Distributed constrained optimisation over cloud-based multi-agent networks

Wei Xu

Department of Automation,
University of Science and Technology of China,
Hefei, Anhui, 230026, China
Email: shuaiwei@mail.ustc.edu.cn

Qing Ling*

School of Data and Computer Science,
Sun Yat-Sen University,
Guangzhou, Guangdong, 510006, China
Email: lingqing556@mail.sysu.edu.cn
*Corresponding author

Yongcheng Li and Manxi Wang

State Key Laboratory of Complex Electromagnetic
Environment Effects on Electronics and Information System,
Luoyang, Henan, 471003, China
Email: lyncan@163.com
Email: manxiwang@126.com

Abstract: We consider a distributed constrained optimisation problem where a group of distributed agents are interconnected via a cloud center, and collaboratively minimise a network-wide objective function subject to local and global constraints. This paper devotes to developing efficient distributed algorithms that fully utilise the computation abilities of the cloud center and the agents, as well as avoid extensive communications between the cloud center and the agents. We address these issues by introducing two divide-and-conquer techniques, the alternating direction method of multipliers (ADMM) and a primal-dual first-order (PDFO) method, which assign the local objective functions and constraints to the agents while the global ones to the cloud center. Both algorithms are proved to be convergent to the primal-dual optimal solution. Numerical experiments demonstrate the effectiveness of the proposed distributed constrained optimisation algorithms.

Keywords: cloud computing; distributed optimisation; ADMM; alternating direction method of multipliers; PDFO; primal-dual first-order method.

Reference to this paper should be made as follows: Xu, W., Ling, Q., Li, Y. and Wang, M. (2018) 'Distributed constrained optimisation over cloud-based multi-agent networks', *Int. J. Sensor Networks*, Vol. 28, No. 1, pp.43–56.

Biographical notes: Wei Xu received his BE in Automation from University of Science and Technology of China in 2016. He is currently a PhD student in the Department of Automation, University of Science and Technology of China. His research interest is decentralised optimisation over networks.

Qing Ling received his BE in Automation and the PhD degree in Control Theory and Control Engineering from the University of Science and Technology of China in 2001 and 2006, respectively. He was a Postdoctoral Research Fellow in the Department of Electrical and Computer Engineering, Michigan Technological University from 2006 to 2009, and an Associate Professor in the Department of Automation, University of Science and Technology of China from 2009 to 2017. He is currently a Full Professor in the School of Data and Computer Science, Sun Yat-Sen University. His research interests are decentralised network optimisation and its applications. He is an Associate Editor of IEEE Transactions on Network and Service Management and IEEE Signal Processing Letters.

Yongcheng Li received his BE in Automation from Northeastern University in 2001 and the MS in Control Theory and Control Engineering from the University of Science and Technology of China in 2012. He is currently a Senior Engineer in the State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics and Information System. His research focuses on cognitive radio networks.

Manxi Wang received his BE, MS and PhD degrees in Telecommunications Engineering from Nanjing Science and Technology University in 2001, 2004, and 2009, respectively. He is currently an Associate Research Fellow in the State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics and Information System. His research focuses on cognitive wireless communications and physical-layer security.

This paper is a revised and expanded version of a paper entitled 'Distributed constrained optimization over cloud-based multi-agent networks' presented at the *11th International Conference on Wireless Algorithms, Systems, and Applications (WASA' 2016)*, Bozeman, USA, 8–10 August, 2016.

1 Introduction

This paper considers a distributed constrained optimisation problem defined over a cloud-based multi-agent network. A group of n distributed agents (also called as nodes) are interconnected via a cloud center, and collaboratively minimise a network-wide objective function subject to local and global constraints. To be specific, every agent i has a local objective function $f_i(x_i)$ and a local constraint $x_i \in \mathcal{X}_i$, where $x_i \in \mathcal{R}^{p_i}$ is a local optimisation variable. The network-wide objective function is defined as $\sum_{i=1}^n f_i(x_i) + h(x)$, with $h(x)$ being the global objective function depending on the stacked optimisation variable $x \triangleq [x_1; \cdots; x_n] \in \mathcal{X} \subseteq \mathcal{R}^p$, $\mathcal{X}$ is the Cartesian product of $\mathcal{X}_i$, $i = 1, \ldots, n$ and $p = \sum_{i=1}^n p_i$. The network is also subject to m global constraints on the stacked optimisation variable x, denoted as $g_j(x) \leq 0,\ j = 1, \ldots, m$. Therefore, the constrained optimisation problem is in the form of

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} f_i(x_i) + h(x), \\ s.t. \quad & x_i \in \mathcal{X}_i,\ i = 1, \ldots, n, \\ & g_j(x) \leq 0,\ j = 1, \ldots, m. \end{aligned} \tag{1}$$

The form of equation (1) appears in a large number of network applications. For example, every agent can be a robot and the team of cooperative robots are coordinated via a cloud center to monitor an interested area. In this setting, the local optimisation variables x_i stand for the positions of the robots. The local objective functions and constraints determine the expected positions of the individual robots, while the global objective functions and constraints determine the relative orientations of all the robots (Hale and Egerstedt, 2014; Hale et al., 2015). Network resource allocation of a cloud network can also be formulated as such a constrained optimisation problem (Beck et al., 2014; Chen et al., 2017). More examples include networks of smart portable devices (Armbrust et al., 2010; Bonomi et al., 2012), smart grids (Giannakis et al., 2013; Li and Scaglione, 2013), large-scale machine learning systems (Boyd et al., 2011; Mokhtari et al., 2016), to name a few.

One naive approach to solving equation (1) is letting the cloud center collect all the local objective functions and constraints, and find an optimal solution in a centralised manner. This centralised approach is costly in communication, inflexible to changes of the network topology and the optimisation task, and insecure due to the danger of leaking local information. A favourable choice is designing distributed algorithms that offload some computation tasks to the agents. That is, every agent processes its own local objective function and constraint while the cloud center processes the global ones; the agents and the cloud center coordinate to guarantee the optimality of solutions (Tsitsiklis et al., 1986; Bertsekas and Tsitsiklis, 1997).

Classical divide-and-conquer methods to handle this problem operate in the primal-dual domain. The dual decomposition method considers a Lagrangian function that dualises the inequality constraints and optimises along dual (sub)gradient ascent directions. Recent works (Hale and Egerstedt, 2014; Hale et al., 2015) propose dual decomposition algorithms to solve equation (1). However, at every iteration they require the cloud center to collect the primal variables from all the agents, update the dual variables, and send all the primal and dual variables to all the agents. Therefore, they incur heavy communication load. In addition, the main disadvantage of the dual decomposition method is its slow convergence (Bertsekas, 1999; Chiang et al., 2007). To address this issue, (Hale et al., 2015) appends Tikhnov regularisation terms to the dual function so as to improve the convergence speed. Nevertheless, this approach leads to an inexact solution that resides in a neighbourhood of the optimum of equation (1).

Another remarkable primal-dual algorithm is the alternating direction method of multipliers (ADMM) (Gabay and Mercier, 1976; Eckstein and Bertsekas, 1992). Through introducing local copies of the optimisation variable in both the cloud center and the agents, imposing and dualising consensus constraints on them, and optimising the augmented Lagrangian in an alternating direction manner, ADMM demonstrates fast convergence both in practice (Boyd et al., 2011; Giannakis et al., 2016) and in theory (He and Yuan, 2012; Deng and Yin, 2016; Hong and Luo, 2017; Shi et al., 2014). Since the alternating optimisations of the primal variables often involve nontrivial subproblems, solving their linear or quadratic approximations is able to reduce the computation costs (Ng and Yuan, 2011; Ling et al., 2015; Chang et al., 2015; Mokhtari et al., 2016).

This paper devotes to developing efficient distributed algorithms that fully utilise the computation abilities of the cloud center and the agents, as well as avoid extensive communications between the cloud center and the agents. We address these issues by introducing two divide-and-conquer techniques, both of which assign the local objective functions and constraints to the agents while the global ones to the cloud center. The first algorithm is based on ADMM and the second one is a primal-dual first-order (PDFO) method. Both algorithms naturally yield two layers, the agent layer and the cloud center layer, which exchange intermediate variables so as to collaboratively obtain a network-wide optimal solution. However, the ADMM-based algorithm requires the cloud center and the agents to solve optimisation subproblems at every iteration, and hence brings high computation cost. The PDFO method significantly reduces the iteration-wise computation cost by replacing the optimisation subproblems with simple first-order gradient descent and projection operations.

The rest of the paper is organised as follows. Section 2 develops the distributed constrained optimisation algorithm based on ADMM. The PDFO method with lower iteration-wise computation cost is proposed in Section 3. Section 4 provides numerical experiments to demonstrate the effectiveness of the two proposed algorithms. Section 5 concludes the paper and gives future research directions.

2 ADMM-based distributed constrained optimisation

Section 2.1 solves equation (1) based on ADMM. The resultant algorithm is outlined in Section 2.2 and implementation issues are discussed in Section 2.3. Section 2.4 compares the proposed algorithm with existing works.

2.1 Algorithm development

ADMM solves an optimisation problem in the form

$$\begin{aligned} \min \quad & c(x) + d(y), \\ s.t. \quad & Ax + By = 0, \\ & x \in \mathcal{X},\ y \in \mathcal{Y}. \end{aligned} \tag{2}$$

Here $c(x)$ and $d(y)$ are functions of optimisation variables x and y, respectively; A and B are two matrices; $\mathcal{X}$ and $\mathcal{Y}$ are the extra constraint sets on x and y, respectively. The two variables are entangled in the linear constraint $Ax + By = 0$, which makes solving equation (2) a difficult task.

ADMM operates on the augmented Lagrangian function of equation (2), minimises the primal variables x and y in an alternating manner, and updates the dual variable corresponding to the linear constraint $Ax + By = 0$ with dual (sub)gradient ascent. The idea of ADMM can be traced back to the 1970s (Gabay and Mercier, 1976). When the objective functions $c(x)$ and $d(y)$ as well as the constraint sets $\mathcal{X}$ and $\mathcal{Y}$ are convex, convergence of the iterate to the optimal primal-dual pair is proved in Eckstein and Bertsekas (1992), while its sublinear convergence rate is established in He and Yuan (2012). When the objective functions are strongly convex, ADMM has linear convergence rate (Deng and Yin, 2016; Hong and Luo, 2017). The fast convergence speed and the superior numerical stability have made ADMM a popular choice for a large number of applications in recent years, particularly for distributed optimisation; readers are referred to the survey papers (Boyd et al., 2011; Giannakis et al., 2016).

Observe that in equation (1) there is only one group optimisation variables $x = [x_1; \cdots; x_n]$. To apply ADMM to equation (1), we introduce another group of auxiliary variables $y \triangleq [y_1; \cdots; y_n] \in \mathcal{R}^p$ where $y_i \in \mathcal{R}^{p_i}$ for all i. Replacing x in $h(x)$ and all $g_j(x)$ by y and appending a consensus constraint $x = y$, we have

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} f_i(x_i) + h(y), \\ s.t. \quad & x - y = 0, \\ & x_i \in \mathcal{X}_i,\ i = 1, \ldots, n, \\ & g_j(y) \le 0,\ j = 1, \ldots, m, \end{aligned} \tag{3}$$

which is equivalent to equation (1). Now equation (3) is in the form of equation (2) and ADMM is hence applicable.

Define the augmented Lagrangian function that dualises the linear constraint $x - y = 0$ as

$$\begin{aligned} \mathcal{L}_\rho(x, y, \mu) = & \sum_{i=1}^{n} f_i(x_i) + h(y) + \langle \mu, x - y \rangle \\ & + \frac{\rho}{2}\|x - y\|^2, \\ s.t. \quad & x_i \in \mathcal{X}_i,\ i = 1, \ldots, n, \\ & g_j(y) \le 0,\ j = 1, \ldots, m, \end{aligned} \tag{4}$$

where $\mu \in \mathcal{R}^p$ is the Lagrange multiplier and ρ is a positive constant. At every iteration, ADMM first fixes the primal variable y and the dual variable μ, and minimises the augmented Lagrangian function over the primal variable x. Then it fixes x and μ to minimise the augmented Lagrangian function over y. Finally, the dual variable μ is updated from x and y through a dual (sub)gradient step. Therefore, at time $k + 1$, the update of the primal variable x is

$$\begin{aligned} x^{k+1} = \arg\min \ & \sum_{i=1}^{n} f_i(x_i) + \frac{\rho}{2}\|x - y^k + \frac{\mu^k}{\rho}\|^2, \\ & s.t.\ x_i \in \mathcal{X}_i,\ i = 1, \ldots, n. \end{aligned} \tag{5}$$

Observe that $y \triangleq [y_1; \cdots; y_n] \in \mathcal{R}^p$ and $\mu \triangleq [\mu_1; \cdots; \mu_n] \in \mathcal{R}^p$ where $y_i \in \mathcal{R}^{p_i}$ and $\mu_i \in \mathcal{R}^{p_i}$. Therefore, equation (5) is separable with respect to all x_i. To be specific, the update of x_i is

$$x_i^{k+1} = \arg\min_{x_i \in \mathcal{X}_i} \ f_i(x_i) + \frac{\rho}{2}\|x_i - y_i^k + \frac{\mu_i^k}{\rho}\|^2. \tag{6}$$

The update of the auxiliary variable y is

$$\begin{aligned} y^{k+1} = \arg\min \ & h(y) + \frac{\rho}{2}\|y - x^{k+1} - \frac{\mu^k}{\rho}\|^2, \\ & s.t. \ g_j(y) \leq 0, \ j = 1, \dots, m. \end{aligned} \tag{7}$$

The update of μ is

$$\mu^{k+1} = \mu^k + \rho(x^{k+1} - y^{k+1}). \tag{8}$$

Notice that in this dual (sub)gradient step, the step size is the same as the positive constant ρ in the augmented Lagrangian function (4).

In the ADMM updates, equation (8) only involves simple arithmetic operations. Given that $\mathcal{X}_i$ is simple such that projecting onto it is computationally efficient and f_i is convex, equation (6) can be solved by the projected (sub)gradient method (Boyd and Vandenberghe, 2004). However, equation (7) can be nontrivial if the objective function $h(y)$ and the inequality constraint functions $g_j(y)$ are neither linear nor quadratic. Below we propose to solve equation (7) with the dual decomposition method (Bertsekas, 1999; Chiang et al., 2007).

By introducing nonnegative Lagrange multipliers $\nu_j \in \mathcal{R}_+$ to the inequality constraints $g_j(y) \leq 0$ in equation (7), we have the Lagrangian function

$$\mathcal{L}(y, \nu) = h(y) + \frac{\rho}{2}\|y - x^{k+1} - \frac{\mu^k}{\rho}\|^2 + \sum_{j=1}^{m} \nu_j g_j(y). \tag{9}$$

Denote $\nu \triangleq [\nu_1; \cdots; \nu_m] \in \mathcal{R}_+^m$ as a vector that stacks all the Lagrange multipliers. Notice that the Lagrangian function is not augmented, since appending the squares of nonlinear inequality constraint functions makes the function nonconvex in the primal variable y. At slot $t+1$ in the inner loop of time $k+1$, the dual decomposition method calculates y with (sub)gradient descent, followed by the update of ν from projected (sub)gradient ascent. Denoting $y^{k+1}(t+1)$ and $\nu^{k+1}(t+1)$ as the calculated values of y and μ at slot $t+1$ in the inner loop of time $k+1$, respectively, we have

$$\begin{aligned} & y^{k+1}(t+1) \\ =& y^{k+1}(t) - c\nabla_{y^{k+1}(t)}\mathcal{L}(y^{k+1}(t), \nu^{k+1}(t)), \end{aligned} \tag{10}$$

where

$$\begin{aligned} \nabla_{y^{k+1}(t)}\mathcal{L}(y^{k+1}(t), \nu^{k+1}(t)) &= \nabla h(y^{k+1}(t)) + \rho y^{k+1}(t) \\ - \rho x^{k+1} - \mu^k + \sum_{j=1}^{m} \nu_j^{k+1}(t) &\nabla g_j(y^{k+1}(t)). \end{aligned}$$

In equation (10), c is a positive dual decomposition step size. With a slight abuse of notations, ∇ is the gradient if the function is differentiable, or a subgradient if the function is non-differentiable. For the update of ν, we have

$$\begin{aligned} & \nu^{k+1}(t+1) \\ =& \Big[\nu^{k+1}(t) + c\nabla_{\nu^{k+1}(t)}\mathcal{L}(y^{k+1}(t+1), \nu^{k+1}(t))\Big]_+, \end{aligned} \tag{11}$$

where $[\cdot]_+$ denotes projection onto the nonnegative orthant. Due to the separable structure of $\mathcal{L}(y, \nu)$ with respect to all ν_j, for every constraint j the update of equation (11) yields

$$\nu_j^{k+1}(t+1) = \left[\nu_j^{k+1}(t) + c g_j(y^{k+1}(t+1))\right]_+. \tag{12}$$

Remark 1: Given that the objective functions and the constraints in equation (3) – and equivalently in (1) – are convex, and the subproblems (6) and (7) are accurately solved, the sequence generated by the ADMM updates (6), (7) and (8) guarantees to converge to an optimal solution of (3) for any positive parameter ρ (Eckstein and Bertsekas, 1992; He and Yuan, 2012). Suppose that equation (6) has been accurately solved by the projected (sub)gradient method. Therefore, convergence of the ADMM updates is determined by the accuracy of solving equation (7). It is known that the dual decomposition updates (10) and (11) also converge to an optimal solution of the convex program (7) when the step size c is sufficiently small (Bertsekas, 1999). The computational stability of the dual decomposition method is favourable due to the existence of the quadratic term in equation (7), which makes the objective function strongly convex.

2.2 Algorithm outline

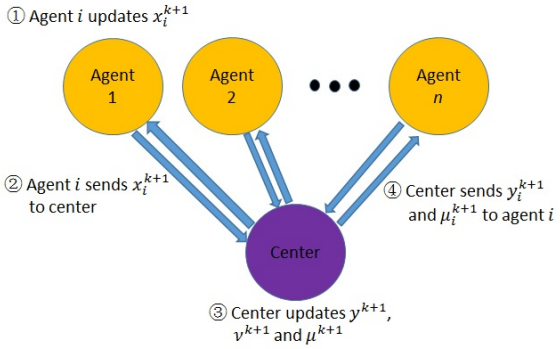
The proposed ADMM-based distributed constrained optimisation algorithm is outlined in Algorithm 1. At time $k = 0$, the cloud center and the agents initialise the variables as $y^0 \in \mathcal{X}$, $\mu^0 = 0$, $\nu^0 = 0$ and $x_i^0 \in \mathcal{X}_i$. At time $k+1$, every agent i updates x_i^{k+1} from equation (6), using the values of y_i^k and μ_i^k that are known after the communication step in Line 12. Then the cloud center collects all x_i^{k+1} from the agents and solves y^{k+1} from equation (7) by an inner loop with T slots. The initial values of the inner loop are $y^{k+1}(0) = y^k$ and $\nu^{k+1}(0) = \nu^k$. The primal update of $y^{k+1}(t+1)$ from equation (10) requires local variables $y^{k+1}(t)$ and μ^k as well as variables x_i^{k+1} available from the communication step in Line 3. The dual update of $\nu^{k+1}(t+1)$ in equation (11) requires only local variables $y^{k+1}(t+1)$ and $\nu^{k+1}(t)$. When the inner loop terminates after T slots, the outer-loop primal variable is set as $y^{k+1} = y^{k+1}(T)$ and the inner-loop dual variable is set as $\nu^{k+1} = \nu^{k+1}(T)$. Both y^{k+1} and x^{k+1} are used in calculating the dual variable μ^{k+1} from equation (8). Finally, y_i^{k+1} and μ_i^{k+1} are disseminated to every agent i by the cloud center. Figure 1 shows a schematic diagram on the computation and communication of cloud center and agents.

2.3 Implementation issues

As we have discussed in Section 2.1, the ADMM-based distributed constrained optimisation algorithm converges to the optimal solution of equation (3) when the subproblems (6)

and (7) are exactly solved. If computing the projection onto the set $\mathcal{X}_i$ is affordable in agent i, equation (6) can be efficiently solved by the projected (sub)gradient method. Indeed, it is possible to run the projected (sub)gradient method for only several steps to obtain an approximate solution, while still guaranteeing convergence, as we have observed in the inexact ADMM approaches (Ng and Yuan, 2011; Ling et al., 2015; Chang et al., 2015; Mokhtari et al., 2016). Further, if the local constraint $x_i \in \mathcal{X}_i$ is absent and the local objective function $f_i(x_i)$ has a special form (for example, the ℓ_1 and ℓ_2 norms), the resultant problem has an explicit solution.

Figure 1 Computation and communication of cloud center and agents (see online version for colours)



Therefore, the main computational bottleneck of the algorithm resides in the inner loops of solving equation (7). Theoretically, the dual decomposition subroutine converges to the optimal solution of equation (7) only when the number of inner iterations T goes to infinity. Though the cloud center is often supposed to have strong computation power, spending a large number of iterations in the inner loops is still inefficient. Therefore, in implementing the algorithm, we set T to be a fixed small number. Numerical experiments in Section 4 show that the solutions of the inner loops will be more and more accurate as the outer loops evolve. With particular note, since the dual decomposition subroutine works on the primal-dual domain, theoretically analysing the difference between the inexact solution and the optimum to equation (7) is challenging. For this reason, the existing convergence guarantee for inexact ADMM no longer holds. We leave the analysis of this inexact version of the proposed algorithm to future research.

2.4 Comparisons with existing works

The main feature of the proposed ADMM-based distributed constrained optimisation algorithm is to utilise the divide-and-conquer technique for assigning the computation to two layers, the cloud center layer and the distributed agent layer. The two layers conduct their computing tasks respectively, while collaborate to obtain the global optimal solution through information exchange. To be specific, the cloud center sends the intermediate primal variable $y^{k+1} \in \mathcal{R}^p$ and the dual variable $\mu^{k+1} \in \mathcal{R}^p$ to the agents, while the agents send the intermediate primal variables $x_i^{k+1} \in \mathcal{R}^{p_i}$ to the cloud center. Notice that y^{k+1} is an estimate of $x = [x_1; \cdots; x_n]$ from the cloud center's perspective. The two primal variables y^{k+1} and $x^{k+1} = [x_1^{k+1}; \cdots; x_n^{k+1}]$ shall converge to the same optimal argument as k goes to infinity, and the role of the dual variable μ^{k+1} is to punish the gap between the two vectors. This setting results in affordable communication costs between the two layers.

A naive approach to solving the constrained optimisation problem (1) is to let the cloud center collect all the local objective functions and local constraints from the agents. Apparently, this approach is costly in communication, inflexible to the changes of the network topology and the optimisation task, and insecure due to the danger of leaking local information. Contrarily, the proposed algorithm has the advantages of lightweight communication, flexibility to uncertainties, and privacy preservation.

Two notable approaches to solving the constrained optimisation problem (1) are the dual decomposition algorithm proposed in Hale and Egerstedt (2014) and the Tikhonov regularised dual decomposition algorithm proposed in Hale et al. (2015). The two algorithms both work in the primal-dual domain of (1) – instead, our algorithm works in the primal-dual domain of the equivalent form (3). The dual decomposition algorithm in Hale and Egerstedt (2014) suffers from computational instability when the objective function is not strongly convex or non-differentiable. This issue is addressed in Hale et al. (2015) by introducing Tikhonov regularisation to both the primal and dual variables, at the cost of yielding a non-optimal solution. Note that at every iteration these two algorithms require the cloud center to collect the primal variables from all the agents, update the dual variables, and send both the primal and dual variables to all the agents. Therefore, at every iteration the cloud center collects a p-dimensional primal variable, while sends to every agent an m-dimensional dual variable and a p-dimensional primal variable. In comparison, at every iteration in Algorithm 1, the cloud center collects p-dimensional primal variables $x = [x_1; \cdots; x_n]$, while sends to agent i a p_i-dimensional dual variable μ_i and a p_i-dimensional primal variable y_i. In summary, at every iteration of Hale and Egerstedt (2014) and Hale et al. (2015), the communication load of the cloud center is p in collecting information, and $p + m$ in sending information in a broadcast mode. In a unicast mode, the communication load of sending increases to $n(m + p)$, which is proportional to the number of agents n. In our algorithm, the communication load of the cloud center is $2p$ in sending and p in collecting, both irrelevant with the network size. Similarly, for agent i the communication loads of sending are both p_i in the two algorithms, but those of collecting are $p + m$ in Hale and Egerstedt (2014); Hale et al. (2015), while $2p_i$ in Algorithm 1. Therefore, the proposed algorithm is more communication-efficient than those in Hale and Egerstedt (2014); Hale et al. (2015).

3 PDFO method for distributed constrained optimisation

To reduce the iteration-wise computational cost of ADMM, Section 3.1 proposes a PDFO method for solving equation (1) such that the cloud center and the agents do not need to

run inner loops to solve optimisation subproblems. Instead, at every iteration the cloud center only uses one projected (sub)gradient step to update its primal variable; the same every agent does. The resultant algorithm is outlined in Section 3.2. Its convergence is analysed in Section 3.3.

Algorithm 1 ADMM-Based Algorithm

Require: Cloud center initializes $y^0 \in \mathcal{X}$, $\mu^0 = 0$, $\nu^0 = 0$.
Require: Every agent i initializes $x_i^0 = \mathcal{X}_i$.
for times $k = 1, 2, \dots$ **do**
  Every agent i updates x_i^{k+1} from (6).
  Cloud center collects all x_i^{k+1} from agents.
  Cloud center solves (7) by the following inner loop.
    Initializes $y^{k+1}(0) = y^k$ and $\nu^{k+1}(0) = \nu^k$.
    for $t = 0, 1, \cdots, T-1$ **do**
      Updates primal variable $y^{k+1}(t+1)$ from (10);
      Updates dual variable $\nu^{k+1}(t+1)$ from (11).
    end for
    Lets $y^{k+1} = y^{k+1}(T)$ and $\nu^{k+1} = \nu^{k+1}(T)$.
  Cloud center calculates dual variable μ^{k+1} from (8).
  Cloud center sends y_i^{k+1} and μ_i^{k+1} to every agent i.
end for

3.1 Algorithm development

The proposed PDFO algorithm considers another equivalent form of equation (1), which is

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} f_i(x_i) + h(y), \\ s.t. \quad & x - y = 0, \\ & x_i \in \mathcal{X}_i,\ i = 1, \dots, n, \\ & g_j(y) \le 0,\ j = 1, \dots, m,\ y \in \mathcal{X}. \end{aligned} \tag{13}$$

Again, $y \triangleq [y_1; \cdots; y_n] \in \mathcal{R}^p$ is an auxiliary variable, where $y_i \in \mathcal{R}^{p_i}$. Observe that equation (13) is slightly different to equation (3) by adding the constraint $y \in \mathcal{X}$, where $\mathcal{X}$ is defined as the Cartesian product of $\mathcal{X}_i$, $i = 1, \dots, n$. This slight modification is important to the development of the PDFO algorithm. Dualising the equality constraint $x - y = 0$ and the inequality constraints $g_j(y) \le 0$ yields a partially augmented Lagrangian function

$$\begin{aligned} \widetilde{\mathcal{L}}_\rho(x, y, \mu, \nu) = & \sum_{i=1}^{n} f_i(x_i) + h(y) + \sum_{j=1}^{m} \nu_j g_j(y) \\ & + \langle \mu, x - y \rangle + \frac{\rho}{2}\|x - y\|^2, \\ & s.t. \quad x_i \in \mathcal{X}_i,\ i = 1, \dots, n,\ y \in \mathcal{X}, \end{aligned} \tag{14}$$

where $\mu \in \mathcal{R}^p$ is the Lagrange multiplier attached to $x - y = 0$, $\nu_j \in \mathcal{R}_+$ are the Lagrange multipliers attached to $g_j(y) \le 0$, $j = 1, \dots, n$, and ρ is a positive constant. Notice that equation (14) is different to equation (4) since $g_j(y) \le 0$ are also dualised. To avoid introducing nonconvex terms, these inequality constraints are not quadratically augmented to equation (14).

At every iteration, the proposed PDFO algorithm first fixes y, μ, and ν to inexactly minimise $\widetilde{\mathcal{L}}_\rho$ over x, then fixes x, μ, and ν to inexactly minimise $\widetilde{\mathcal{L}}_\rho$ over y, and finally updates μ and ν. At time $k+1$, the update of x_i is

$$\begin{aligned} x_i^{k+1} &= \mathcal{P}_{\mathcal{X}_i}\big[x_i^k - a\nabla_{x_i^k}\widetilde{\mathcal{L}}_\rho(x^k, y^k, \mu^k, \nu^k)\big] \\ &= \mathcal{P}_{\mathcal{X}_i}\big[x_i^k - a(\nabla f_i(x_i^k) + \mu_i^k + \rho x_i^k - \rho y_i^k)\big]. \end{aligned} \tag{15}$$

Here a is a positive step size and $\mathcal{P}_{\mathcal{X}_i}$ denotes projection onto the set $\mathcal{X}_i$. The update of the auxiliary variable y is

$$\begin{aligned} y^{k+1} &= \mathcal{P}_{\mathcal{X}}\big[y^k - b\nabla_{y^k}\widetilde{\mathcal{L}}_\rho(x^{k+1}, y^k, \mu^k, \nu^k)\big] \\ &= \mathcal{P}_{\mathcal{X}}\Big[y^k - b\Big\{\nabla h(y^k) - \mu^k - \rho x^{k+1} + \rho y^k \\ &\qquad + \sum_{j=1}^{m} \nu_j^k \nabla g_j(y^k)\Big\}\Big]. \end{aligned} \tag{16}$$

Again, b is a positive step size and $\mathcal{P}_{\mathcal{X}}$ denotes projection onto the set $\mathcal{X}$. Then the update of μ is

$$\begin{aligned} \mu^{k+1} &= \mu^k + \rho\nabla_{\mu^k}\widetilde{\mathcal{L}}_\rho(x^{k+1}, y^{k+1}, \mu^k, \nu^k) \\ &= \mu^k + \rho(x^{k+1} - y^{k+1}), \end{aligned} \tag{17}$$

where the dual ascent step size ρ is the same as the positive constant used in the augmented Lagrangian $\widetilde{\mathcal{L}}_\rho$. Denote $\nu \triangleq [\nu_1; \cdots; \nu_m] \in \mathcal{R}_+^m$. The update of ν is

$$\nu^{k+1} = \mathcal{P}_{\mathcal{D}}\big[\nu^k + b\nabla_{\nu^k}\widetilde{\mathcal{L}}_\rho(x^{k+1}, y^{k+1}, \mu^{k+1}, \nu^k)\big], \tag{18}$$

and thus for every ν_j we have

$$\nu_j^{k+1} = \mathcal{P}_{\mathcal{D}}\big[\nu_j^k + b g_j(y^{k+1})\big]. \tag{19}$$

Here the dual ascent step size b is the same as the one we used in the update of y, and $\mathcal{P}_{\mathcal{D}}$ denotes projection onto the set $\mathcal{D} \triangleq [0, \nu_{\max}]$. The value of the positive constant $\nu_{\max}$ will be given in equation (28) of Theorem 1.

Remark 2: Comparing the PDFO method with the ADMM-based algorithm, we can observe that their major differences are three-fold. First, the update of x_i in equation (15) is a simple projected (sub)gradient step; while in the ADMM-based algorithm, the update of x_i in equation (6) solves an optimisation problem, which requires running multiple projected (sub)gradient steps. Second, at every iteration, the PDFO method updates y and ν_j only once, but the ADMM-based algorithm uses inner loops and updates them multiple times. Third, the update of y in equation (10) is a (sub)gradient descent step and that in equation (16) projects onto $\mathcal{X}$ after (sub)gradient descent. Meanwhile, the update of ν_j in equation (12) projects onto the nonnegative orthant, but that in equation (19) has an additional upper bound. The first two differences enable the PDFO method to achieve much better iteration-wise computational efficiency than the ADMM-based algorithm. The third difference compensates the lack of inner loops to update y and ν_j, and guarantees the computational stability of the PDFO method. We also

emphasise that the PDFO method has the same iteration-wise communication cost as that of the ADMM-based algorithm, and is hence advantageous over the existing works (Hale and Egerstedt, 2014; Hale et al., 2015) as we have discussed in Section 2.4.

3.2 Algorithm Outline

The proposed PDFO method that solves the distributed constrained optimisation problem (1) is outlined in Algorithm 2. At time $k = 0$, the cloud center and the agents initialise the variables as $y^0 \in \mathcal{X}$, $\mu^0 = 0$, $\nu^0 = 0$ and $x_i^0 \in \mathcal{X}_i$. At time $k+1$, every agent i updates x_i^{k+1} from equation (15), using the values of y_i^k and μ_i^k that are known after the communication step in Line 7. Then the cloud center collects all x_i^{k+1} from the agents, while updates y^{k+1} from equation (16), μ^{k+1} from equation (17), and ν^{k+1} from equation (18). The updates of y^{k+1} and μ^{k+1} require variables x_i^{k+1} available from the communication step in Line 3. The update of ν^{k+1} in equation (11) requires only local variables. Finally, y_i^{k+1} and μ_i^{k+1} are disseminated to every agent i by the cloud center. Notice that PDFO and ADMM, though totally different in algorithmic details, have the same flow of computation and communication, as we have shown in Figure 1.

Algorithm 2 PDFO Method

Require: Cloud center initializes $y^0 \in \mathcal{X}$, $\mu^0 = 0$, $\nu^0 = 0$.
Require: Every agent i initializes $x_i^0 = \mathcal{X}_i$.
1: **for** times $k = 1, 2, \dots$ **do**
2: Every agent i updates x_i^{k+1} from (15).
3: Cloud center collects all x_i^{k+1} from agents.
4: Cloud center updates primal variable y^{k+1} from (16);
5: Cloud center calculates dual variable μ^{k+1} from (17).
6: Cloud center calculates dual variable ν^{k+1} from (18).
7: Cloud center sends y_i^{k+1} and μ_i^{k+1} to every agent i.
8: **end for**

3.3 Convergence analysis

To prove convergence of the PDFO method, we make the following assumptions on the distributed constrained optimisation problem (1).

Assumption 1: *There exists a slater point (Bertsekas et al., 2003)* $(\bar{x}, \bar{y})$ *of equation* (13) *such that* $\bar{x} \in \mathcal{X}$, $\bar{y} \in \mathcal{X}$, $\bar{x} = \bar{y}$, *and* $g_j(\bar{y}) < 0$, $j = 1, \dots, m$.

Assumption 2: *The cost function* $f(x) \triangleq \sum_{i=1}^n f_i(x_i)$ *has Lipschitz continuous gradient with constant* L_f *and is convex. That is, for any two arbitrary points* x *and* $\tilde{x}$ *in the domain of* $f(x)$, *we have*

$$f(x) \le f(\tilde{x}) + \langle \nabla f(\tilde{x}), x - \tilde{x} \rangle + \frac{L_f}{2}\|x - \tilde{x}\|^2, \quad (20)$$

$$f(x) \ge f(\tilde{x}) + \langle \nabla f(\tilde{x}), x - \tilde{x} \rangle. \quad (21)$$

The cost function $h(x)$ *has Lipschitz continuous gradient with constant* L_h *and is strongly convex with constant* M_h. *That is, for any arbitrary points* x *and* $\tilde{x}$ *in the domain of* $h(x)$, *we have*

$$h(x) \le h(\tilde{x}) + \langle \nabla h(\tilde{x}), x - \tilde{x} \rangle + \frac{L_h}{2}\|x - \tilde{x}\|^2, \quad (22)$$

$$h(x) \ge h(\tilde{x}) + \langle \nabla h(\tilde{x}), x - \tilde{x} \rangle + \frac{M_h}{2}\|x - \tilde{x}\|^2. \quad (23)$$

Assumption 3: *The constraint set* $\mathcal{X}$ *is convex and bounded. The constraint functions* $g_j(x)$, $j = 1, \dots, m$ *have Lipschitz continuous gradients with constants* L_{g_j} *and are convex. That is, for any arbitrary points* x *and* $\tilde{x}$ *in the domain of* $g_j(x)$, *we have*

$$g_j(x) \le g_j(\tilde{x}) + \langle \nabla g_j(\tilde{x}), x - \tilde{x} \rangle + \frac{L_{g_j}}{2}\|x - \tilde{x}\|^2, \quad (24)$$

$$g_j(x) \ge g_j(\tilde{x}) + \langle \nabla g_j(\tilde{x}), x - \tilde{x} \rangle. \quad (25)$$

For future usage, collect all the Lipschitz continuity constants L_{g_j} *into a vector* $L_G \triangleq [L_{g_1}; \cdots; L_{g_m}] \in \mathcal{R}^m$. *Also define* $G(x) \triangleq [g_1(x); \cdots; g_m(x)] \in \mathcal{R}^m$. *For any two points* x *and* $\tilde{x}$ *in* $\mathcal{X}$, *there is a positive constant* δ *such that it holds*

$$\|G(x) - G(\tilde{x})\|^2 \le \delta \|x - \tilde{x}\|^2. \quad (26)$$

Assumptions 1 and 2 are common in the convergence analysis of convex optimisation algorithms. Recall the definition of $f(x) = \sum_{i=1}^n f_i(x_i)$, we know that $f(x)$ being convex is equivalent to all $f_i(x_i)$ being convex, and the sufficient condition of $f(x)$ having Lipschitz continuous gradient with constant L_f is that all $\nabla f_i(x_i)$ have Lipschitz continuous gradient with constant L_f. Also notice that the strong convexity of $h(x)$ guarantees that the objective function in equation (1) is also strongly convex, which is important to the computational stability the PDFO method since it only utilises first-order information in the primal-dual domain. Assumption 3 is reasonable because the local decision variables x_i are often bounded, such that $\mathcal{X}$, the Cartesian product of the local constraints $\mathcal{X}_i$, is also bounded. In addition, the inequality (26) naturally satisfies given that all $g_j(x)$ are differentiable and $\mathcal{X}$ is bounded.

An immediate conclusion from Assumptions 1–3 is that any optimal primal-dual solution of equation (13) is bounded as shown by Lemma 1, whose is given in Appendix B.

Lemma 1: *Under Assumptions 1–3, any optimal primal-dual solution* (x^*, y^*, μ^*, ν^*) *of equation* (13) *is bounded.*

The following theorem shows convergence of the PDFO method. Its proof is given in Appendix A.

Theorem 1: *Under Assumptions 1–3 and given that the PDFO parameters* a, b *and* ρ *are chosen such that*

$$\begin{aligned} a &\le \frac{1}{L_f + \rho}, \\ b &\le \min\left\{ \frac{1}{\nu_{\max} \sum_{j=1}^m L_{g_j} + L_h + \rho}, \frac{1}{\sqrt{\delta}}, \frac{M_h}{\delta} \right\}, \end{aligned} \quad (27)$$

and

$$\nu_{\max} > \|\nu^*\| + \left(\|\nu^0 - \nu^*\|^2 + \left(\frac{b}{a} - b\rho \right) \|x^0 - x^*\|^2 + (1 - b^2\delta)\|y^0 - y^*\|^2 + \frac{b}{\rho}\|\mu^0 - \mu^*\|^2 \right)^{1/2}, \quad (28)$$

then the sequence (x^k, y^k, μ^k, ν^k) *generated by the PDFO recursion* (15), (16), (17) *and* (19) *converges to* (x^*, y^*, μ^*, ν^*), *which is an optimal primal-dual solution of equation* (13), *as* $k \to \infty$.

Observe the lower bound of the parameter $\nu_{\max}$ given in equation (28) is not necessarily easy to calculate. In practice, we simply use a sufficiently large value to replace this theoretical bound. Nevertheless, the theoretical bound in equation (28) is critical to the convergence analysis of the PDFO algorithm.

4 Numerical experiments

In the numerical experiments, we adopt the same benchmark problem as given in Hale et al. (2015). There are eight distributed agents connected via the cloud center. Every agent i has a local decision variable $x_i = [x_{i,1}; x_{i,2}] \in \mathcal{R}^2$, which represents the position of the agent in a two-dimensional plane. The summation of the local objective functions is

$$\begin{aligned}\sum_{i=1}^{8} f_i(x_i) = {} & x_{1,1}^2 + x_{1,2}^2 + (x_{2,1} + 1)^2 + (x_{2,2} - 1)^2 \\ & + (x_{3,1} - 0.2)^2 + (x_{3,2} + 0.6)^2 + (x_{4,1} + 1.4)^2 \\ & + (x_{4,2} - 1.4)^2 + (x_{5,1} + 0.1)^2 + (x_{5,2} - 0.5)^2 \\ & + (x_{6,1} + 0.7)^2 + (x_{6,2} - 0.7)^2 + (x_{7,1} - 0.5)^2 \\ & + x_{7,2} - 1.1 + (x_{8,1} + 0.3)^2 + x_{8,2}^4.\end{aligned}$$

The global objective function is

$$h(x) = \frac{1}{200}\left(\|x_1 - x_4\|^2 + \|x_1 - x_8\|^2 + \|x_4 - x_8\|^2 \right).$$

The agents are subject to five global distance constraints

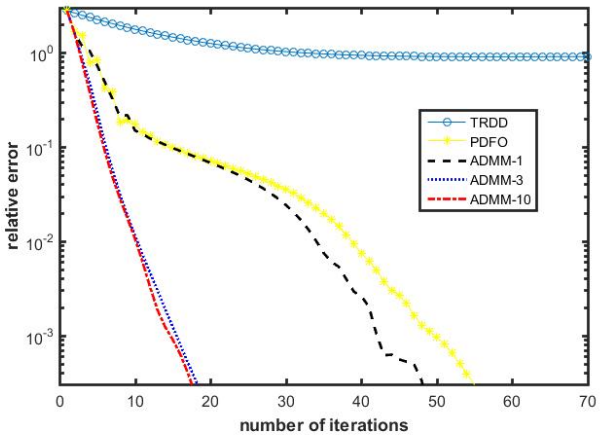
$$\begin{aligned} g_1(x) &= \|x_1 - x_2\|^2 - 0.6 \le 0, \\ g_2(x) &= \|x_1 - x_5\|^2 - 1.2 \le 0, \\ g_3(x) &= \|x_7 - x_8\|^2 - 1.8 \le 0, \\ g_4(x) &= \|x_1 - x_3\|^2 - 0.4 \le 0, \\ g_5(x) &= \|x_4 - x_6\|^2 - 0.9 \le 0. \end{aligned}$$

The variables are also subject to local box constraints, which are $x_i \in \mathcal{X}_i \triangleq [-1.5, 1.5] \times [-1, 1.5]$ for all the agents i.

We compare the proposed distributed constrained optimisation algorithms ADMM and PDFO with the Tikhonov regularised dual decomposition (TRDD) algorithm in Hale et al. (2015). The parameters of TRDD are the same as those in Hale et al. (2015): the primal regularisation constant is 0.1, the dual regularisation constant is 0.1, the primal step size is 2.804×10^{-2}, and the dual step size is 9.835×10^{-4}. The ADMM-based algorithm has three parameters: the augmented Lagrangian constant is $\rho = 1.5$, the inner loop gradient step size is $c = 0.3$, and the number of inner loop slots T is set to be 1, 3 and 10. To distinguish the algorithms with different T, we denote them as ADMM-1, ADMM-3 and ADMM-10, respectively. The PDFO method also has three parameters: the augmented Lagrangian constant is $\rho = 1.5$, the gradient step size of x update is $a = 0.4$ and that of y update is $b = 0.3$.

Convergence properties of all the algorithms are shown in Figure 2. The performance metric is relative error, which is defined as the distance between x^k and the optimal solution x^*. The proposed algorithms quickly to the optimal solution with an accuracy of 10^{-3}, ADMM-3 and ADMM-10 within 20 iterations while ADMM-1 and PDFO within 50 iterations. As a comparison, TRDD converges slowly to a neighbourhood of the optimal solution. The slow convergence is due to the small step size that is used to guarantee the computational stability of the dual decomposition method. Meanwhile, the Tikhonov regularisation essentially yields a new problem, whose optimal solution is different to that of equation (1), so that the convergence is inaccurate.

Figure 2 Relative errors of the proposed distributed constrained optimisation algorithms ADMM-T and PDFO, as well as the Tikhonov regularised dual decomposition (TRDD) algorithm. Here T denotes the number of inner loops of ADMM and its value is set to be 1, 3 and 10 (see online version for colours)



Comparing ADMM-1 and PDFO, we see that both of them do not use inner loops to update y. ADMM-1 is slightly faster than PDFO, because the former exactly updates x while the latter only runs one projected gradient step. For the ADMM-based algorithm with different numbers of inner loops T, apparently larger T yields more accurate y update, and leads to faster convergence of the outer loop iterations. Nevertheless, the side effect is higher computation cost. Our suggestion is that, if the computation cost is not an issue, we can let the cloud center accurately update x and y so as to achieve fast convergence; otherwise, PDFO and ADMM-T with small T are better approaches. Again, we emphasise that there is no convergence guarantee for ADMM-T as long as T is finite, as we have discussed in Section 2.3. PDFO converges to

the optimal solution of equation (1) as we have theoretically established in Section 3.3.

Below we consider consensus constraint violation, which is defined as the distance between x^k and y^k. In Figure 3 we only compare the proposed ADMM and PDFO algorithms, since TRDD does not involve the consensus constraint. The consensus constraint is quickly satisfied in both ADMM-3 and ADMM-10, while the violations vanish slower for ADMM-1 and PDFO. Combining Figures 2 and 3, we can conclude that in ADMM and PDFO, the primal variables x and y both converges to the optimal solution of equation (1).

Figure 3 Consensus constraint violations of the proposed distributed constrained optimisation algorithms ADMM-T and PDFO. Here T denotes the number of inner loops of ADMM and its value is set to be 1, 3 and 10 (see online version for colours)

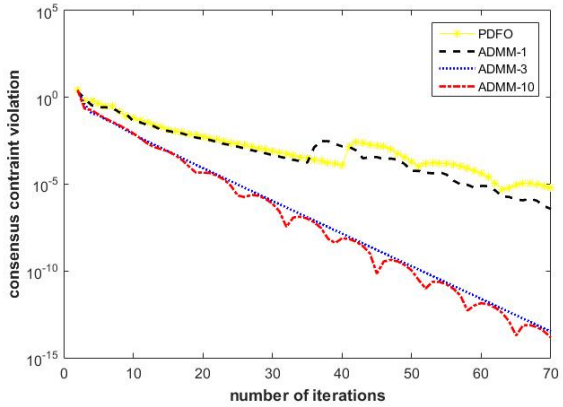
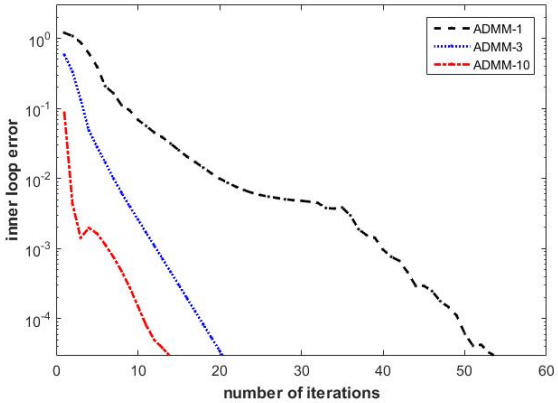


Figure 4 Inner loop errors vs. the evolution of outer loop iterations in ADMM-T, with different numbers of inner loop slots T = 1, 3 and 10 (see online version for colours)



We further investigate the impact of the number of inner loop slots T on the convergence of ADMM. Letting all the other parameters unaltered, we vary the value of T as 1, 3 and 10. Figure 4 shows the inner loop error, which is defined as the distance between the solved result and the optimal solution of equation (7). Apparently, when we spend more slots in the inner loops, the error shall become smaller. However, for any particular value of T, the error decays to zero along with the evolution of the outer loop, which explains the exact convergence of the outer loop iterations demonstrated in Figure 2. This observation implies that it is possible to prove the convergence of ADMM-T even when T is finite.

5 Conclusions

This paper considers a distributed constrained optimisation problem where a group of distributed agents are interconnected via a cloud center, and collaboratively minimise a network-wide objective function subject to local and global constraints. We introduce divide-and-conquer techniques that assign the local objective functions and constraints to the agents while the global ones to the cloud center. This yields two fully distributed constrained optimisation algorithms, one is an ADMM-based algorithm and another is a primal-dual first-order method, with an agent layer and a cloud center layer. The two layers exchange their intermediate variables so as to collaboratively obtain a network-wide optimal solution. Effectiveness of the proposed algorithms is validated by numerical experiments.

One of our future research direction is to analyse the convergence properties of ADMM when the number of inner loops is finite. Of particular interest to us is the effect of the inexact inner loops on the convergence rate and accuracy. Another topic is to consider the impact of communication delays, which are inevitable in the information exchange between the agents and the cloud center, on the performance of the proposed algorithms.

Acknowledgement

This work was supported in part by the China National Science Foundation under grant 61573331, the Anhui Provincial Natural Science Foundation under grant 1608085QF130, and the State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics and Information System under grant CEMEE2016K0202B.

References

Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I. and Zaharia, M. 'A view of cloud computing', *Communications of the ACM*, Vol. 53, pp.50–58.

Beck, A., Nedic, A., Ozdaglar, A. and Teboulle, M. (2014) 'An $O(1/k)$ gradient method for network resource allocation problems', *IEEE Transactions on Control of Network Systems*, Vol. 1, pp.64–73.

Bertsekas, D. and Tsitsiklis, J. (1997) *Parallel and Distributed Computation: Numerical Methods*, 2nd ed., Athena Scientific, Lexington, MA, USA.

Bertsekas, D. (1999) *Nonlinear Programming*, 2nd ed., Athena Scientific, Lexington, MA, USA.

Bertsekas, D., Ozdaglar, A. and Nedic, A. (2003) *Convex Analysis and Optimization*, Athena Scientific.

Bonomi, F., Milito, R., Zhu, J. and Addepalli, S. (2012) 'Fog computing and its role in the Internet of things', *Proceedings of MCC Workshop on Mobile Cloud Computing*, 17 August, Helsinki, Finland, pp.13–16.

Boyd, S. and Vandenberghe, L. (2008) *Convex Optimization*, Cambridge University Press, Cambridge, England.

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

Chang, T., Hong, M. and Wang, X. (2015) 'Multi-agent distributed optimization via inexact consensus ADMM', *IEEE Transactions on Signal Processing*, Vol. 63, pp.482–497.

Chen, T., Marques, A. and Giannakis, G. (2017) 'DGLB: Distributed stochastic geographical load balancing over cloud networks', *IEEE Transactions on Parallel and Distributed Systems*, Vol. 28, pp.1866–1880.

Chiang, M., Low, S., Calderbank, A. and Doyle, J. (2007) 'Layering as optimization decomposition: A mathematical theory of network architectures', *Proceedings of the IEEE*, Vol. 95, pp.255–312.

Deng, W. and Yin, W. (2016) 'On the global and linear convergence of the generalized alternating direction method of multipliers', *Journal of Scientific Computing*, Vol. 66, pp.889–916.

Eckstein, J. and Bertsekas, D. (1992) 'On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators', *Mathematical Programming*, Vol. 55, pp.293–318.

Gabay, D. and Mercier, B. (1976) 'A dual algorithm for the solution of nonlinear variational problems via finite element approximation', *Computers and Mathematics with Applications*, Vol. 2, pp.17–40.

Giannakis, G., Gatsis, N., Kekatos, V., Kim, S., Zhu, H. and Wollenberg, B. 'Monitoring and optimization for power systems: A signal processing perspective', *IEEE Signal Processing Magazine*, Vol. 30, pp.107–128.

Giannakis, G., Ling, Q., Mateos, G., Schizas, I. and Zhu, H. (2016) 'Decentralized learning for wireless communications and networking', in Glowinski, R., Osher, S. and Yin, W. (Eds.): *Splitting Methods in Communication and Imaging, Science and Engineering*, Springer, Gewerbestrasse, Switzerland.

Hale, M. and Egerstedt, M. (2014) 'Cloud-based optimization: a quasi-decentralized approach to multi-agent coordination', *Proceedings of IEEE Conference on Decision and Control*, 15–17 December, Los Angeles, CA, USA, pp.6635–6640.

Hale, M., Nedich, A. and Egerstedt, M. (2015) 'Cloud-based centralized/decentralized multi-agent optimization with communication delays', *Proceedings of IEEE Conference on Decision and Control*, 15–18 December, Osaka, Japan, pp.700–705.

He, B. and Yuan, X. (2012) 'On the $O(1/t)$ convergence rate of the alternating dirction method', *SIAM Journal on Numerical Analysis*, Vol. 50, pp.700–709.

Hong, M. and Luo, Z. (2017) 'On the linear convergence of the alternating direction method of multipliers', *Mathematical Programming*, Vol. 162, pp.165–199.

Li, X. and Scaglione, A. (2013) 'Robust decentralized state estimation and tracking for power systems via network gossiping', *IEEE Journal on Selected Areas in Communications*, Vol. 31, pp.1184–1194.

Ling, Q., Shi, W., Wu, G. and Ribeiro, A. (2015) 'DLM: Decentralized linearized alternating direction method of multipliers', *IEEE Transactions on Signal Processing*, Vol. 63, pp.4051–4064.

Mokhtari, A., Shi, W., Ling, Q. and Ribeiro, A. (2016) 'DQM: Decentralized quadratically approximated alternating direction method of multipliers', *IEEE Transactions on Signal Processing*, Vol. 64, pp.5158–5173.

Mokhtari, A., Koppel, A. and Ribeiro, A. (2016) 'Doubly random parallel stochastic methods for large scale learning', *Proceedings of IEEE American Control Conference*, 6–8 July, Boston, MA, USA, pp.4847–4852.

Ng, M., Wang, F. and Yuan, X. (2011) 'Inexact alternating direction methods for image recovery', *SIAM Journal on Scientific Computing*, Vol. 33, pp.1643–1668.

Shi, W., Ling, Q., Yuan, K., Wu, G. and Yin, W. (2014) 'On the linear convergence of the ADMM in decentralized consensus optimization', *IEEE Transactions on Signal Processing*, Vol. 62, pp.1750–1761 .

Tsitsiklis, J., Bertsekas, D. and Athans, M. (1986) 'Distributed asynchronous deterministic and stochastic gradient optimization algorithms', *IEEE Transactions on Automatic Control*, Vol. 31, pp.803–812.

Appendix A: Bound of optimal primal-dual solution

By Assumption 3, $\mathcal{X}$ is bounded. Therefore, $x^* \in \mathcal{X}$ and $y^* \in \mathcal{X}$ must be bounded.

Consider the augmented Lagrangian $\widetilde{\mathcal{L}}_\rho(x, y, \mu, \nu)$ defined in equation (14) and let $\rho = 0$ so that it becomes a Lagrangian function

$$\widetilde{\mathcal{L}}_0(x, y, \mu, \nu) = f(x) + h(y) + \sum_{j=1}^{m} \nu_j g_j(y) + \langle \mu, x - y \rangle,$$

where $x, y \in \mathcal{X}$. Given any dual variables $\bar{\mu} \in \mathcal{R}^p$ and $\bar{\nu} \in \mathcal{R}^m_+$, define

$$q(\bar{\mu}, \bar{\nu}) = \min_{x,y \in \mathcal{X}} \widetilde{\mathcal{L}}_0(x, y, \bar{\mu}, \bar{\nu}).$$

It is easy to verify that for any optimal primal-dual solution (x^*, y^*, μ^*, ν^*), we have

$$\begin{aligned} q(\bar{\mu}, \bar{\nu}) &\le \widetilde{\mathcal{L}}_0(x^*, y^*, \bar{\mu}, \bar{\nu}) \\ &\le \widetilde{\mathcal{L}}_0(x^*, y^*, \mu^*, \nu^*) \\ &\le \widetilde{\mathcal{L}}_0(\bar{x}, \bar{y}, \mu^*, \nu^*), \end{aligned} \tag{A1}$$

where $(\bar{x}, \bar{y})$ is chosen as any pair of slater point of equation (13). Applying the definition of slater point to the right hand side of equation (A1) yields

$$q(\bar{\mu}, \bar{\nu}) \le f(\bar{x}) + h(\bar{y}) + \sum_{j=1}^{m} \nu_j^* g_j(\bar{y}). \tag{A2}$$

Changing the order of equation (A2), we have

$$\sum_{j=1}^{m} \nu_j^* (-g_j(\bar{y})) \le f(\bar{x}) + h(\bar{y}) - q(\bar{\mu}, \bar{\nu}). \tag{A3}$$

Observing that $\nu_j^* \geq 0$ and $-g_j(\bar{y}) > 0$, $j = 1, \ldots, m$, we obtain

$$\sum_{j=1}^{m} \nu_j^* \leq \frac{f(\bar{x}) + h(\bar{y}) - q(\bar{\mu}, \bar{\nu})}{\min_j(-g_j(\bar{y}))}. \tag{A4}$$

Letting $\bar{\mu} = 0$ and $\bar{\nu} = 0$ in equation (A4), it holds

$$\sum_{j=1}^{m} \nu_j^* \leq \frac{f(\bar{x}) + h(\bar{y}) - (\min_{x,y\in\mathcal{X}}\{f(x) + h(y)\})}{\min_j(-g_j(\bar{y}))}, \tag{A5}$$

Since $\nu_j^* \geq 0$, for all $j = 1, \ldots, m$, equation (A5) implies that

$$\nu_j^* \leq \frac{f(\bar{x}) + h(\bar{y}) - (\min_{x,y\in\mathcal{X}}\{f(x) + h(y)\})}{\min_j(-g_j(\bar{y}))}. \tag{A6}$$

To bound μ^*, define a sign vector $s_{\mu^*} \in \mathcal{R}^p$ whose element is $+1$ if the corresponding element of μ^* is positive, -1 if the corresponding element of μ^* is negative, and 0 otherwise. Apparently, $\langle \mu^*, s_{\mu^*} \rangle = \|\mu^*\|_1$. Consider any pair of slater point of equation (13) $(\bar{x}, \bar{y})$. Due to the definition of slater point, we know that $\bar{x}$ satisfies $\bar{x} \in \mathcal{X}$ and $g_j(\bar{x}) < 0$ for all $j = 1, \ldots, m$. By the continuity of $g_j(x)$, there exists a sufficiently small positive constant κ such that $\bar{x} - \kappa s_{\mu^*} \in \mathcal{X}$ and $g_j(\bar{x} - \kappa s_{\mu^*}) < 0$. Therefore, similar to equation (A1), we have

$$q(\bar{\mu}, \bar{\nu}) \leq \widetilde{\mathcal{L}}_0(\bar{x} - \kappa s_{\mu^*}, \bar{y}, \mu^*, \nu^*), \tag{A7}$$

Expanding the right hand side of equation (A1) yields

$$\begin{aligned} q(\bar{\mu}, \bar{\nu}) &\leq f(\bar{x} - \kappa s_{\mu^*}) + h(\bar{y}) + \sum_{j=1}^{m} \nu_j^* g_j(\bar{y}) - \kappa\langle \mu^*, s_{\mu^*} \rangle \\ &\leq f(\bar{x} - \kappa s_{\mu^*}) + h(\bar{y}) - \kappa\|\mu^*\|_1, \end{aligned} \tag{A8}$$

where the second inequality is because $\nu_j^* \geq 0$ and $g_j(\bar{y}) < 0$. Letting $\bar{\mu} = 0$ and $\bar{\nu} = 0$ in equation (A8), it holds

$$\begin{aligned} &\|\mu^*\|_1 \\ \leq &\frac{1}{\kappa}\Big\{ f(\bar{x} - \kappa s_{\mu^*}) + h(\bar{y}) - (\min_{x,y\in\mathcal{X}}\{f(x) + h(y)\}) \Big\}. \end{aligned} \tag{A9}$$

Appendix B: Proof of Theorem 1

We give a key lemma before proving Theorem 1.

Lemma 2: *Under Assumptions 1–3, for any* (x^*, y^*, μ^*, ν^*) *that is an optimal primal-dual solution of equation* (13)*, the sequence* (x^k, y^k, μ^k, ν^k) *generated by the PDFO recursion* (15)–(17) *and* (19) *satisfies*

$$\begin{aligned} &\frac{1}{2}\left(\frac{1}{a} - L_f - \rho\right)\|x^{k+1} - x^k\|^2 \\ &+ \frac{1}{2}\left(\frac{1}{b} - \nu_{\max}\sum_{j=1}^{m} L_{g_j} - L_h - \rho\right)\|y^{k+1} - y^k\|^2 \\ &+ \frac{1}{2}\|\frac{1}{\sqrt{\rho}}(\mu^{k+1} - \mu^k) + \sqrt{\rho}(y^{k+1} - y^k)\|^2 \\ \leq &\frac{1}{2}\left(\frac{1}{a} - \rho\right)(\|x^k - x^*\|^2 - \|x^{k+1} - x^*\|^2) \\ &+ \frac{1}{2}(\frac{1}{b} - b\delta)(\|y^k - y^*\|^2 - \|y^{k+1} - y^*\|^2) \\ &- \frac{1}{2}(M_h - b\delta)\|y^k - y^*\|^2 \\ &+ \frac{1}{2\rho}(\|\mu^k - \mu^*\|^2 - \|\mu^{k+1} - \mu^*\|^2) \\ &+ \frac{1}{2b}(\|\nu^k - \nu^*\|^2 - \|\nu^{k+1} - \nu^*\|^2). \end{aligned} \tag{B1}$$

Proof of Lemma 2: The proof contains five steps.

Step 1. Define an auxiliary function

$$\begin{aligned} \phi(x) = f(x) &+ \langle x, \mu^{k+1} + \rho(y^{k+1} - y^k)\rangle \\ &+ \frac{\rho}{2}\|x - x^{k+1}\|^2. \end{aligned}$$

Taking the gradient of $\phi(x)$ at $x = x^k$ yields

$$\begin{aligned} \nabla\phi(x^k) &= \nabla f(x^k) + \mu^{k+1} + \rho(y^{k+1} - y^k + x^k - x^{k+1}) \\ &= \nabla f(x^k) + \mu^k + \rho(x^k - y^k), \end{aligned}$$

where the second equality is due to the update $\mu^{k+1} = \mu^k + \rho(x^{k+1} - y^{k+1})$ in equation (17). Thus, it is easy to verify that the updates of x_i in equation (15) are equivalent to

$$\begin{aligned} x^{k+1} &= P_{\mathcal{X}}\left[x^k - a\nabla\phi(x^k)\right] \\ &= \arg\min_{x\in\mathcal{X}} \|x - x^k + a\nabla\phi(x^k)\|^2 \\ &= \arg\min_{x\in\mathcal{X}} \phi(x^k) + \langle x - x^k, \nabla\phi(x^k)\rangle \\ &\quad + \frac{1}{2a}\|x - x^k\|^2. \end{aligned} \tag{B2}$$

Consider the function $\tilde{\phi}(x) \triangleq \phi(x^k) + \langle x - x^k, \nabla\phi(x^k)\rangle + \|x - x^k\|^2/(2a)$ that is strongly convex with constant $1/a$. Since x^{k+1} is its minimum point within the set $\mathcal{X}$, thus for any $x \in \mathcal{X}$, it must hold

$$\begin{aligned} \tilde{\phi}(x) \geq &\tilde{\phi}(x^{k+1}) + \langle x - x^{k+1}, \nabla\tilde{\phi}(x^{k+1})\rangle \\ &+ \frac{1}{2a}\|x - x^{k+1}\|^2 \\ \geq &\tilde{\phi}(x^{k+1}) + \frac{1}{2a}\|x - x^{k+1}\|^2. \end{aligned} \tag{B3}$$

Here the first inequality comes from the strong convexity of $\tilde{\phi}(x)$, while the second inequality comes from the

first-order optimality condition $\langle x - x^{k+1}, \nabla\tilde{\phi}(x^{k+1})\rangle \geq 0$. Substituting $x = x^* \in \mathcal{X}$ into equation (B3) yields $\tilde{\phi}(x^*) \geq \tilde{\phi}(x^{k+1}) + \|x^{k+1} - x^*\|^2/(2a)$ and consequently

$$\begin{aligned}&\phi(x^k) + \langle x^{k+1} - x^k, \nabla\phi(x^k)\rangle + \frac{1}{2a}\|x^{k+1} - x^k\|^2\\ \leq&\phi(x^k) + \langle x^k - x^*, \nabla\phi(x^k)\rangle\\ &+ \frac{1}{2a}\|x^k - x^*\|^2 - \frac{1}{2a}\|x^{k+1} - x^*\|^2.\end{aligned} \tag{B4}$$

Now we look for an upper bound for the right hand side of equation (B4) and a lower bound for the left hand side. Since $f(x)$ is convex and has Lipschitz continuous gradient with constant L_f, $\phi(x)$ is strongly convex with constant ρ and has Lipschitz continuous gradient with constant $L_f + \rho$. From the strong convexity of $\phi(x)$ we know

$$\phi(x^k) - \langle x^k - x^*, \nabla\phi(x^k)\rangle \leq \phi(x^*) - \frac{\rho}{2}\|x^k - x^*\|^2.$$

From the Lipschitz continuity of $\nabla\phi(x)$ we know

$$\begin{aligned}&\phi(x^k) + \langle x^{k+1} - x^k, \nabla\phi(x^k)\rangle\\ \geq&\phi(x^{k+1}) - \frac{L_f + \rho}{2}\|x^{k+1} - x^k\|^2.\end{aligned}$$

Substituting this inequality into the two sides of equation (B4) yields

$$\begin{aligned}&\phi(x^{k+1}) + \left(\frac{1}{2a} - \frac{L_f}{2} - \frac{\rho}{2}\right)\|x^{k+1} - x^k\|^2\\ \leq&\phi(x^*) - \left(\frac{\rho}{2} - \frac{1}{2a}\right)\|x^k - x^*\|^2 - \frac{1}{2a}\|x^{k+1} - x^*\|^2.\end{aligned} \tag{B5}$$

Applying the definition of $\phi(x)$, from equation (B5) we have

$$\begin{aligned}&\left(\frac{1}{2a} - \frac{L_f}{2} - \frac{\rho}{2}\right)\|x^{k+1} - x^k\|^2 + f(x^{k+1}) - f(x^*)\\ \leq&\left(\frac{1}{2a} - \frac{\rho}{2}\right)\left(\|x^k - x^*\|^2 - \|x^{k+1} - x^*\|^2\right)\\ &- \langle x^{k+1} - x^*, \mu^{k+1} + \rho(y^{k+1} - y^k)\rangle.\end{aligned} \tag{B6}$$

Step 2. Define another auxiliary function

$$\psi(y) = h(y) - \langle y, \mu^{k+1}\rangle + \langle G(y), \nu\rangle + \frac{\rho}{2}\|y - y^{k+1}\|^2.$$

Similar to what we have done for $\phi(x)$ above, we are able to obtain

$$\begin{aligned}&\left(\frac{1}{2b} - \frac{L_h}{2} - \frac{\rho}{2} - \frac{\langle L_G, \nu^k\rangle}{2}\right)\|y^{k+1} - y^k\|^2\\ &+ h(y^{k+1}) - h(y^*)\\ \leq&\left(\frac{1}{2b} - \frac{\rho}{2}\right)\left(\|y^k - y^*\|^2 - \|y^{k+1} - y^*\|^2\right)\\ &- \frac{M_h}{2}\|y^k - y^*\|^2\\ &+ \langle y^{k+1} - y^*, \mu^{k+1}\rangle + \langle G(y^*) - G(y^{k+1}), \nu^k\rangle.\end{aligned} \tag{B7}$$

Step 3. Observe that $f(x^{k+1}) - f(x^*)$ and $h(y^{k+1}) - h(y^*)$ appear in the left hand sides of equation (B6) and (B7), respectively. They denote the gaps between the current primal iterate (x^{k+1}, y^{k+1}) to the optimal point (x^*, y^*) measured by function values. Below we bound these function values. Consider the augmented Lagrangian $\widetilde{\mathcal{L}}_\rho(x, y, \mu, \nu)$ defined in equation (14) and let $\rho = 0$ so that it becomes a Lagrangian function. Apparently, for any $x^{k+1}, y^{k+1} \in \mathcal{X}$, it holds

$$\widetilde{\mathcal{L}}_0(x^*, y^*, \mu^*, \nu^*) \leq \widetilde{\mathcal{L}}_0(x^{k+1}, y^{k+1}, \mu^*, \nu^*).$$

Expanding the inequality yields

$$\begin{aligned}&f(x^*) + h(y^*) + \langle G(y^*), \nu^*\rangle + \langle x^* - y^*, \mu^*\rangle\\ \leq&f(x^{k+1}) + h(y^{k+1}) + \langle G(y^{k+1}), \nu^*\rangle\\ &+ \langle \mu^*, x^{k+1} - y^{k+1}\rangle.\end{aligned} \tag{B8}$$

Using the facts $x^* = y^*$ and $\langle G(y^*), \nu^*\rangle = 0$, we reorganize the terms of equation (B8) and obtain

$$\begin{aligned}&f(x^*) - f(x^{k+1}) + h(y^*) - h(y^{k+1})\\ \leq&\langle G(y^{k+1}) - G(y^*), \nu^*\rangle + \langle \mu^*, x^{k+1} - y^{k+1}\rangle.\end{aligned} \tag{B9}$$

Step 4. Consider the terms $\langle G(y^*) - G(y^{k+1}), \nu^k\rangle$ and $\langle G(y^{k+1}) - G(y^*), \nu^*\rangle$ appearing at the right hand sides of equation (B7) and (B9), respectively. We shall bound their values. Due to the ν_j update (19) we have $\nu_j^{k+1} = \mathcal{P}_{\mathcal{D}}\left[\nu_j^k + bg_j(y^{k+1})\right]$. Meanwhile, for the optimal primal variable ν^* it must hold $\nu_j^* = \mathcal{P}_{\mathcal{D}}\left[\nu_j^* + bg_j(y^*)\right]$. The non-expansive property of the projection $\mathcal{P}_{\mathcal{D}}$ implies

$$(\nu_j^{k+1} - \nu_j^*)^2 \leq (\nu_j^k - \nu_j^* + b(g_j(y^{k+1}) - g_j(y^*)))^2,$$

and consequently

$$\|\nu^{k+1} - \nu^*\|^2 \leq \|\nu^k - \nu^* + b(G(y^{k+1}) - G(y^*))\|^2.$$

Expanding the inequality and changing the order, we have

$$\begin{aligned}&\langle G(y^*) - G(y^{k+1}), \nu^k - \nu^*\rangle\\ \leq&\frac{1}{2b}(\|\nu^k - \nu^*\|^2 - \|\nu^{k+1} - \nu^*\|^2)\\ &+ \frac{b}{2}\|G(y^{k+1}) - G(y^*)\|^2.\end{aligned} \tag{B10}$$

By equation (26) in Assumption 3, it holds $\|G(y^{k+1}) - G(y^*)\|^2 \leq \delta\|y^{k+1} - y^*\|^2$. Substituting it into equation (B10) yields

$$\begin{aligned}&\langle G(y^*) - G(y^{k+1}), \nu^k - \nu^*\rangle\\ \leq&\frac{1}{2b}(\|\nu^k - \nu^*\|^2 - \|\nu^{k+1} - \nu^*\|^2)\\ &+ \frac{b\delta}{2}\|y^{k+1} - y^*\|^2.\end{aligned} \tag{B11}$$

Step 5. Summing up the inequalities (B6), (B7), (B9) and (B11) that are results of the above four steps and noticing that $x^* = y^*$, we have

$$\begin{aligned}
&\left(\frac{1}{2b} - \frac{L_h}{2} - \frac{\rho}{2} - \frac{\langle L_G, \nu^k\rangle}{2}\right)\|y^{k+1} - y^k\|^2 \\
&+ \left(\frac{1}{2a} - \frac{L_f}{2} - \frac{\rho}{2}\right)\|x^{k+1} - x^k\|^2 \\
\leq &\left(\frac{1}{2b} - \frac{\rho}{2}\right)\left(\|y^k - y^*\|^2 - \|y^{k+1} - y^*\|^2\right) \\
&- \frac{M_h}{2}\|y^k - y^*\|^2 \\
&+ \left(\frac{1}{2a} - \frac{\rho}{2}\right)\left(\|x^k - x^*\|^2 - \|x^{k+1} - x^*\|^2\right) \\
&+ \frac{1}{2b}(\|\nu^k - \nu^*\|^2 - \|\nu^{k+1} - \nu^*\|^2) \\
&+ \frac{b\delta}{2}\|y^* - y^{k+1}\|^2 \\
&+ \langle \mu^* - \mu^{k+1}, x^{k+1} - y^{k+1}\rangle \\
&+ \rho\langle y^{k+1} - y^k, x^* - x^{k+1}\rangle.
\end{aligned} \tag{B12}$$

Now we go to handle the terms $\langle \mu^* - \mu^{k+1}, x^{k+1} - y^{k+1}\rangle$ and $\rho\langle y^{k+1} - y^k, x^* - x^{k+1}\rangle$ at the right hand side of equation (B12). For $\langle \mu^* - \mu^{k+1}, x^{k+1} - y^{k+1}\rangle$, using the μ update $\mu^{k+1} = \mu^k + \rho(x^{k+1} - y^{k+1})$ in equation (17), we have

$$\begin{aligned}
&\langle \mu^* - \mu^{k+1}, x^{k+1} - y^{k+1}\rangle \\
=&\frac{1}{\rho}\langle \mu^* - \mu^{k+1}, \mu^{k+1} - \mu^k\rangle \\
=&\frac{1}{2\rho}(\|\mu^k - \mu^*\|^2 - \|\mu^{k+1} - \mu^*\|^2 - \|\mu^{k+1} - \mu^k\|^2).
\end{aligned} \tag{B13}$$

For $\rho\langle y^{k+1} - y^k, x^* - x^{k+1}\rangle$, since $x^* = y^*$ we can write $x^* - x^{k+1}$ as the summation of $y^* - y^{k+1}$ and $y^{k+1} - x^{k+1}$, and hence obtain

$$\begin{aligned}
\rho\langle y^{k+1} - y^k, x^* - x^{k+1}\rangle = &\rho\langle y^{k+1} - y^k, y^* - y^{k+1}\rangle \\
&+ \rho\langle y^{k+1} - y^k, y^{k+1} - x^{k+1}\rangle.
\end{aligned} \tag{B14}$$

Similar to what we have had in equation (B13), the first term at the right hand side of equation (B14) is

$$\begin{aligned}
&\rho\langle y^{k+1} - y^k, y^* - y^{k+1}\rangle \\
=&\frac{\rho}{2}(\|y^k - y^*\|^2 - \|y^{k+1} - y^*\|^2 - \|y^{k+1} - y^k\|^2).
\end{aligned} \tag{B15}$$

Again, applying the μ update $\mu^{k+1} = \mu^k + \rho(x^{k+1} - y^{k+1})$ in equation (17) to the second term at the right hand side of equation (B14) yields

$$\begin{aligned}
&\rho\langle y^{k+1} - y^k, y^{k+1} - x^{k+1}\rangle \\
=&-\langle y^{k+1} - y^k, \mu^{k+1} - \mu^k\rangle \\
=&\frac{1}{2\rho}\|\mu^{k+1} - \mu^k\|^2 + \frac{\rho}{2}\|y^{k+1} - y^k\|^2 \\
&- \frac{1}{2}\|\frac{1}{\sqrt{\rho}}(\mu^{k+1} - \mu^k) + \sqrt{\rho}(y^{k+1} - y^k)\|^2.
\end{aligned} \tag{B16}$$

Substituting equations (B15) and (B16) into equation (B14) and then combining with equation (B13), we have

$$\begin{aligned}
&\langle \mu^* - \mu^{k+1}, x^{k+1} - y^{k+1}\rangle \\
&+ \rho\langle y^{k+1} - y^k, y^{k+1} - x^{k+1}\rangle \\
=&\frac{1}{2\rho}(\|\mu^k - \mu^*\|^2 - \|\mu^{k+1} - \mu^*\|^2) \\
&+ \frac{\rho}{2}(\|y^* - y^k\|^2 - \|y^* - y^{k+1}\|^2) \\
&- \frac{1}{2}\|\frac{1}{\sqrt{\rho}}(\mu^{k+1} - \mu^k) + \sqrt{\rho}(y^{k+1} - y^k)\|^2.
\end{aligned} \tag{B17}$$

Substituting equation (B17) into equation (B12), noticing the fact that $\nu_j^k \in [0, \nu_{\max}]$, $j = 1, \ldots, n$, and reorganizing the terms, we obtain the main result (B1) and complete the proof. □

Now we are ready to prove Theorem 1.

Proof of Theorem 1: By equation (27), we have

$$\begin{aligned}
&\frac{1}{a} - L_f - \rho > 0,\ \frac{1}{b} - \nu_{\max}\sum_{j=1}^{m} L_{g_j} - L_h - \rho > 0, \\
&\frac{1}{a} - \rho > 0,\ \frac{1}{b} - b\delta > 0,\ M_h - b\delta > 0.
\end{aligned} \tag{B18}$$

Such values of a, b and ρ guarantee all other coefficients in equation (B1) to be positive, except that $-(M_h - b\delta)/2$ is negative.

Summing up equation (B1) from time $k = 0$ to $k = K$ yields

$$\begin{aligned}
&\sum_{k=0}^{K}\Big\{\frac{1}{2}\left(\frac{1}{a} - L_f - \rho\right)\|x^{k+1} - x^k\|^2 \\
&+ \frac{1}{2}\left(\frac{1}{b} - \nu_{\max}\sum_{j=1}^{m} L_{g_j} - L_h - \rho\right)\|y^{k+1} - y^k\|^2 \\
&+ \frac{1}{2}\|\frac{1}{\sqrt{\rho}}(\mu^{k+1} - \mu^k) + \sqrt{\rho}(y^{k+1} - y^k)\|^2\Big\} \\
\leq &\frac{1}{2}\left(\frac{1}{a} - \rho\right)(\|x^0 - x^*\|^2 - \|x^{K+1} - x^*\|^2) \\
&+ \frac{1}{2}\left(\frac{1}{b} - b\delta\right)(\|y^0 - y^*\|^2 - \|y^{K+1} - y^*\|^2) \\
&+ \frac{1}{2\rho}(\|\mu^0 - \mu^*\|^2 - \|\mu^{K+1} - \mu^*\|^2) \\
&+ \frac{1}{2b}(\|\nu^0 - \nu^*\|^2 - \|\nu^{K+1} - \nu^*\|^2) \\
&- \sum_{k=0}^{K}\frac{1}{2}(M_h - b\delta)\|y^k - y^*\|^2.
\end{aligned} \tag{B19}$$

Throwing away some terms in equation (B19) and only keeping those of $\|x^0 - x^*\|^2, \|y^0 - y^*\|^2, \|\mu^0 - \mu^*\|^2, \|\nu^0 - \nu^*\|^2$ and $\|\nu^{K+1} - \nu^*\|^2$, we have

$$\begin{aligned}
\frac{1}{2b}\|\nu^{K+1} - \nu^*\|^2 \leq &\frac{1}{2b}\|\nu^0 - \nu^*\|^2 \\
&+ \frac{1}{2}\left(\frac{1}{a} - \rho\right)\|x^0 - x^*\|^2
\end{aligned}$$

$$+\frac{1}{2}\left(\frac{1}{b}-b\delta\right)\|y^0-y^*\|^2 \\ ++\frac{1}{2\rho}\|\mu^0-\mu^*\|^2.$$

It is easy to verify that for every j and K, we have

$$\nu_j^{K+1} \le \|\nu^*\| + \left(\|\nu^0-\nu^*\|^2 + \left(\frac{b}{a}-b\rho\right)\|x^0-x^*\|^2 \right. \\ \left. +(1-b^2\delta)\|y^0-y^*\|^2 + \frac{b}{\rho}\|\mu^0-\mu^*\|^2\right)^{\frac{1}{2}} \quad \text{(B20)} \\ \triangleq \nu_{\text{upp}}.$$

With particular note, equation (B20) implies that, if the parameters of the PDFO algorithm is properly chosen, the dual variables ν_j^k shall not exceed the upper bound ν_{upp}. Recall the ν_j update $\nu_j^{k+1} = \mathcal{P}_{\mathcal{D}}\left[\nu_j^k + bg_j(y^{k+1})\right]$ in equation (19), where $\mathcal{D} \triangleq [0, \nu_{\max}]$. Therefore, if we choose $\nu_{\max}$ to be strictly larger than ν_{upp} as in equation (28), the projection on the boundary of $\nu_{\max}$ will never happen and the ν_j update can be written as $\nu_j^{k+1} = \left[\nu_j^k + bg_j(y^{k+1})\right]_+$. This is critical to the following proof.

Now we manipulate equation (B19) again. Throwing away the nonpositive terms at the right hand side of equation (B19), we have

$$\sum_{k=0}^{K}\Big\{\frac{1}{2}\left(\frac{1}{a}-L_f-\rho\right)\|x^{k+1}-x^k\|^2 \\ +\frac{1}{2}\left(\frac{1}{b}-\nu_{\max}\sum_{j=1}^{m}L_{g_j}-L_h-\rho\right)\|y^{k+1}-y^k\|^2 \\ +\frac{1}{2}\|\frac{1}{\sqrt{\rho}}(\mu^{k+1}-\mu^k)+\sqrt{\rho}(y^{k+1}-y^k)\|^2\Big\} \quad \text{(B21)} \\ \le \frac{1}{2}\left(\frac{1}{a}-\rho\right)\|x^0-x^*\|^2+\frac{1}{2}\left(\frac{1}{b}-b\delta\right)\|y^0-y^*\|^2 \\ +\frac{1}{2\rho}\|\mu^0-\mu^*\|^2+\frac{1}{2b}\|\nu^0-\nu^*\|^2.$$

Observe that the right hand side of equation (B21) is a nonnegative constant determined by the initial point (x^0, y^0, μ^0, ν^0) and the optimal solution (x^*, y^*, μ^*, ν^*). Letting $K \to \infty$ and noticing that all the coefficients in equation (B21) are positive, we know that in the limit (namely, $k \to \infty$) the following equalities must hold

$$x^{k+1} = x^k,\ y^{k+1} = y^k,\ \mu^{k+1} = \mu^k. \quad \text{(B22)}$$

The fact of $y^{k+1} = y^k$ when $k \to \infty$ and the continuity of $g_j(y)$ guarantee that $g_j(y^{k+1}) = g_j(y^k)$ when $k \to \infty$. According to $\nu_j^{k+1} = \mathcal{P}_{\mathcal{D}}\left[\nu_j^k + bg_j(y^{k+1})\right]$ in equation (19), we have $\nu_j^{k+1} = \nu_j^k$ when $k \to \infty$ for all $j = 1, \ldots, m$. Consequently, it holds in the limit that

$$\nu^{k+1} = \nu^k. \quad \text{(B23)}$$

Combining equations (B22) and (B23), we conclude that the PDFO iterate (x^k, y^k, μ^k, ν^k) converges to a stationary point

$$\lim_{k\to\infty}(x^k, y^k, \mu^k, \nu^k) = (x^\infty, y^\infty, \mu^\infty, \nu^\infty). \quad \text{(B24)}$$

To prove Theorem 1, it remains to show that the stationary point $(x^\infty, y^\infty, \mu^\infty, \nu^\infty)$ satisfies the KKT conditions of equation (13). Below we check its primal feasibility, dual feasibility, complimentarity slackness and stationarity (Boyd and Vandenberghe, 2004).

Primal feasibility. The projections onto $\mathcal{X}$ in the x update in equation (15) and the y update in equation (16) guarantee that $x^\infty, y^\infty \in \mathcal{X}$. Consider the μ update in equation (17) at its limit form $\mu^\infty = \mu^\infty + \rho(x^\infty - y^\infty)$, it holds $x^\infty = y^\infty$. We show $g_j(y^\infty) \le 0$ by contradiction. If $g_j(y^\infty) > 0$, then by the ν update $\nu_j^{k+1} = \mathcal{P}_{\mathcal{D}}\left[\nu_j^k + bg_j(y^{k+1})\right]$ in equation (19), ν_j^∞ must arrive at $\nu_{\max}$, which is the upper bound of $\mathcal{D}$. However, since $\nu_j^\infty \le \nu_{\text{upp}}$ as we have proved in equation (B20) and $\nu_{\text{upp}} < \nu_{\max}$ as we have assumed in equation (28), there exists contradiction and $g_j(y^\infty) \le 0$ must hold.

Dual feasibility. This is trivial because by the projection operator of the ν update $\nu_j^{k+1} = \mathcal{P}_{\mathcal{D}}\left[\nu_j^k + bg_j(y^{k+1})\right]$ in equation (19), where $\mathcal{D} \triangleq [0, \nu_{\max}]$, it holds $\nu_j^\infty \ge 0$, for all $j = 1, \ldots, m$.

Complimentarity slackness. If $g_j(y^\infty) < 0$, the update $\nu_j^{k+1} = \mathcal{P}_{\mathcal{D}}\left[\nu_j^k + bg_j(y^{k+1})\right]$ in equation (19) drives ν_j^∞ to 0. Because $g_j(y^\infty) \le 0$, $g_j(y^\infty)\nu_j^\infty = 0$ holds for all $j = 1, \ldots, m$.

Stationarity. When $k \to \infty$, the x update in equation (15) and the y update in equation (16) are equivalent to

$$x^\infty = \mathcal{P}_{\mathcal{X}}\left[x^\infty - a\nabla_{x^\infty}\widetilde{\mathcal{L}}_\rho(x^\infty, y^\infty, \mu^\infty, \nu^\infty)\right],$$

$$y^\infty = \mathcal{P}_{\mathcal{X}}\left[y^\infty - b\nabla_{y^\infty}\widetilde{\mathcal{L}}_\rho(x^\infty, y^\infty, \mu^\infty, \nu^\infty)\right].$$

Therefore, for any $x, y \in \mathcal{X}$, we have

$$\langle x - x^\infty, \nabla_{x^\infty}\widetilde{\mathcal{L}}_\rho(x^\infty, y^\infty, \mu^\infty, \nu^\infty)\rangle \ge 0, \\ \langle y - y^\infty, \nabla_{y^\infty}\widetilde{\mathcal{L}}_\rho(x^\infty, y^\infty, \mu^\infty, \nu^\infty)\rangle \ge 0.$$

Thus, the stationary point $(x^\infty, y^\infty, \mu^\infty, \nu^\infty)$ of the PDFO algorithm satisfies the KKT conditions of equation (13), which completes the proof. □