method (21), $x^k \longrightarrow x^* \in \mathcal{X}^*$.*

The proof of above theorem is given by Polyak in his book [51]. Now, we discuss and analyze some studies done on subgradient methods. In [22], authors have considered

a subgradient method to optimize a sum of convex objective functions corresponding to multiple agents. This work analyzes large-scale networked systems, where it is essential to design decentralized resource allocation methods, since the centralized solution methods are not suitable. This paper has considered a scenario where agents cooperatively minimize a common additive cost. The corresponding optimization problem can be posed as follows:

$$\begin{aligned} &\text{minimize} \sum_{i=1}^{m} f_i(x) \\ &\text{subject to } x \in \mathbb{R}^n, \end{aligned} \tag{25}$$

where the function f_i: $\mathbb{R}^n \longrightarrow \mathbb{R}$ represents the cost function of agent i, which is convex and not necessarily to be differentiable, and $x \in \mathbb{R}^n$ is the decision vector. To analyze this problem, authors have proposed the following subgradient method:

$$x^i(k+1) = \sum_{j=1}^{m} w_j^i(k) x^j(k) - \alpha^i(k) d_i(k); \quad i = 1, \ldots, m, \tag{26}$$

where w_j^i represents the weight that agent i assigns to the information x^j received from a neighboring agent j and the scalar $\alpha^i(k) > 0$ represents the step size used by agent i. The vector $d_i(k)$ is a subgradient of agent i's objective function $f_i(x)$ at $x = x^i(k)$. Next, to analyze the convergence of method (26), they have used a different representation of that method in a way that each iteration $x^i(k+1)$ can be estimated using the information $w_j^i(s)$ and estimates $x^i(s)$, where $i, j = 1, \ldots, m$ and $s \le k$. In this study, the authors have considered an unconstrained optimization problem, but in general, this problem can be viewed in a more advanced setting, in the presence of constraints. This fact motivates readers to extend this seminal work done by Nedić and Ozdaglar to a different path of research, which will lead to a different line of convergence analysis. Furthermore, their model assumes that agents can exchange exact information, which is not possible in practice due to limited communication bandwidths. Therefore, the information is usually quantized before being sent, and it is considered that the quantization reduces the communication cost in networked control systems [59–61].

In [11], authors have considered the distributed subgradient method discussed in [22] and they have presented improved convergence results. Furthermore, they have shown that upper bounds for the difference between the estimated objective function value and the exact optimal value of the problem have a polynomial dependence on the number of agents m, by using results of their prior work [62]. We can view these bounds as improved versions of error bounds obtained in studies [22, 42], which involve exponential dependence on m. Moreover, the authors have studied the subgradient method when the communicated information is quantized to address the issue that perfect message exchange between agents cannot be performed. Some other works related to the same line of research are [9, 10, 12].

5.2. Projected Subgradient Method. Projected subgradient method is an extension of the basic subgradient method used in constrained optimization problems. Consider the optimization problem of the form

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } x \in \mathcal{X}, \end{aligned} \tag{27}$$

where f and $\mathcal{X}$ are convex. Then, the projected subgradient method can be given by

$$x^{k+1} = \left[x^k - \alpha_k g^k\right]_{\mathcal{X}}, \tag{28}$$

where g^k is any subgradient of f at x^k. Convergence of method (28) can be attained under the same step size rules described under the basic subgradient method [57].

Authors in [43] have presented distributed algorithms to solve a constrained consensus problem and a constrained optimization problem. They have used a distributed projected subgradient method to solve the constrained optimization problem, which consist of minimizing a sum of convex local objective functions. They have shown that their method converges to the optimal solution with square summable but not summable step size rule. In [24], Madan and Lall have proposed two distributed projected subgradient methods to find an optimal routing flow to maximize the network lifetime in a partially and fully decentralized manner. In their solution, subgradient methods have been applied with their dual problem. We noticed that most of the studies performed on distributed optimization have used their original primal objective function in the optimization process. They have not shown much interest on duality theory, which provide many advantages in solving constrained optimization problems. Under these circumstances, Madan's and Lall's work [24] provides immense value addition to the study of distributed optimization.

6. Alternating Direction Method of Multipliers

ADMM is a simple but strong method that is used in distributed optimization [32]. ADMM is a variant of augmented Lagrangian and method of multipliers that uses the decomposability of dual ascent. In [32], augmented Lagrangian and method of multipliers are discussed under the following equality constrained optimization problem:

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } Ax = b, \end{aligned} \tag{29}$$

where $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, and f is convex. Then, the augmented Lagrangian for problem (29) is given by

$$L_p\left(x, \lambda^T\right) = f(x) + \lambda^T (Ax - b) + \left(\frac{p}{2}\right) \|Ax - b\|^2, \tag{30}$$

where $p > 0$ is known as the penalty parameter. Then, the corresponding dual function is given by $g_p(\lambda) = \inf_x L_p(x, \lambda)$. The authors have used the gradient method to minimize negative $g_p(\lambda)$ with penalty parameter p as the step size. The method of multipliers can be viewed as more

robust version of the dual ascent method, and it yields convergence under more general conditions than the dual ascent. However, "when f is separable, L_p is not separable" is the fact that the authors in [32] have concerns with. When f is not separable, the minimization process cannot be continued in parallel, and hence, the method of multipliers cannot be used in dual decomposition. Therefore, an alternative way of observing problem (29) is needed, and consequently the ADMM has been introduced to address this issue. ADMM is a method well suited for distributed optimization settings that consist of large-scaled problems. In [32], authors have considered another variation of problem (29) as follows, to view it in separable form which has then led to the introduction of ADMM:

$$\begin{aligned} &\text{minimize } f(x) + g(y) \\ &\text{subject to } Ax + By = c, \end{aligned} \tag{31}$$

where $x \in \mathbb{R}^n, y \in \mathbb{R}^m, A \in \mathbb{R}^{q\times n}, B \in \mathbb{R}^{q\times m}$, and $c \in \mathbb{R}^q$. Moreover, f and g are convex functions. Then, the distributed algorithm for ADMM can be given using Algorithm 1.

There are many early studies done on the method of multipliers and ADMM [63, 64]. Some recent studies done on ADMM can be found in [65–69]. In [65], Erseghe has proposed a fully distributed algorithm for optimal power flow using ADMM. In this paper, the author has introduced another variation on ADMM Algorithm 1 with assumptions such as $g(y) = 0$, where y is contained in a linear space with associated orthogonal projector and also with certain assumptions on initial choices. In the study [66], authors have presented a decomposed solution approach with ADMM to solve a cost minimization problem, where the objective consists of energy and battery degradation cost. This work has used a modified version of ADMM, which helps to reduce the computations cost and ensures the stability of the solution. Most of the researchers including the ones mentioned above who have worked on ADMM have no concerns on noises that can be embedded in their models due to different types of errors occurring in practice, for an example, due to limited communication bandwidths. This fact motivates readers to work on this path with ADMM.

7. Distributed Optimization with Noise

The distributed methods for solving optimization problems can be applied in pure form only if errors and inaccuracies are fully avoided, which is hardly possible in the real world. As an example, errors or noises can occur due to inexact computation or measurement of subgradients and function values, sparsification [70], and quantization [8, 71]. The noise can be deterministic or random according to the behaviour of the application domain. Most of the real world problems consist of large-scale networked systems and mostly solve a common objective function interactively. In such situations, subsystems have to exchange their private information with neighboring subsystems during the optimization process. However, the subsystems may not be able to communicate exact information due to several reasons such as security measures and communication overheads. Therefore, it is very important to analyze distributed methods with noise imposed on the system.

7.1. Distributed Methods with Noise for Optimizing Smooth Functions. In distributed methods for optimizing differentiable (smooth) functions, we always deal with a computation of the gradient, and instead of the exact value of the gradient $\nabla f(x^k)$, we may have it computed with error

$$s^k = \nabla f\left(x^k\right) + r^k, \tag{32}$$

where r^k is introduced to be the noise. In chapter 4 of [51], Polyak has discussed four types of most important classes of noise:

(1) Absolute deterministic noise: r^k is deterministic and satisfies the boundedness condition $\|r^k\| \le \varepsilon$

(2) Relative deterministic noise: r^k is deterministic and satisfies the condition $\|r^k\| \le \varepsilon\|\nabla f(x^k)\|$

(3) Absolute random noise: r^k is random, independent, centered, and has bounded variance, $E[r^k] = 0$ and $E[\|r^k\|] \le \sigma^2$

(4) Relative random noise: r^k satisfies the condition $E[r^k] = 0, E[\|r^k\|^2] \le \tau\|\nabla f(x^k)\|^2$

In the above classes of noise, ε, σ, and τ represent positive constants. In the same text [51], the convergence of the gradient method (14) was discussed, where the gradient is computed with error as given in (32). Here, the convergence properties of the gradient method were analyzed under all four types of errors mentioned above, under the assumption that the objective function is strongly convex and with a gradient satisfying a Lipschitz condition.

Most of the related literatures available to solve optimization problems with the use of gradient like methods under the presence of noise were analyzed under boundedness assumptions on the objective function and the decision variable or show only $\lim_{k\longrightarrow\infty}\inf\|\nabla f(x^k)\| = 0$ [72–75]. Authors in [55] discussed convergence results for the following method, by removing various boundedness conditions such as boundedness from below of f, boundedness of $\nabla f(x^k)$, or boundedness of x^k:

$$x^{k+1} = x^k + \alpha_k\left(s^k + w^k\right), \tag{33}$$

where s^k represents a descent direction of a function $f\colon \mathbb{R}^n \longrightarrow \mathbb{R}$ and w^k is a deterministic or stochastic error. They first focus on the above method with deterministic error, with w^k satisfying following conditions:

$$\left\|w^k\right\| \le \alpha_k\left(q + p\left\|\nabla f\left(x^k\right)\right\|\right), \tag{34}$$

Given initial λ.
Set $k = 1$.
while (stopping criterion)
(1) x minimization step:
minimize$_x L_p(x, y^k, \lambda^k)$
Let $x^* = \text{argmin}_x L_p(x, y^k, \lambda^k)$.
(2) y minimization step:
minimize$_y L_p(x^*, y, \lambda^k)$
Let $y^* = \text{argmin}_y L_p(x^*, y, \lambda^k)$.
(3) Dual variable update:
$\lambda^{k+1} := \lambda^k + p(Ax^* + By^* - c)$

Algorithm 1: Alternating direction method of multipliers (ADMM).

where p and q are some positive scalars. Then, the convergence of method (33) was obtained using following theorem.

Theorem 5 (see [55]). *Suppose that s^k in method (33) is a descent direction satisfying for some positive scalars c_1 and c_2, and for all k,*

$$c_1\left\|\nabla f\left(x^k\right)\right\|^2 \leq -\nabla f\left(x^k\right)^T s^k, \qquad \left\|s^k\right\| \leq c_2\left(1 + \left\|\nabla f\left(x^k\right)\right\|\right). \tag{35}$$

Then, for $\alpha_k > 0$ with square summable but not summable step size rule, method (33) guaranteed to convergent to the optimal solution.

Next, the authors have obtained convergence results for minimizing a sum of large number of functions using incremental gradient methods. Moreover, they have focused on stochastic gradient methods. In the recent study [68], authors have analyzed the convergence of distributed ADMM for consensus optimization in the presence of random error. They have presented lower and upper bounds on the mean squared steady state error of the algorithm when individual objective functions are strongly convex and when the gradients are Lipschitz continuous. Furthermore, authors have presented that steady state error of their noisy ADMM algorithm is bounded when they have a bounded random error and when individual objectives are proper, closed, and convex.

7.2. Distributed Methods with Noise for Optimizing Nonsmooth Functions. In chapter 5 of [51], Polyak has introduced the well-known subgradient method of optimizing nondifferentiable (nonsmooth) problems with noise,

$$x^{k+1} = x^k - \alpha_k s^k, \qquad s^k = \partial f\left(x^k\right) + r^k, \tag{36}$$

where r^k is the noise imposed on the subgradient. The convergence results of the noisy subgradient method (36) have been discussed by the same author under the same classes of noises discussed in the previous subsection. In the early study [76], Polyak has studied minimization methods of a nonlinear function with nonlinear constraints when the values of the objective function, constraints, and gradients are computed with errors. In [77], authors have studied the effect of noise on subgradient methods for convex constrained optimization problems of form (27). They have discussed the convergence properties of the following projected subgradient method when the noise is deterministic and bounded:

$$x^{k+1} = \left[x^k - \alpha_k \widetilde{g}^k\right]_{\mathcal{X}}, \tag{37}$$

where $\widetilde{g}^k$ is an approximate subgradient of the form $\widetilde{g}^k = g^k + r^k$, where r^k is the noise and g^k is an e_k subgradient of f at x^k for some $e_k \geq 0$. Convergence properties of method (37) have been analyzed under three step size rules, namely, constant step size rule, diminishing step size rule, and dynamic step size rule which is given by

$$\alpha_k = \gamma_k \frac{\widetilde{f}(x_k) - \widetilde{f}_k^{\text{lev}}}{\left\|\widetilde{g}_k\right\|^2}, \quad 0 < \gamma \leq \gamma_k \leq 2, \forall k \geq 0, \tag{38}$$

where $\widetilde{f}(x_k)$ is an error involved function value and $\widetilde{f}_k^{\text{lev}}$ is a target level approximating the optimal value f^*. First, the convergence of method (37) has been obtained when the constrained set is compact. Secondly, the authors have analyzed their method using a convex objective function which has a sharp set of minima. The important results observed by authors were as follows: (a) in the first scenario, the method converges to the optimal value with some tolerance and (b) in the second scenario, the method converges exactly to the optimal value without any error.

It is very important to pay attention to the stochastic optimization processes since many practical problems cannot be viewed as deterministic structures. Some studies that paid attention to this particular area can be found in [76, 78]. Authors in [78] have studied stochastic quasigradient methods which allow solving optimization problems without calculating exact values of objectives and constraints. In [76], a general convex problem with noise was solved with assumptions as follows:

(i) The objective function and inequality constraint functions are convex continuous

(ii) Feasible set is a convex closed bounded set

(iii) Slater condition holds

(iv) All noises are with mean zero with bounded variance and are independent at different points

8. Numerical Results

In this section, we discuss the convergence of the gradient method, subgradient method, and ADMM empirically by using some numerical examples.

8.1. Example 1 (Gradient Method: Primal Decomposition). Here, we consider an unconstrained minimization problem with two users as follows:

$$\text{minimize } f(x_1, x_2, y) = f_1(x_1, y) + f_2(x_2, y), \tag{39}$$

where $f_1(x_1, y) = \begin{bmatrix} x_1^T & y^T \end{bmatrix} A_1 \begin{bmatrix} x_1^T & y^T \end{bmatrix}^T$ and $f_2(x_2, y) = \begin{bmatrix} x_2^T & y^T \end{bmatrix} A_2 \begin{bmatrix} x_2^T & y^T \end{bmatrix}^T$ with $x_1, x_2 \in \mathbb{R}^{n_1}$ and $y \in \mathbb{R}^{n_2}$. Here, $A_1 \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)}$ and $A_2 \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)}$ are positive definite matrices. We use primal decomposition and analyze the convergence of the gradient method (14) for this problem with the use of Theorem 1. The subproblems related to (39) can be given as follows:

$$\textit{Subproblem 1: } S_1(y) = \underset{x_1}{\text{minimize}} \begin{bmatrix} x_1^T & y^T \end{bmatrix} A_1 \begin{bmatrix} x_1^T & y^T \end{bmatrix}^T$$

$$\textit{Subproblem 2: } S_2(y) = \underset{x_2}{\text{minimize}} \begin{bmatrix} x_2^T & y^T \end{bmatrix} A_2 \begin{bmatrix} x_2^T & y^T \end{bmatrix}^T$$

Then, the master problem corresponding to (39) is given by

$$\underset{y}{\text{minimize}}\, S(y) = S_1(y) + S_2(y). \tag{40}$$

Analytically, by solving the subproblems, we can show that $S_i(y) = y^T A_{i4} y + (x_i^{*T} A_{i3}^T + x_i^{*T} A_{i2}) y + x_i^{*T} A_{i1} x^*$, where $x_i^* = \text{argmin}_{x_i} \begin{bmatrix} x_i^T & y^T \end{bmatrix} A_i \begin{bmatrix} x_i^T & y^T \end{bmatrix}^T$ and $A_i = \begin{bmatrix} A_{i1} & A_{i2} \\ A_{i3} & A_{i4} \end{bmatrix}$ with $A_{i1} \in \mathbb{R}^{n_1\times n_1}, A_{i2} \in \mathbb{R}^{n_1\times n_2}, A_{i3} \in \mathbb{R}^{n_2\times n_1}$, and $A_{i4} \in \mathbb{R}^{n_2\times n_2}$ for $i = 1, 2$. Then, $S(y)$ is quadratic as $S_1(y)$ and $S_2(y)$ are quadratic. Moreover, $S_1(y)$ and $S_2(y)$ are strongly convex since A_{14} and A_{24} are positive definite. Hence, $S(y)$ is also strongly convex and $\nabla S(y)$ is Lipschitz continuous. Therefore, we can apply Theorem 1 to solve problem (40) using the gradient method (14). We use Algorithm 2 to solve (40). In this algorithm, at each iteration, the gradient update is given by $\nabla S(y^k) = \nabla S_1(y^k) + \nabla S_2(y^k)$, where $\nabla S_1(y^k) = A_{12}^T x_1^k + A_{13} x_1^k + 2A_{14} y^k$ and $\nabla S_2(y^k) = A_{22}^T x_2^k + A_{23} x_2^k + 2A_{24} y^k$.

First, we illustrate our results with scalar valued primal variables x_1, x_2, and y ($n_1 = n_2 = 1$ case) for different values of constant step sizes $\alpha_k = \alpha$. Figure 2 shows the convergence of y^k with $\alpha = 0.001$, $\alpha = 0.01$, $\alpha = 0.1$, and $\alpha = 0.5$. Next, we show the convergence results for different dimensions of the complicating variable y with $x_1 \in \mathbb{R}^{10}$ and $x_2 \in \mathbb{R}^{10}$. Figure 3 shows the convergence of the residuals $\|y^k - y^*\|$ with step size $\alpha = 0.001$, for $y \in \mathbb{R}, y \in \mathbb{R}^2, y \in \mathbb{R}^3, y \in \mathbb{R}^5$, and $y \in \mathbb{R}^{10}$, where y^* represents the optimal value of y. We present Figure 4, which indicates log values of $\|y^k - y^*\|$, to analyze the convergence of residuals when they approach to zero. For this same set of dimensions of y with same step size, the convergence of iterates $S(y^k)$ is shown under Figure 5. Moreover, Figure 6 shows that the primal variable iterates x_1^k and x_2^k converge exactly to their optimal solutions using $\alpha_k = 0.001$ and $y \in \mathbb{R}$.

8.2. Example 2 (Subgradient Method: Dual Decomposition). Here, we focus on a problem which is not quadratic. We consider the problem in the following form with two users:

$$\begin{aligned} &\text{minimize } f(x_1, y_1, x_2, y_2) = f_1(x_1, y_1) + f_2(x_2, y_2) \\ &\text{subject to } y_1 = y_2, x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2, y_1, y_2 \in \mathcal{Y}, \end{aligned} \tag{41}$$

where $f_1(x_1, y_1) = \cosh(a_1^T x) + a_2^T x$ and $f_2(x_2, y_2) = \cosh(b_1^T y) + b_2^T y$ with $x = (x_1, y_1)$, $y = (x_2, y_2)$, $\mathcal{X}_1 \subseteq \mathbb{R}^{n_1}$, $\mathcal{X}_2 \subseteq \mathbb{R}^{n_1}$, $\mathcal{Y} \subseteq \mathbb{R}^{n_2}$, and $a_1, a_2, b_1, b_2 \in \mathbb{R}^{(n_1+n_2)}$. Here, we intend to solve this problem in a fully distributed manner using dual decomposition. We implement our results for $n_1 = n_2 = 1$ (scalar valued variables). We consider $\mathcal{X}_1 = [-1, 0], \mathcal{X}_2 = [1, 2], \mathcal{Y} = [-2, 2], a_1 = [1, 1]^T, a_2 = [3, -2]^T$, $b_1 = [1, 1]^T$, and $b_2 = [-2, 5]^T$. The dual function corresponding to the primal problem (41) is given by

$$g(\lambda) = \inf_{\substack{x_1\in\mathcal{X}_1 \\ x_2\in\mathcal{X}_2 \\ y_1, y_2\in\mathcal{Y}}} \left(f_1(x_1, y_1) + f_2(x_2, y_2) + \lambda^T (y_1 - y_2) \right), \tag{42}$$

and we use corresponding subproblems in dual decomposition as follows:

$$\begin{aligned} \text{Subproblem 1: } g_1(\lambda) &= \inf_{x_1\in\mathcal{X}_1, y_1\in\mathcal{Y}} f_1(x_1, y_1) + \lambda^T y_1, \\ \text{Subproblem 2: } g_2(\lambda) &= \inf_{x_2\in\mathcal{X}_2, y_2\in\mathcal{Y}} f_2(x_2, y_2) - \lambda^T y_2. \end{aligned} \tag{43}$$

Then, the dual problem corresponding to the primal problem (41) is given by $\text{maximize}_{\lambda\in\mathbb{R}^n}\, g(\lambda) = g_1(\lambda) + g_2(\lambda)$. We know that $g(\lambda)$ is always concave (see chapter 05 of [5]). We have obtained the graph of $g(\lambda)$ as given in Figure 7. This figure also confirms the concavity of $g(\lambda)$. Moreover, this figure shows that $g(\lambda)$ is nondifferentiable as it has a sharp point around $\lambda = 5$. Hence, $-g(\lambda)$ is convex and nondifferentiable, and therefore we use subgradient method (21) to minimize $-g(\lambda)$ using Algorithm 3.

We analyze the convergence results of the subgradient method using Theorem 3 discussed under Section 5. Therefore, we have to check whether Assumptions 1–3 used in Theorem 3 hold for our particular problem considered here. Figure 7 shows that there exists an optimal solution λ^*

Given initial y, y^0.
Set $k = 0$.
while (stopping criterion)
(1) x_1 and x_2 minimization steps:
Step 1: $x_1^k = \text{argmin}_{x_1} \begin{bmatrix} x_1^T & y^{kT} \end{bmatrix} A_1 \begin{bmatrix} x_1^T & y^{kT} \end{bmatrix}^T$
Step 2: $x_2^k = \text{argmin}_{x_2} \begin{bmatrix} x_2^T & y^{kT} \end{bmatrix} A_2 \begin{bmatrix} x_2^T & y^{kT} \end{bmatrix}^T$
(2) gradient information update:
$\nabla S(y^k) = A_{11}^T x_1^k + A_{13} x_1^k + 2A_{14} y^k + A_{21}^T x_2^k + A_{23} x_2^k + 2A_{24} y^k$
(3) y variable update:
$y^{k+1} = y^k + \alpha \nabla S(y^k)$

Algorithm 2: Gradient method: primal decomposition.

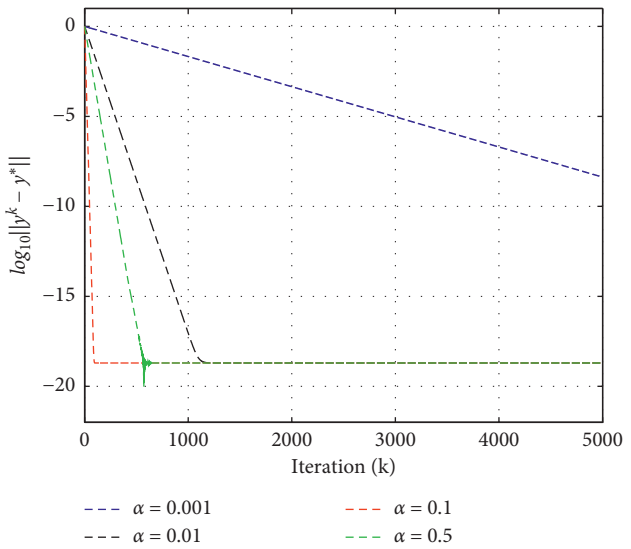

Figure 2: Convergence of $\log_{10}\|y^k - y^*\|$ using different constant step sizes in the gradient method with primal decomposition. The figure shows that slow convergence for relatively small step sizes.

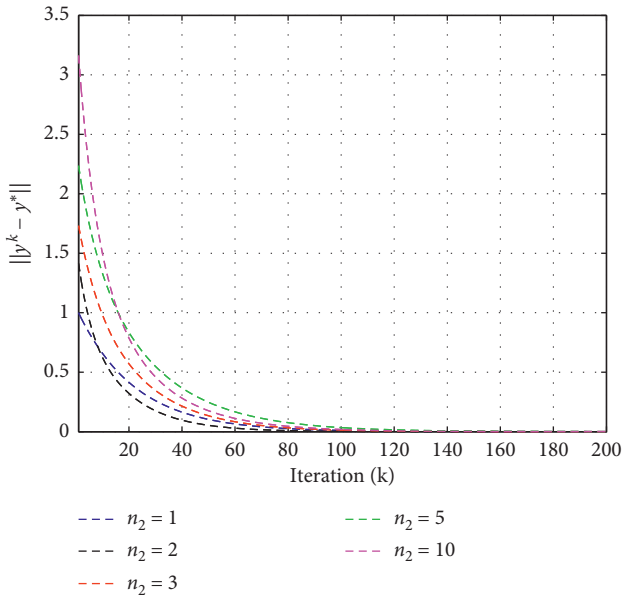

Figure 3: Convergence of the residuals $\|y^k - y^*\|$ for different dimensions of y when the primal problem (39) is solved using the gradient method followed by primal decomposition. The figure shows that $\|y^k - y^*\| \longrightarrow 0$ as the iteration number increases. This shows that the convergence of y^k to y^* is guaranteed even for high dimensional complicating variables.

to the dual function $g(\lambda)$. Hence, Assumption 1 holds. At each iteration in Algorithm 3, $y_2^k - y_1^k$ used in the dual variable update represents a subgradient s^k of $-g(\lambda)$ at λ^k ($y_1^k - y_2^k$ represents a subgradient of $g(\lambda)$ at λ^k). We can observe that $\|s^k\| \leq 4$ as $y_1, y_2 \in \mathcal{Y} = [-2, 2]$. Hence, Assumption 2 holds. Moreover, we use the initialization $\lambda^0 = 1$, and we found that $\lambda^* \approx 5.14$ using the CVX solver in Matlab. Therefore, it turns out that $\|\lambda^0 - \lambda^*\| \approx 4.14$, from which Assumption 3 follows. Hence, we can use Theorem 3 to analyze the convergence of the subgradient method.

We have obtained the convergence results with constant, square summable but not summable, and nonsummable diminishing step size rules. Figure 8 shows the convergence of log values of $\|\lambda^k - \lambda^*\|$ for different constant step sizes. This figure shows that large step sizes give fast convergence. Next, we show the convergence with step sizes $\alpha_k = 0.1, \alpha_k = 1/k$, and $\alpha_k = 0.1/\sqrt{k}$, in the same figure (Figure 9) so as to identify the effect of different step size rules. Here, we considered the convergence up to 10^{-5} tolerance. We can observe a slower convergence using $\alpha_k = 1/k$ and $\alpha_k = 0.1/\sqrt{k}$ than that for the constant step size rule.

In our Algorithm 3, both users solve their subproblems separately and find optimal primal variables locally at each iteration. Next, they exchange their information y_1^k and y_2^k with each other and update the dual variable individually. In general, their iterates y_1^k and y_2^k are not feasible. Therefore, at each iteration, they agreed to have a feasible solution as $\overline{y}^k = (y_1^k + y_2^k)/2$. Next, by using these primal variable iterates and updated dual variable λ^k, user 1 and user 2 can compute $g_1(\lambda^k)$ and $g_2(\lambda^k)$, respectively. Then, $g(\lambda^k) = g_1(\lambda^k) + g_2(\lambda^k)$ can be calculated. This is always a lower bound on f^*, the optimal value of the primal problem [5]. Moreover, at each iteration, users can compute two upper bounds on f^* as follows [14]:

$$w\left(x_1^k, x_2^k, \overline{y}^k\right) = f_1\left(x_1^k, \overline{y}^k\right) + f_2\left(x_2^k, \overline{y}^k\right),$$
$$b\left(\overline{y}^k\right) = b_1\left(\overline{y}^k\right) + b_2\left(\overline{y}^k\right), \tag{44}$$

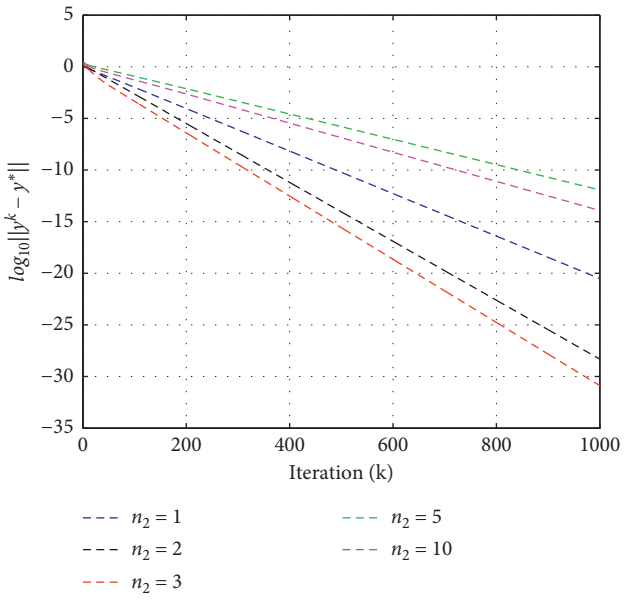

FIGURE 4: Convergence of $\log_{10}\|y^k - y^*\|$ for different dimensions of y when the primal problem (39) is solved using the gradient method followed by the primal decomposition. This shows that a high accuracy for the convergence of y^k can be achieved within 1000 iterations even for high dimensional y.

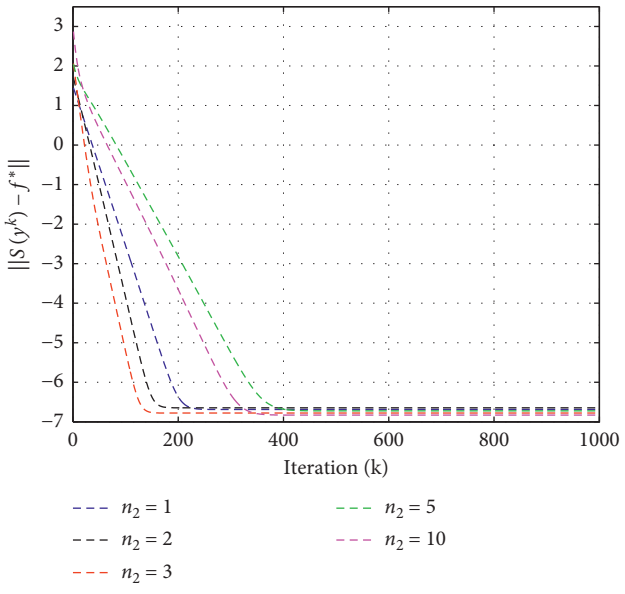

FIGURE 5: Convergence of $S(y^k)$ illustrated with different dimensions of complicating variable y. The figure shows that $S(y^k)$ converges almost to f^*, the optimal value of the primal problem (39) regardless of the dimension of y.

where $b_1(\overline{y}^k) = \text{minimize}_{x_1 \in \mathcal{X}_1} f_1(x_1, \overline{y}^k)$ and $b_2(\overline{y}^k) = \text{minimize}_{x_2 \in \mathcal{X}_2} f_2(x_2, \overline{y}^k)$. In [14], w and b are defined as the worst bound and the better bound. Worst bound represents

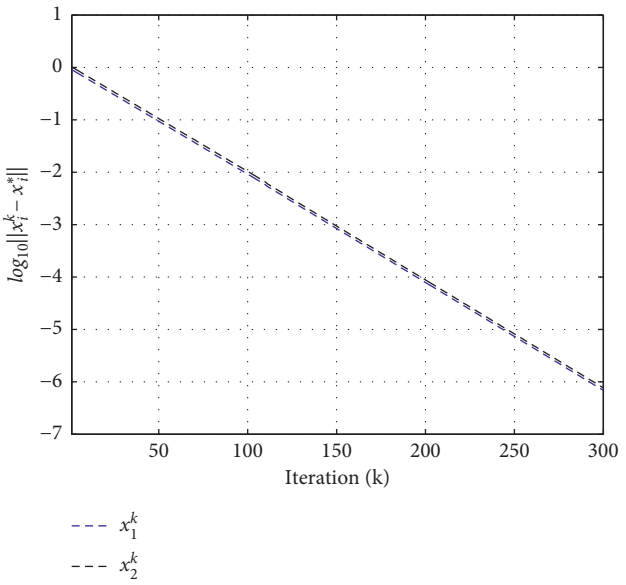

FIGURE 6: Convergence of the primal variables $x_1 \in \mathbb{R}^{10}$ and $x_2 \in \mathbb{R}^{10}$ illustrated with scalar valued complicating variable y, when the primal problem (39) is solved using the gradient method followed by primal decomposition. The figure shows x_1^k and x_2^k converge to their optimal solutions x_1^* and x_2^*, respectively.

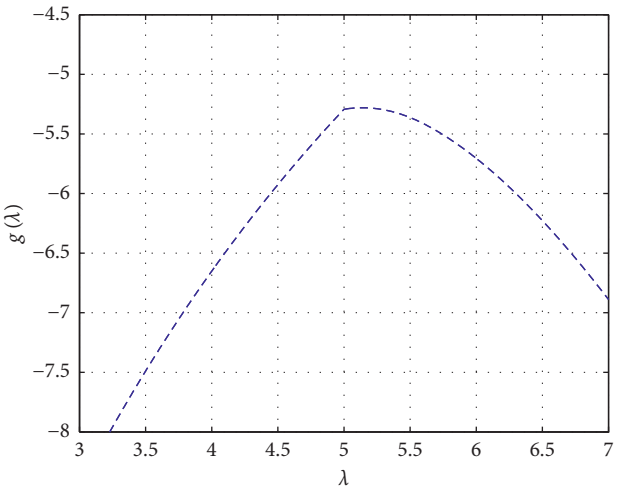

FIGURE 7: Dual function corresponding to problem (41). The figure reveals the concavity and nondifferentiability of $g(\lambda)$. λ^* attains around $\lambda = 5$.

the primal objective function values evaluated at each iteration using feasible points $(x_1^k, \overline{y}^k)$ and $(x_2^k, \overline{y}^k)$. Better bound can be obtained by replacing y_1^k and y_2^k with $\overline{y}$ and then solving subproblems involved with related primal decomposition structure of (41). Figure 10 shows the convergence of $g(\lambda^k)$, better bound, and worst bound using constant step size rule $\alpha_k = 0.1$ and scalar valued primal

Given initial λ, λ^0.
Set $k = 0$.
while (stopping criterion)
(1) Primal variables minimization steps:
Step 1: $(x_1^k, y_1^k) = \operatorname{argmin}_{x_1 \in \mathcal{X}_1, y_1 \in \mathcal{Y}} f_1(x_1, y_1) + \lambda^k y_1$
Step 2: $(x_2^k, y_2^k) = \operatorname{argmin}_{x_2 \in \mathcal{X}_2, y_2 \in \mathcal{Y}} f_2(x_2, y_2) - \lambda^k y_2$
(2) Dual variable update:
$\lambda^{k+1} = \lambda^k + \alpha_k (y_1^k - y_2^k)$

Algorithm 3: Subgradient method: dual decomposition.

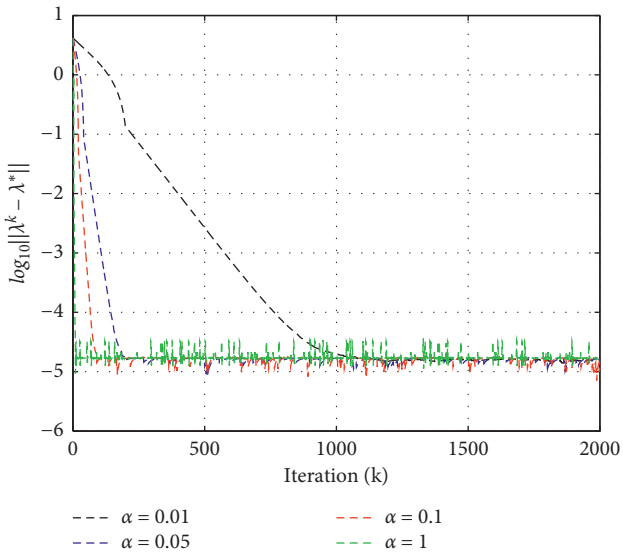

Figure 8: Convergence of $\log_{10}\|\lambda^k - \lambda^*\|$ in the subgradient method followed by dual decomposition with different constant step sizes. The figure shows a trade off, large α yields fast convergence.

variables. Here, we can observe that for this particular problem, the lower bound $g(\lambda)$ and two upper bounds converge exactly to f^*.

8.3. Example 3 (ADMM). Here, we first discuss the robustness of ADMM compared with the gradient method. Let us consider the following linear programme:

$$\begin{aligned} &\text{minimize } a^T x \\ &\text{subject to } Ax = b \\ &Bx + Cy = 0, \end{aligned} \tag{45}$$

where $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$ are decision variables of the problem, $a \in \mathbb{R}^n, A \in \mathbb{R}^{m_1 \times n}, B \in \mathbb{R}^{m_2 \times n}, C \in \mathbb{R}^{m_2 \times n}$ with $m_1, m_2 < n$ and $b \in \mathbb{R}^{m_1}$ is a constant vector. Suppose that the set of solutions of (45) is nonempty.

The dual function $g(\lambda)$, where $\lambda = \begin{bmatrix} \lambda_1^T & \lambda_2^T \end{bmatrix}^T$ with $\lambda_1 \in \mathbb{R}^{m_1}$ and $\lambda_2 \in \mathbb{R}^{m_2}$, for problem (45) is given by

$$\begin{aligned} g(\lambda) &= \inf_{x \in \mathbb{R}^n, y \in \mathbb{R}^n} \left(a^T x + \lambda_1^T (Ax - b) + \lambda_2^T (Bx + Cy) \right) \\ &= \inf_{x \in \mathbb{R}^n} \left(a + A^T \lambda_1 + B^T \lambda_2 \right)^T x + \inf_{y \in \mathbb{R}^n} \left(\lambda_2^T C y \right) - \lambda_1^T b. \end{aligned} \tag{46}$$

Then, analytically we can obtain

$$g(\lambda) = \begin{cases} -\lambda_1^T b; & a + A^T \lambda_1 + B^T \lambda_2 = 0 \text{ and } C^T \lambda_2 = 0, \\ -\infty; & \text{otherwise.} \end{cases} \tag{47}$$

Next, the dual problem is given by

$$\text{maximize } g(\lambda). \tag{48}$$

Here, we can easily observe that the optimal value of the dual problem (48) is $-\lambda_1^T b$, which is attained when $a + A^T \lambda_1 + B^T \lambda_2 = 0$ and $C^T \lambda_2 = 0$. Usually we use following subproblems when we use the gradient method to solve (48):

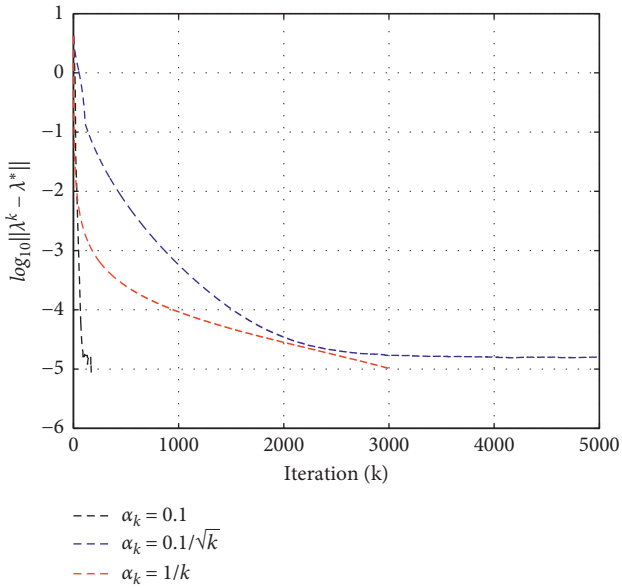

FIGURE 9: Convergence of $\log_{10}\|\lambda^k - \lambda^*\|$ in the subgradient method with the constant, nonsummable diminishing, and square summable but not summable step size rules. The figure shows a slower convergence using $\alpha_k = 0.1/\sqrt{k}$ and $\alpha_k = 1/k$ corresponding to nonsummable diminishing and square summable but not summable step size rules, respectively.

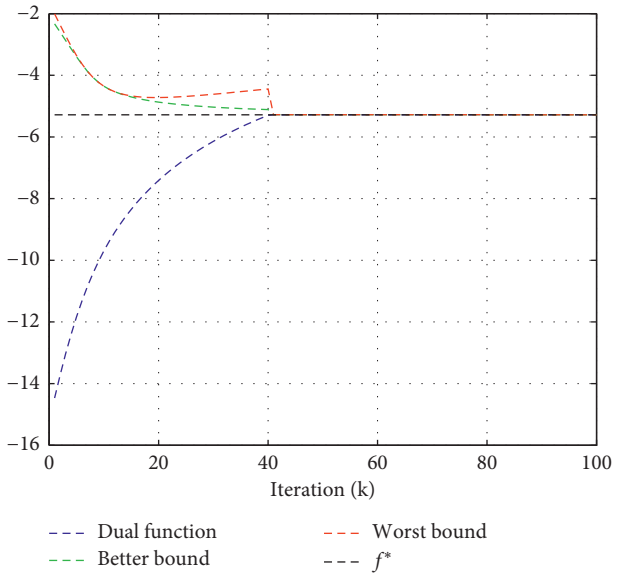

FIGURE 10: Convergence of the dual function iterates $g(\lambda^k)$, the better bound and the worst bound. The figure shows clearly that $g(\lambda)$ is being a lower bound on f^* and the better bound gives a more better approximation on f^* than that of the worst bound. However, all 3 bounds converge to the optimal value f^* of the original problem (41) for sufficiently large k.

$$\text{Subproblem 1: } g_1(\lambda_1, \lambda_2) = \inf_{x \in \mathbb{R}^n} \left(a + A^T\lambda_1 + B^T\lambda_2\right)^T x,$$
$$\text{Subproblem 2: } g_2(\lambda_1, \lambda_2) = \inf_{y \in \mathbb{R}^n} \left(\lambda_2^T C y - \lambda_1^T b\right). \tag{49}$$

Algorithm 4 represents the corresponding gradient algorithm.

We can observe that x and y minimization steps (Algorithm 4) given in Algorithm 4 cannot proceed for any arbitrarily chosen λ as $(a + A^T\lambda_1^k + B^T\lambda_2^k)^T x$ and

Given initial $\lambda, \lambda^0 = (\lambda_1^0, \lambda_2^0)$.
Set $k = 0$.
while (stopping criterion)
(1) Primal variables minimization steps:
Step 1: $x^k = \text{argmin}_{x \in \mathbb{R}^n} (a + A^T \lambda_1^k + B^T \lambda_2^k)^T x$
Step 2: $y^k = \text{argmin}_{y \in \mathbb{R}^n} (\lambda_2^{kT} Cy - \lambda_1^{kT} b)$
(2) Dual variable update:
$\lambda^{k+1} = \lambda^k + \alpha_k \nabla g(\lambda^k)$

Algorithm 4: Gradient algorithm to solve (48).

Given initial $\lambda, \lambda^0 = (\lambda_1^0, \lambda_2^0)$, and initial y, y^0.
Set $k = 0$.
while (stopping criterion)
(1) Primal variables minimization steps:
Step 1: $x^{k+1} = \arg\min_{x \in \mathbb{R}^n} (a^T x + \lambda_1^{kT} (Ax - b) + \lambda_2^{kT} (Bx + Cy^k) + (p/2)\|Ax - b\|^2 + (p/2)\|Bx + Cy^k\|^2)$
Step 2: $y^{k+1} = \arg\min_{y \in \mathbb{R}^n} (a^T x^{k+1} + \lambda_1^{kT} (Ax^{k+1} - b) + \lambda_2^{kT} (Bx^{k+1} + Cy) + (p/2)\|Ax^{k+1} - b\|^2 + (p/2)\|Bx^{k+1} + Cy\|^2)$
(2) Dual variable update:
$\lambda_1^{k+1} = \lambda_1^k + \alpha (Ax^{k+1} - b)$
$\lambda_2^{k+1} = \lambda_2^k + \alpha (Bx^{k+1} + Cy^{k+1})$

Algorithm 5: Example 3: (ADMM).

$\lambda_2^{kT} Cy - \lambda_1^{kT} b$ are unbounded below. Hence, the gradient method fails to solve (48), and therefore the linear programme (45) also cannot be solved. However, the interesting fact is that ADMM solves this problem without any issue, showing its robustness compared with the gradient method.

To solve (48) using ADMM, we consider the augmented Lagrangian as follows:

$$L(x, y, \lambda) = a^T x + \lambda_1^T (Ax - b) + \lambda_2^T (Bx + Cy) + \frac{p}{2}\|Ax - b\|^2 + \frac{p}{2}\|Bx + Cy\|^2, \tag{50}$$

where p represents the penalty parameter. Then, the corresponding dual function is given by $g(\lambda) = \inf_{x \in \mathbb{R}^n, y \in \mathbb{R}^n} L(x, y, \lambda)$. Next, we maximize $g(\lambda)$ by using Algorithm 5. In this algorithm, α represents a suitably chosen step size. Here, we discuss the convergence of iterates (Algorithm 5) with

$$a = \begin{bmatrix} -1 & -3 & -4 \end{bmatrix}^T,$$
$$A = \begin{bmatrix} 1 & 1 & 2 \\ 1 & 2 & 3 \end{bmatrix},$$
$$B = \begin{bmatrix} -2 & 1 & 3 \\ 5 & 2 & -2 \end{bmatrix}, \tag{51}$$
$$C = \begin{bmatrix} -1 & 4 & -2 \\ 1 & 5 & -3 \end{bmatrix},$$
$$b = \begin{bmatrix} 4 & 5 \end{bmatrix}^T.$$

We choose $p = \alpha = 0.1$. Figure 11 shows that our method guarantees the convergence of the dual variable λ exactly to its optimal value λ^*. Convergence of the dual function iterates $g(\lambda^k)$ and the objective function iterates $f(x^k) = a^T x^k$ is given in Figure 12, and it shows that the both dual function and objective function converge exactly to the optimal value $f^* = -6$ of our primal problem (45).

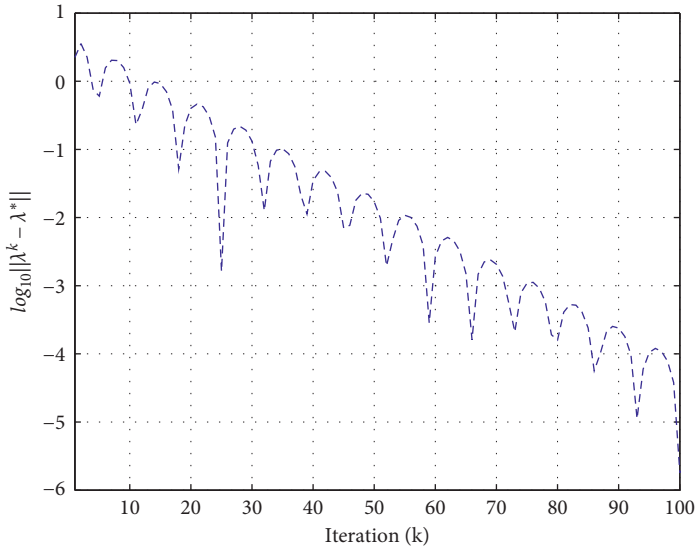

Figure 11: Convergence of the dual variable iterates λ^k in ADMM. The figure shows the log values of $\|\lambda^k - \lambda^*\|$ with iteration number. A better convergence up to 10^{-6} tolerance is achieved within 100 iterations.

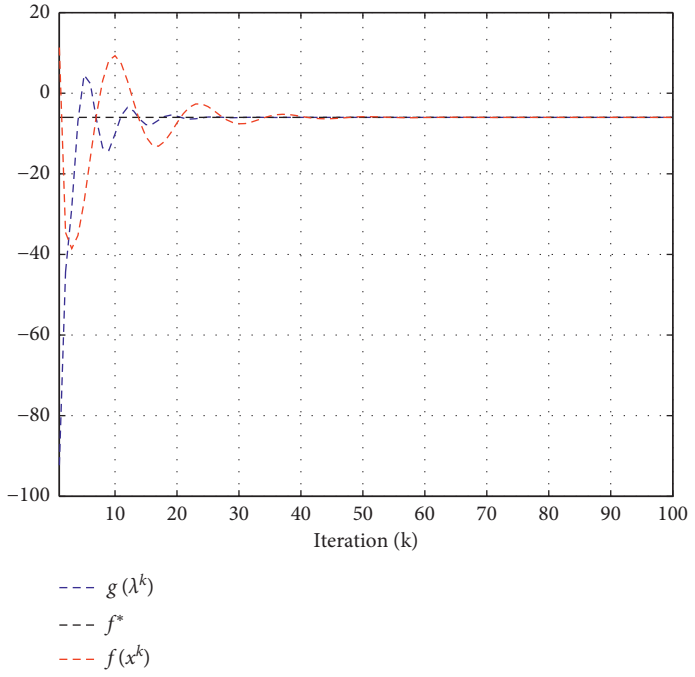

Figure 12: Convergence of the dual function and objective function iterates in ADMM. Horizontal line represents the optimal value $f^* = -6$ of the primal problem (45). The figure shows the iterates almost converge to f^* within 100 iterations.

9. Conclusions

Centralized methods are hardly being used or applied as they are not suitable or they fail to deploy in many optimization settings due to the high demanding necessity of distributed techniques among large-scale networked systems. Therefore, an attempt has been made by this paper to discuss the most important methods that currently exist to solve distributed optimization problems. A detailed analysis on gradient like methods, subgradient methods, and ADMM has been presented with numerical results. Gaps in previous studies that need to be developed to enhance the process of distributed optimization over networked systems have been discussed under each section. Here, we summarize the areas

in which future research can be conducted in distributed optimization.

(i) Many studies have shown their interest to solve distributed problems using primal measures. Therefore, more theoretical studies related to duality need to be established to make use of the advantage of optimizing more general (nonconvex) distributed problems.

(ii) Methods of finding primal solutions from the dual under more relaxed assumptions are critical, as the dual measures do not converge to primal measures in general.

(iii) Distributed methods over limited communications between networked systems need to be analyzed in depth, and related proper quantization schemes that guarantee the convergence of corresponding distributed methods should be identified.

(iv) Inexact message exchange between subsystems due to limited communication bandwidths is common in distributed optimization. Consequently, analysis of error-based distributed methods is essential over many distributed application domains.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors thank Dr. Chathuranga Weeraddana for his continuous assistance and valuable comments given during this study. The authors would also like to show their gratitude to Mr. Susil Palihakkara for his feedback and comments provided to improve the presentation of this paper.

References

[1] M. Zibulevsky and M. Elad, "L1-L2 Optimization in signal and image processing," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 76–88, 2010.

[2] Z. Luo and W. Yu, "An introduction to convex optimization for communications and signal processing," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1426–1438, 2006.

[3] S. Dedu and F. Şerban, "Multiobjective mean-risk models for optimization in finance and insurance," *Procedia Economics and Finance*, vol. 32, pp. 973–980, 2015.

[4] I. Fister, A. Iglesias, A. Galvez et al., "Novelty search for global optimization," *Applied Mathematics and Computation*, vol. 347, pp. 865–881, 2019.

[5] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.

[6] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, USA, 2nd edition, 1999.

[7] J. Partan, J. Kurose, and B. N. Levine, "A survey of practical issues in underwater networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 4, pp. 23–33, 2007.

[8] S. Magnússon, C. Enyioha, N. Li, C. Fischione, and V. Tarokh, "Convergence of limited communication gradient methods," *IEEE Transactions on Automatic Control*, vol. 63, no. 5, pp. 1356–1371, 2018.

[9] J. Li, G. Chen, Z. Wu, and X. He, "Distributed subgradient method for multi-agent optimization with quantized communication," *Mathematical Methods in the Applied Sciences*, vol. 40, no. 4, pp. 1201–1213, 2016.

[10] C. Huang, H. Li, D. Xia, and L. Xiao, "Quantized subgradient algorithm with limited bandwidth communications for solving distributed optimization over general directed multi-agent networks," *Neurocomputing*, vol. 185, pp. 153–162, 2016.

[11] A. Nedić, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "Distributed subgradient methods and quantization effects," in *Proceedings of the 2008 47th IEEE Conference on Decision and Control*, pp. 4177–4184, Cancun, Mexico, December 2008.

[12] D. Yuan, S. Xu, H. Zhao, and L. Rong, "Distributed dual averaging method for multi-agent optimization with quantized communication," *Systems & Control Letters*, vol. 61, no. 11, pp. 1053–1061, 2012, http://www.sciencedirect.com/science/article/pii/S0167691112001193.

[13] P. Yi and Y. Hong, "Quantized subgradient algorithm and data-rate analysis for distributed optimization," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 4, pp. 380–392, 2014.

[14] S. Boyd, L. Xiao, A. Mutapcic, and J. Mattingley, "Notes on decomposition methods," 2007, http://stanford.edu/class/ee364b/lectures/decomposition_notes.pdf.

[15] D. K. Molzahn, F. Dörfler, H. Sandberg et al., "A survey of distributed optimization and control algorithms for electric power systems," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2941–2962, 2017.

[16] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, no. 3, pp. 237–252, 1998.

[17] S. H. Low and D. E. Lapsely, "Optimization flow control. I. basic algorithm and convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 861–874, 1999.

[18] D. P. Palomar and M. Mung Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1439–1451, 2006.

[19] D. P. Palomar and M. Chiang, "Alternative distributed algorithms for network utility maximization: framework and applications," *IEEE Transactions on Automatic Control*, vol. 52, no. 12, pp. 2254–2269, Dec 2007.

[20] N. Li, L. Chen, and S. H. Low, "Optimal demand response based on utility maximization in power networks," in *Proceedings of the 2011 IEEE Power and Energy Society General Meeting*, pp. 1–8, Detroit, MI, USA, July 2011.

[21] L. Chen, N. Li, S. H. Low, and J. C. Doyle, "Two market models for demand response in power networks," in *IEEE International Conference on Smart Grid Communications*, pp. 397–402, Gaithersburg, MD, USA, October 2010.

[22] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[23] Q. Zhang, K. Dehghanpour, Z. Wang, F. Qiu, and D. Zhao, "Multi-agent safe policy learning for power management of

networked microgrids," *IEEE Transactions on Smart Grid*, vol. 12, no. 2, pp. 1048–1062, 2021.

[24] R. Madan and S. Lall, "Distributed algorithms for maximum lifetime routing in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 5, no. 8, pp. 2185–2193, 2006.

[25] Q. Zhang and M. Sahraei-Ardakani, "Distributed DCOPF with flexible transmission," *Electric Power Systems Research*, vol. 154, pp. 37–47, 2018.

[26] Q. Zhang and M. Sahraei-Ardakani, "Impacts of communication limits on convergence of distributed DCOPF with flexible transmission," in *Proceedings of the 2017 North American Power Symposium (NAPS)*, pp. 1–6, Morgantown, WV, USA, September 2017.

[27] X. Shi, J. Cao, G. Wen, and M. Perc, "Finite-time consensus of opinion dynamics and its applications to distributed optimization over digraph," *IEEE Transactions on Cybernetics*, vol. 49, no. 10, pp. 3767–3779, 2019.

[28] J. N. Tsitsiklis, *Problems in decentralized decision making and computation*, Ph.D. dissertation, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge, MA, USA, 1984.

[29] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.

[30] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, Belmont, MA, USA, 2nd edition, 1997.

[31] G. B. Dantzig and P. Wolfe, "Decomposition principle for linear programs," *Operations Research*, vol. 8, no. 1, pp. 101–111, 1960.

[32] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[33] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, 1995.

[34] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[35] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[36] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[37] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[38] A. Nedić and A. Ozdaglar, *Cooperative Distributed Multi-Agent Optimization*, Convex Optimization in Signal Processing and Communications, New York, NY, USA, 2009.

[39] J. O. Ramsay, "A family of gradient methods for optimization," *The Computer Journal*, vol. 13, no. 4, pp. 413–417, 1970.

[40] S. Pu, W. Shi, J. Xu, and A. Nedic, "Push-pull gradient methods for distributed optimization in networks," *IEEE Transactions on Automatic Control*, vol. 66, no. 1, pp. 1–16, 2021.

[41] P. H. Calamai and J. J. Moré, "Projected gradient methods for linearly constrained problems," *Mathematical Programming*, vol. 39, no. 1, pp. 93–116, 1987.

[42] A. Nedić and A. Ozdaglar, "On the rate of convergence of distributed subgradient methods for multi-agent optimization," in *Proceedings of the 2007 46th IEEE Conference on Decision and Control*, pp. 4711–4716, New Orleans, LA, USA, January 2007.

[43] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[44] M. Amini and F. Yousefian, "An iterative regularized incremental projected subgradient method for a class of bilevel optimization problems," in *Proceedings of the 2019 American Control Conference (ACC)*, pp. 4069–4074, Philadelphia, PA, USA, May 2019.

[45] D. Liu, S. Li, and Y. Shen, "One-bit compressive sensing with projected subgradient method under sparsity constraints," *IEEE Transactions on Information Theory*, vol. 65, no. 10, pp. 6650–6663, 2019.

[46] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.

[47] X. Kou, F. Li, J. Dong et al., "A scalable and distributed algorithm for managing residential demand response programs using alternating direction method of multipliers (ADMM)," *IEEE Transactions on Smart Grid*, vol. 11, no. 6, pp. 4871–4882, 2020.

[48] L. Yang, J. Luo, Y. Xu, Z. Zhang, and Z. Dong, "A distributed dual consensus ADMM based on partition for DC-DOPF with carbon emission trading," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1858–1872, 2020.

[49] D. Hajinezhad and Q. Shi, "Alternating direction method of multipliers for a class of nonconvex bilinear optimization: convergence analysis and applications," *Journal of Global Optimization*, vol. 70, no. 1, pp. 261–288, 2018.

[50] Q. Zhang, K. Dehghanpour, and Z. Wang, "Distributed CVR in unbalanced distribution systems with PV penetration," *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5308–5319, 2019.

[51] B. T. Polyak, *Introduction to Optimization*, Optimization Software, Inc., Publications Division, Park Avenue, NY, USA, 1987.

[52] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, 1988.

[53] K. S. Narendra and K. Parthasarathy, "Gradient methods for the optimization of dynamical systems containing neural networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 252–262, 1991.

[54] S. Magnússon, C. Enyioha, N. Li, C. Fischione, and V. Tarokh, "Communication complexity of dual decomposition methods for distributed resource allocation optimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 4, pp. 717–732, 2018.

[55] D. P. Bertsekas and J. N. Tsitsiklis, "Gradient convergence in gradient methods with errors," *SIAM Journal on Optimization*, vol. 10, no. 3, pp. 627–642, 1999.

[56] N. Z. Shor, *Minimization Methods for Non-Differentiable Functions*, Springer-Verlag Berlin, Heidelberg, Germany, 1985.

[57] S. Boyd, "Subgradient methods," 2007, http://stanford.edu/class/ee364b/lectures/subgrad_method_notes.pdf.

[58] G. David, *Luenberger and Yinyu Ye, Linear and Nonlinear Programming*, Springer International publishing, Berlin, Germany, 2016.

[59] G. N. Nair, F. Fagnani, S. Zampieri, and R. J. Evans, "Feedback control under data rate constraints: an overview," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 108–137, 2007.

[60] R. W. Brockett and D. Liberzon, "Quantized feedback stabilization of linear systems," *IEEE Transactions on Automatic Control*, vol. 45, no. 7, pp. 1279–1289, 2000.

[61] E. J. Msechu and G. B. Giannakis, "Sensor-centric data reduction for estimation with WSNs via censoring and quantization," *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 400–414, 2012.

[62] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "On distributed averaging algorithms and quantization effects," *IEEE Transactions on Automatic Control*, vol. 54, no. 11, pp. 2506–2517, 2009.

[63] M. Fukushima, "Application of the alternating direction method of multipliers to separable convex programming problems," *Computational Optimization and Applications*, vol. 1, no. 1, pp. 93–111, 1992.

[64] M. R. Hestenes, "Multiplier and gradient methods," *Journal of Optimization Theory and Applications*, vol. 4, no. 5, pp. 303–320, 1969.

[65] T. Erseghe, "Distributed optimal power flow using ADMM," *IEEE Transactions on Power Systems*, vol. 29, no. 5, pp. 2370–2380, 2014.

[66] P. Olivella-Rosell, F. Rullan, P. Lloret-Gallego et al., "Centralised and distributed optimization for aggregated flexibility services provision," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3257–3269, 2020.

[67] M. Wenlong, H. Zhang, and M. Fu, "Distributed convex optimization based on ADMM and belief propagation methods," *Asian Journal of Control*, vol. 23, no. 2, pp. 1040–1051, 2020.

[68] L. Majzoobi, F. Lahouti, and V. Shah-Mansouri, "Analysis of distributed ADMM algorithm for consensus optimization in presence of node error," *IEEE Transactions on Signal Processing*, vol. 67, no. 7, pp. 1774–1784, 2019.

[69] V. Dvorkin, J. Kazempour, L. Baringo, and P. Pinson, ""A consensus-ADMM approach for strategic generation investment in electricity markets," in *Proceedings of the 2018 IEEE Conference on Decision and Control (CDC)*, pp. 780–785, Miami Beach, FL, USA, December 2018.

[70] J. Wangni, J. Wang, J. Liu, and T. Zhang, "Gradient sparsification for communication-efficient distributed optimization," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS 2018)*, pp. 1306–1316, Montréal, Canada, December 2018.

[71] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 1707–1718, Long Beach, CA, USA, December 2017.

[72] L. Grippo, "A class of unconstrained minimization methods for neural network training," *Optimization Methods and Software*, vol. 4, no. 2, pp. 135–150, 1994.

[73] L. Zhi-Quan and T. Paul, "Analysis of an approximate gradient projection method with applications to the backpropagation algorithm," *Optimization Methods and Software*, vol. 4, no. 2, pp. 85–101, 1994.

[74] O. L. Mangasarian and M. V. Solodov, "Serial and parallel backpropagation convergence via nonmonotone perturbed minimization," *Optimization Methods and Software*, vol. 4, no. 2, pp. 103–116, 1994.

[75] B. T. Polyak and Y. Z. Tsypkin, "Pseudogradient adaptation and training algorithms," *Automation and Remote Control*, vol. 34, no. 3, pp. 377–397, 1973.

[76] B. Polyak, "Nonlinear programming methods in the presence of noise," *Mathematical Programming*, vol. 14, no. 1, pp. 87–97, 1978.

[77] A. Nedić and D. P. Bertsekas, "The effect of deterministic noise in subgradient methods," *Mathematical Programming*, vol. 125, no. 1, pp. 75–99, 2010.

[78] Y. Ermoliev, "Stochastic quasigradient methods and their application to system optimization," *Stochastics*, vol. 9, no. 1-2, pp. 1–36, 1983.