Automatica 146 (2022) 110551

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Proximal ADMM for nonconvex and nonsmooth optimization ☆

Yu Yang [a,*], Qing-Shan Jia [b], Zhanbo Xu [a], Xiaohong Guan [a,b], Costas J. Spanos [c]

[a] School of Automation Science and Engineering, Xi'an Jiaotong University, Shaanxi, China
[b] CFINS, Department of Automation, BNRist, Tsinghua University, Beijing, China
[c] Electrical Engineering and Computer Sciences, University of California, Berkeley, United States of America

ARTICLE INFO

Article history:
Received 3 January 2022
Received in revised form 4 May 2022
Accepted 19 June 2022
Available online 13 September 2022

Keywords:
Distributed nonconvex and nonsmooth optimization
Proximal ADMM
Bounded Lagrangian multipliers
Global convergence
Smart buildings

ABSTRACT

By enabling the nodes or agents to solve small-sized subproblems to achieve coordination, distributed algorithms are favored by many networked systems for efficient and scalable computation. While for convex problems, substantial distributed algorithms are available, the results for the more broad nonconvex counterparts are extremely lacking. This paper develops a distributed algorithm for a class of nonconvex and nonsmooth problems featured by (i) a nonconvex objective formed by both separate and composite components regarding the decision variables of interconnected agents, (ii) local bounded convex constraints, and (iii) coupled linear constraints. This problem is directly originated from smart buildings and is also broad in other domains. To provide a distributed algorithm with convergence guarantee, we revise the powerful alternating direction method of multiplier (ADMM) method and proposed a proximal ADMM. Specifically, noting that the main difficulty to establish the convergence for the nonconvex and nonsmooth optimization with ADMM is to assume the boundness of dual updates, we propose to update the dual variables in a discounted manner. This leads to the establishment of a so-called sufficiently decreasing and lower bounded Lyapunov function, which is critical to establish the convergence. We prove that the method converges to some approximate stationary points. We besides showcase the efficacy and performance of the method by a numerical example and the concrete application to multi-zone heating, ventilation, and air-conditioning (HVAC) control in smart buildings.

1. Introduction

By enabling the nodes or agents to solve small-sized subproblems to achieve coordination, distributed algorithms are favored by many networked systems to achieve efficient and scalable computation. While distributed algorithms for convex optimization have been studied extensively (Deng, Lai, Peng, & Yin, 2017; Falsone, Notarnicola, Notarstefano, & Prandini, 2020; Shi, Ling, Yuan, Wu, & Yin, 2014), the results for the more broad nonconvex counterparts are extremely lacking. The direct extension of distributed algorithms for convex problems to nonconvex counterparts is in general not applicable either due to the failure of convergence or the lack of theoretical convergence guarantee (see Houska, Frasch, and Diehl (2016) and Wang, Yin, and Zeng (2019) for some divergent examples). This paper focuses on developing a distributed algorithm for a class of nonconvex and nonsmooth problems in the canonical form of

$$\min_{\mathbf{x}=(\mathbf{x}_i)_{i=1}^N} \; F(\mathbf{x}) = g(\mathbf{x}) + \sum_{i=1}^N f_i(\mathbf{x}_i) \tag{P}$$

$$\text{s.t.} \; \sum_{i=1}^N \mathbf{A}_i\mathbf{x}_i = \mathbf{b}. \tag{1a}$$

$$\mathbf{x}_i \in \mathbf{X}_i, \; i = 1, 2, \ldots, N. \tag{1b}$$

where $i = 1, 2, \ldots, N$ denotes the computing nodes or agents, $\mathbf{x}_i \in \mathbf{R}^{n_i}$ is the local decision variables of agent i and $\mathbf{x} = (\mathbf{x}_i)_{i=1}^N \in \mathbf{R}^n$ with $n = \sum_{i=1}^N n_i$ stacks the decision variables of all agents. We have $f_i : \mathbf{R}^{n_i} \to \mathbf{R}$ and $g : \mathbf{R}^n \to \mathbf{R}$ denote the separate and composite objective components, which are continuously differentiable but possibly nonconvex. We have $\mathbf{X}_i$ represent the local bounded and convex constraints of agent i. As expressed by the formulation, the agents are expected to optimize their local decision variables in a cooperative manner so as to achieve the optimal system performance measured by $F(\mathbf{x}) = g(\mathbf{x}) + \sum_{i=1}^N f_i(\mathbf{x}_i)$ considering both their local constraints

☆ The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Kok Lay Teo under the direction of Editor Ian R. Petersen.

* Corresponding author.
E-mail addresses: yangyu21@xjtu.edu.cn (Y. Yang), jiaqs@tsinghua.edu.cn (Q.-S. Jia), zhanbo.xu@xjtu.edu.cn (Z. Xu), xhguan@xjtu.edu.cn (X. Guan), spanos@berkeley.edu (C.J. Spanos).

https://doi.org/10.1016/j.automatica.2022.110551
0005-1098/

$\mathbf{X}_i$ and the global coupled linear constraints (1a) encoded by $\mathbf{A}_i \in \mathbf{R}^{m\times n_i}$ and $b \in \mathbf{R}^m$. By defining $\mathbf{A} = (\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_N) \in \mathbf{R}^{m\times n}$ and $f(\mathbf{x}) = \sum_{i=1}^N f_i(\mathbf{x}_i)$, the coupled constraints and objective can be expressed by $\mathbf{Ax} = \mathbf{b}$ and $F(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x})$. Note that the presence of local constraints $\mathbf{X}_i$ and nonconvex objectives f_i and g makes the problem nonconvex and nonsmooth, which represents the major challenge to develop distributed algorithm with convergence guarantee.

Problem (P) is directly originated from smart buildings where smart devices are empowered to make local decisions while accounting for the interactions or the shared resource limits with the other devices in the proximity (see, for examples Yang, Hu, & Spanos, 2020; Yang, Srinivasan, Hu and Spanos, 2021a). Many other applications also fit into this formulation, including but not limited to smart sensing (Ansere, Han, Liu, Peng, & Kamal, 2020), energy storage sharing (Yang, Hu and Spanos, 2021), electric vehicle charging management (Yang et al., 2018; Zhang, Kekatos, & Giannakis, 2016), power system control (Arpanahi, Golshan, & Siano, 2020), wireless communication control (Hashempour, Suratgar, & Afshar, 2021). When the number of nodes is large, centralized methods usually suffer bottlenecks from the heavy computation, data storing and communication (see Arpanahi et al. (2020), Hashempour et al. (2021) and Yang et al. (2020) and the references therein). Also, centralized methods may disrupt privacy as the complete information of all agents (e.g., the private local objectives) are required by a central computing agent. As a result, distributed algorithms are usually preferred for privacy, computing efficiency, small data storage, and scaling properties.

When problem (P) is convex, plentiful distributed solution methods are available. The methods can be distinguished by the presence of the composite objective component g and the number of decision blocks N. When g is null, we have the classic dual decomposition methods (Falsone, Margellos, Garatti, & Prandini, 2017; Necoara & Nedelcu, 2015), the well-known alternating direction method of multiplier (ADMM) for two decision blocks ($N = 2$) (Boyd, Parikh, & Chu, 2011) and the variations for multi-block settings ($N \geq 3$) (Bai, Li, Xu, & Zhang, 2018; Cai, Han, & Yuan, 2017; Lin, Ma, & Zhang, 2015). While the classic ADMM and its variations propose to update the decision components in a sequential manner (usually called *Gauss–Seidel* decomposition), the works Chatzipanagiotis and Zavlanos (2017) and Deng et al. (2017) have made some effort in developing parallel ADMM and its variations (usually called *Jacobian ADMM* or *parallel ADMM*). The above methods are generally limited to separable objective functions (i.e., only f_i exist and $g = 0$). For the general case with composite objective component g, linearized ADMM (Aybat, Wang, Lin, & Ma, 2017) and inexact linearized ADMM (Bai, Hager, & Zhang, 2022) are also studied.

The above results are all for convex problems. Nevertheless, massive applications arising from the engineering systems and machine learning domains require to handle the type of problem (P) with possibly nonconvex objectives f_i and g. The nonconvexity may originate from the complex system performance metrics or the penalties imposed on the operation constraints. When the objectives f_i and g lack convexity (i.e., the monotonically non-decreasing property of gradients or subgradients is lost), developing distributed methods with theoretical convergence guarantee becomes a much more challenging problem. Though some fresh distributed methods for constrained nonconvex problems have been developed, they cannot be applied to problem (P) due to the nonsmoothness caused by the local constraints $\mathbf{X}_i$. This can be perceived from the following literature.

The existing works for constrained nonconvex optimization can be distinguished by `problem structures`, `main assumptions`, `decomposition scheme` (i.e., *Jacobian* or *Gauss–Seidel*) and `convergence guarantee` as reported in Table 1. Overall, they can be uniformly expressed by the template of problem (P) but are slightly different in the settings and assumptions.

The first category (Type 1) is concerned with problem (P) without any composite objective component g (Chatzipanagiotis & Zavlanos, 2017). An accelerated distributed augmented Lagrangian (ADAL) method was proposed to handle the possibly nonconvex but continuously differentiable objectives f_i. This method follows the classic ADMM framework but introduces an interpolation procedure regarding the primal updates at each iteration, which reads as $\mathbf{A}_i\mathbf{x}_i^{k+1} = \mathbf{A}_i\mathbf{x}_i^k + \mathbf{T}\left(\mathbf{A}_i\hat{\mathbf{x}}_i^k - \mathbf{A}_i\mathbf{x}_i^k\right)$ (k the iteration and $\mathbf{T}$ is a weighted matrix). To our understanding, this can be interpreted as a means to slow down the primal update for enhancing the convergence in nonconvex settings. By assuming the existence of stationary points that satisfy the strong second-order optimality condition, this paper established the local convergence of the method. The notion of local convergence is that the convergence towards some local optima can be assured if starting with a point sufficiently close to that local optima.

The subsequent four categories (Type 2, 3, 4, 5) differ from the first one mainly in the presence of a last block encoded by $\mathbf{B}$. Note that Hong et al. (2016) can be viewed as a special case with $\mathbf{B} = \mathbf{I}$, where $\mathbf{I}$ are identity matrices of suitable sizes. The last block is exceptional due to the unconstrained and Lipschitz differentiable property, which are critical to bound the dual updates for establishing convergence (see the references therein). That is why the last decision block is usually distinguished by some special notations (i.e., $\mathbf{y}$, $\mathbf{x}_0$). While the first category employs *Jacobian* decomposition for primal update, these four categories fall into *Gauss–Seidel* decomposition (i.e., alternating minimization). Specially, the works Liu et al. (2019) and Wang et al. (2019) have made some effort in handling possible composite objective components g but via different ways. Specifically, Wang et al. (2019) employed block coordinate and Liu et al. (2019) used linearization technique. Particularly, Hong et al. (2016) and Wang et al. (2019) build a general framework to establish the convergence for *Gauss–Seidel* ADMM towards local optima or stationary points in nonconvex settings, which comprises two key steps: (1) identifying a so-called sufficiently decreasing Lyapunov function, and (2) establishing the lower boundness property of the Lyapunov function. The sufficiently decreasing and lower boundness property of a proper Lyapunov function state that (Wang et al., 2019)

$$\begin{aligned} &T(\mathbf{x}^{k+1}, \boldsymbol{\lambda}^{k+1}) - T(\mathbf{x}^k, \boldsymbol{\lambda}^k) \\ &\qquad \leq -a_{\mathbf{x}}\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 - a_{\boldsymbol{\lambda}}\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2. \\ &T(\mathbf{x}^k, \boldsymbol{\lambda}^k) > -\infty. \end{aligned} \tag{2}$$

where $T(\cdot,\cdot)$ is a general Lyapunov function, $\mathbf{x}$ and $\boldsymbol{\lambda}$ are primal and dual variables, $a_{\mathbf{x}}$ and $a_{\boldsymbol{\lambda}}$ are positive coefficients. The augmented Lagrangian (AL) function has been often used as the Lyapunov function in nonconvex settings (see Hong et al. (2016) and Wang et al. (2019) and the references therein). However, they depend on the following two necessary conditions on the last decision block encoded by $\mathbf{B}$ to bound the dual updates $\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2$ by the primal updates $\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2$ (Hong et al., 2016; Wang et al., 2019).

(a) $\mathbf{B}$ has full column rank and $\mathrm{Im}(\mathbf{A}) \subseteq \mathrm{Im}(\mathbf{B})$ ($\mathrm{Im}(\cdot)$ represents the image of a matrix).
(b) The last decision block is unconstrained and with Lipschitz differentiable objective.

Noted that the fourth and fifth category (Type 4, 5) originated from Hong et al. (2016) are a special case with $\mathbf{B} = \mathbf{I}$ and thus satisfy the necessary **condition** (a).

Following the line of works, the sixth category (Type 6) studied the extension of ADMM to non-linearly constrained nonconvex

Table 1
Distributed constrained nonconvex optimization.

# Type	Problem structures	Main assumptions	Methods	Scheme	Convergence	Papers
1	$\min_{(\mathbf{x}_i)_{i=1}^N} \sum_{i=1}^N f_i(\mathbf{x}_i)$ s.t. $\sum_{i=1}^N \mathbf{A}_i\mathbf{x}_i = \mathbf{b}$. $\mathbf{x}_i \in \mathbf{X}_i,\ i = 1, 2, \ldots, N$.	f_i continuously differentiable. Strong second-order optimality condition.	ADAL	Jacobian	Local convergence. Local optima.	Chatzipanagiotis and Zavlanos (2017)
2	$\min_{\mathbf{x}=(\mathbf{x}_i)_{i=0}^p, \mathbf{y}} g(\mathbf{x}) + \sum_{i=0}^p f_i(\mathbf{x}_i) + h(\mathbf{y})$ s.t. $\sum_{i=0}^p \mathbf{A}_i\mathbf{x}_i + \mathbf{B}\mathbf{y} = 0$.	g and h Lipschitz continuous gradient. f_i weakly convex. $\mathrm{Im}(\mathbf{A}) \subseteq \mathrm{Im}(\mathbf{B})$.	ADMM	Gauss–Seidel	Global convergence. Stationary points.	Guo, Han, and Wu (2017), Li and Pong (2015), Wang et al. (2019) and Yang, Pong, and Chen (2017)
3	$\min_{\mathbf{x}=(\mathbf{x}_i)_{i=1}^N, \mathbf{y}} g(\mathbf{x}, \mathbf{y}) + \sum_{i=1}^N f_i(\mathbf{x}_i) + h(\mathbf{y})$ s.t. $\sum_{i=1}^N \mathbf{A}_i\mathbf{x}_i + \mathbf{B}\mathbf{y} = 0$.	g and h Lipschitz continuous gradient. $\mathrm{Im}(\mathbf{A}) \subseteq \mathrm{Im}(\mathbf{B})$.	Linearized ADMM	Gauss–Seidel	Global convergence. Stationary points.	Liu, Shen, and Gu (2019)
4	$\min_{(\mathbf{x}_k)_{k=0}^K} \sum_{k=1}^K g_k(\mathbf{x}_k) + h(\mathbf{x}_0)$ s.t. $\mathbf{x}_k = \mathbf{x}_0$. $\mathbf{x}_0 \in \mathbf{X}$.	g Lipschitz continuous gradient. h convex.	Flexible ADMM	Gauss–Seidel	Global convergence. Stationary points.	Hong, Luo, and Razaviyayn (2016)
5	$\min_{(\mathbf{x}_k)_{k=0}^K} \sum_{k=1}^N g_x(\mathbf{x}_k) + \ell(\mathbf{x}_0)$ s.t. $\sum_{k=1}^K \mathbf{A}_k\mathbf{x}_k = \mathbf{x}_0$. $\mathbf{x}_k \in \mathbf{X}_k,\ k = 1, \ldots, N$.	ℓ Lipschitz continuous gradient. g nonconvex but smooth or convex but non-smooth.	Flexible ADMM	Gauss–Seidel	Global convergence. Stationary points.	Hong et al. (2016)
6	$\min_{(\mathbf{x}_i)_{i=1}^N, \bar{\mathbf{x}}} \sum_{i=1}^N f_i(\mathbf{x}_i)$ s.t. $\sum_{i=1}^N \mathbf{A}_i\mathbf{x}_i + \mathbf{B}\bar{\mathbf{x}} = 0$. $\mathbf{x}_i \in \mathbf{X}_i, h_i(\mathbf{x}_i) = 0,\ i = 1, \ldots, N$. $\bar{\mathbf{x}} \in \bar{\mathbf{X}}$.	f_i continuously differentiable. h_i non-linear (possibly nonconvex). $\mathbf{B}$ full column rank. $\mathbf{X}_i$ possibly nonconvex.	ALM + ADMM	Gauss–Seidel	Global convergence. Stationary points.	Sun and Sun (2019, 2021)
7	$\min_{(\mathbf{x}_i)_{i=1}^N} g(\mathbf{x}) + \sum_{i=1}^N f_i(\mathbf{x}_i)$ s.t. $\sum_{i=1}^N \mathbf{A}_i\mathbf{x}_i = \mathbf{b}$. $\mathbf{x}_i \in \mathbf{X}_i,\ i = 1, 2, \ldots, N$.	f_i and g Lipschitz continuous gradient.	Proximal ADMM	Jacobian	Global convergence. Approximate stationary points.	This paper

Note: the set $\mathbf{X}_i$ and $\bar{\mathbf{X}}$ are bounded convex sets.

problems (Sun & Sun, 2019, 2021). Since it is difficult (if not impossible) to directly handle the non-linear couplings by the AL framework, Sun and Sun (2019) proposed to first convert the non-linearly constrained problems to linearly constrained ones by introducing decision copies for interconnected agents. This yields linearly constrained nonconvex problems with local non-linear constraints. The work (Sun & Sun, 2019) argues that the direct extension of ADMM to the reformulated problem is not applicable for the two necessary conditions **condition** (a) and (b) cannot be satisfied simultaneously. To bypass the challenge, Sun and Sun (2019) proposed to introduce a block of slack variables working as the last block. To force the slack block to *zero*, this paper developed a two-level method where the inner-level uses classic ADMM to solve a relaxed problem with a penalty on the slack variables, and the outer-level gradually forces the slack variables towards *zero*.

As can be perceived from the literature, it is difficult (if not impossible) to develop a distributed method with convergence guarantee for (**P**) due to the lack of a well-behaved last block satisfying **condition** (a) and (b). The work (Chatzipanagiotis & Zavlanos, 2017) provided a solution with local convergence guarantee but cannot handle the probable composite objective components g. Though the idea of introducing slack variables in Sun and Sun (2019) can provide a solution with global convergence guarantee but at the cost of heavy iteration complexity caused by the two-level structure. Despite these limitations, what we can learn from the literature is that the behaviors of dual variables is important to draw the convergence of ADMM for nonconvex problems.

This paper focuses on developing a distributed method for problem (**P**) with theoretical convergence guarantee. Our main contributions are

- We propose a proximal ADMM by revising the dual update procedure of classic ADMM into a discounted manner. This leads to the boundness of dual updates, which is critical to establish the convergence.
- We establish the global convergence of the method towards approximate stationary points by identifying a proper Lyapunov function which is sufficiently decreasing and lower bounded as required.
- We showcase the performance of the distributed method with a numerical example and a concrete application arising from smart buildings, which demonstrate the method's effectiveness.

The remainder of this paper is organized as follows. In Section 2, we present the proximal ADMM. In Section 3, we study the convergence of the method. In Section 4, we showcase the method's performance with a numerical example and smart building application. In Section 5, we conclude this paper and discuss the future work.

2. Proximal ADMM

2.1. Notations

Throughout the paper, we will visit the following notations. We use the bold alphabets $\mathbf{x}$, $\mathbf{y}$, $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$ and $\mathbf{A}$, $\mathbf{A}_i$, $\mathbf{Q}$, $\mathbf{M}$ to represent vectors and matrices. We define $\mathbf{I}_n$ or $\mathbf{I}$ as identity matrices of $n \times n$ or suitable size. We use the operator $:=$ to give definitions. We have $\mathbf{R}^n$ represent the n-dimensional real space and $(\mathbf{x}_i)_{i=1}^N := (\mathbf{x}_1^\top, \mathbf{x}_2^\top, \ldots, \mathbf{x}_N^\top)^\top$ is the stack of sub-vector $\mathbf{x}_i \in \mathbf{R}^{n_i}$. We refer to $\|\cdot\|$ as Euclidean norm without specification, i.e., $\|\mathbf{x}\| = \sqrt{\sum_{i=1}^n x_i^2}$ for $\mathbf{x} \in \mathbf{R}^n$, and $\langle \mathbf{x}, \mathbf{y}\rangle$ denote the dot product of vector $\mathbf{x}, \mathbf{y} \in \mathbf{R}^n$. We besides have $\|\mathbf{x}\|_{\mathbf{A}}^2 = \mathbf{x}^\top \mathbf{A}\mathbf{x}$. We use $\mathrm{diag}(\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_N)$ to denote the diagonal matrix formed by the sub-matrices $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_N$. We have the normal cone to a convex set $\mathbf{X} \subseteq \mathbf{R}^n$ at $\mathbf{x}^*$ defined by $N_{\mathbf{X}}(\mathbf{x}^*) := \{\nu \in \mathbf{R}^n | \langle \nu, \mathbf{x} - \mathbf{x}^*\rangle \le 0, \forall \mathbf{x} \in \mathbf{X}\}$. For $g : \mathbf{R}^n \to \mathbf{R}$ and $\mathbf{x} = (\mathbf{x}_i)_{i=1}^N \in \mathbf{R}^n$, we denote $\nabla_i g(\mathbf{x}) = \nabla_{\mathbf{x}_i} g(\mathbf{x})$ as the partial differential of g with respect to component $\mathbf{x}_i \in \mathbf{R}^{n_i}$. We define $\mathrm{dist}(\mathbf{x}, \mathbf{X}) = \min_{\mathbf{y}\in\mathbf{X}} \|\mathbf{x} - \mathbf{y}\|$ as the distance of vector $\mathbf{x} \in \mathbf{R}^n$ to the subset $\mathbf{X} \subseteq \mathbf{R}^n$.

2.2. Algorithm

In this part, we introduce the proximal ADMM for solving problem (P) in a distributed manner. The proximal ADMM is a type of AL methods that depend on the AL technique to relax constraints and employ the primal–dual scheme to update variables. By defining Lagrangian multipliers $\boldsymbol{\lambda} \in \mathbf{R}^m$ for the coupled constraints (1a), we have the AL function for problem (P)

$$\mathbb{L}_\rho(\mathbf{x}, \boldsymbol{\lambda}) = F(\mathbf{x}) + \langle \boldsymbol{\lambda}, \mathbf{A}\mathbf{x} - \mathbf{b}\rangle + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 \tag{3}$$

where $F(\mathbf{x}) = g(\mathbf{x}) + \sum_{i=1}^N f_i(\mathbf{x}_i)$ and ρ is the penalty parameter.

Following the standard AL methods, the proximal ADMM is composed of `Primal update` and `Dual update` as shown in Algorithm 1. In `Primal update`, the primal variables $\mathbf{x} = (\mathbf{x}_i)_{i=1}^N$ are updated in a distributed manner via *Jacobian* decomposition. Particularly, to handle the composite objective component g, we linearize the composite term at each iteration k by $g(\mathbf{x}^k) + \langle \nabla g(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k\rangle$ (the constant part $g(\mathbf{x}^k)$ is dropped). Note that the local objective terms f_i can also be linearized similarly if necessary and the proof of this paper still applies. To favor computation efficiency and scaling properties, we adopt the *Jacobian* scheme and empower the agents to update their decision components in parallel at each iteration with the preceding information from their interconnected agents. Particularly, to enhance convergence, a proximal term $\|\mathbf{x}_i - \mathbf{x}_i^{k+1}\|^2$ is imposed on the local objective of each agent (Step 3). This has been used in many *Jacobian* ADMM both in convex (Chang, Hong, & Wang, 2014; Deng et al., 2017; Li, Feng, & Xie, 2020) and nonconvex (Liu et al., 2019; Lu, Lee, Razaviyayn, & Hong, 2021) settings. Note that the subproblems (6) are either convex or nonconvex optimization over the local constraints $\mathbf{X}_i$, depending on f_i. There are many first-order solvers to solve those subproblems, such as the projected gradient method (Jain & Kar, 2017) and the proximal gradient method (Li & Lin, 2015). This paper focuses on developing a general distribute framework for solving problem (P) and will not discuss the subproblems in detail. The major difference of the proximal ADMM from the existing distributed AL methods is that we have modified the `Dual update` by imposing a discounting factor $(1-\tau)$ $(\tau \in [0, 1))$ (Step 4). The idea and motivation behind are to update the dual variables by the constraints residual in a discounted manner so as to bound the dual variables in the iterative process, which has been identified as critical to draw theoretical convergence. In this setting, the dual variables are the *discounted* running sum of the constraints residual, i.e.,

$$\begin{aligned}\boldsymbol{\lambda}^{k+1} &= (1-\tau)\boldsymbol{\lambda}^k + \rho(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b})\\ &= (1-\tau)^2\boldsymbol{\lambda}^{k-1} + (1-\tau)\rho(\mathbf{A}\mathbf{x}^k - \mathbf{b})\\ &\qquad + \rho(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b})\\ &\cdots\\ &= (1-\tau)^{k+1}\boldsymbol{\lambda}^0 + \sum_{\ell=0}^k (1-\tau)^{k-\ell}\rho(\mathbf{A}\mathbf{x}^{\ell+1} - \mathbf{b}).\end{aligned} \tag{4}$$

This differs from classic ADMM where the dual variables are the running sum of the constraints residual, i.e.,

$$\begin{aligned}\boldsymbol{\lambda}^{k+1} &= \boldsymbol{\lambda}^k + \rho(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b})\\ &= \boldsymbol{\lambda}^{k-1} + \rho(\mathbf{A}\mathbf{x}^k - \mathbf{b}) + \rho(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b})\\ &\cdots\\ &= \boldsymbol{\lambda}^0 + \sum_{\ell=0}^k \rho(\mathbf{A}\mathbf{x}^{\ell+1} - \mathbf{b}).\end{aligned}$$

From this perspective, classic ADMM can be viewed as a special case of the proximal ADMM with $\tau = 0$. In the proximal ADMM, the `Primal update` and `Dual update` are alternated until the stopping criterion

$$\|T_c^{k+1} - T_c^k\| \le \epsilon \tag{5}$$

is reached, where T_c^k is the Lyapunov function to be discussed later. The parameter ϵ is a user-defined positive threshold.

Algorithm 1 Proximal ADMM for problem (P)

1: **Initialize:** $\mathbf{x}^0$, $\boldsymbol{\lambda}^0$ and $\rho > 0$, $\tau \in [0, 1)$, and set $k \to 0$.
2: **Repeat:**
3: `Primal update`:

$$\mathbf{x}_i^{k+1} = \mathrm{argmin}_{\mathbf{x}_i\in\mathbf{X}_i}\left\{\begin{array}{l}\langle \nabla_i g(\mathbf{x}^k), \mathbf{x}_i - \mathbf{x}_i^k\rangle\\ +f_i(\mathbf{x}_i) + \langle \boldsymbol{\lambda}^k, \mathbf{A}_i\mathbf{x}_i^k\rangle\\ +\rho/2\|\mathbf{A}_i\mathbf{x}_i + \sum_{j\neq i}\mathbf{A}_j\mathbf{x}_j^k - \mathbf{b}\|^2\\ +\beta/2\|\mathbf{x}_i - \mathbf{x}_i^k\|_{\mathbf{B}_i}^2\end{array}\right\} \tag{6}$$

4: `Dual update`:

$$\boldsymbol{\lambda}^{k+1} = (1-\tau)\boldsymbol{\lambda}^k + \rho\left(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}\right) \tag{7}$$

5: Until the stopping criterion (5) is reached.

3. Convergence analysis

Before establishing the convergence of Algorithm 1, we first clarify the main assumptions.

3.1. Main assumptions

(A1) Function $f : \mathbf{R}^n \to \mathbf{R}$ and $g : \mathbf{R}^n \to \mathbf{R}$ have Lipschitz continuous gradient (i.e., Lipschitz differentiable) with modulus L_f and L_g over the set $\mathbf{X} = \mathbf{X}_1 \times \mathbf{X}_2 \times \cdots \times \mathbf{X}_N$, i.e., (Guo et al., 2017)

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \le L_f \|\mathbf{x} - \mathbf{y}\|, \ \ \forall \mathbf{x}, \mathbf{y} \in \mathbf{X}.$$
$$\|\nabla g(\mathbf{x}) - \nabla g(\mathbf{y})\| \le L_g \|\mathbf{x} - \mathbf{y}\|, \ \ \forall \mathbf{x}, \mathbf{y} \in \mathbf{X}.$$

(A2) Function $f : \mathbf{R}^n \to \mathbf{R}$ and $g : \mathbf{R}^n \to \mathbf{R}$ are lower bounded over the set $\mathbf{X} = \mathbf{X}_1 \times \mathbf{X}_2 \times \cdots \times \mathbf{X}_N$, i.e.,

$$f(\mathbf{x}) > -\infty, \quad \forall \mathbf{x} \in \mathbf{X}.$$
$$g(\mathbf{x}) > -\infty, \quad \forall \mathbf{x} \in \mathbf{X}.$$

3.2. Main results

As discussed, there are two key steps to draw convergence for a distributed AL method in nonconvex settings: (1) identifying a so-called sufficiently decreasing Lyapunov function; and (2) establishing the lower boundness property of the Lyapunov function. To achieve the objective, we first draw the following two propositions.

Proposition 1. *For the sequences $\{\mathbf{x}^k\}_{k\in\mathbf{K}}$ and $\{\boldsymbol{\lambda}^k\}_{k\in\mathbf{K}}$ generated by Algorithm 1, we have*

$$\begin{aligned}
&\frac{1-2\tau^2}{2\rho}\left\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\right\|^2+\frac{1}{2}\|\mathbf{x}^{k+1}-\mathbf{x}^k\|_{\mathbf{Q}}^2\\
&\quad+\frac{L_g}{2}\|\mathbf{x}^{k+1}-\mathbf{x}^k\|^2+\frac{1}{2}\|\mathbf{w}^k\|_{\mathbf{Q}}^2\\
&\le\frac{1-2\tau^2}{2\rho}\left\|\boldsymbol{\lambda}^{k}-\boldsymbol{\lambda}^{k-1}\right\|^2+\frac{1}{2}\|\mathbf{x}^{k}-\mathbf{x}^{k-1}\|_{\mathbf{Q}}^2\\
&\quad+\frac{L_g}{2}\|\mathbf{x}^{k}-\mathbf{x}^{k-1}\|^2+\rho_F\left\|\mathbf{x}^{k+1}-\mathbf{x}^k\right\|^2\\
&\quad-\tau(1+\tau)/\rho\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\|^2.
\end{aligned}$$

where we have the iterations $\mathbf{K} := \{1, 2, \ldots, K\}$ and

$$\begin{aligned}
\mathbf{w}^k &:= (\mathbf{x}^{k+1}-\mathbf{x}^k)-(\mathbf{x}^k-\mathbf{x}^{k-1})\\
G_{\mathbf{A}} &:= diag\left(\mathbf{A}_1^\top\mathbf{A}_1,\ldots,\mathbf{A}_N^\top\mathbf{A}_N\right)\\
G_{\mathbf{B}} &:= diag\left(\mathbf{B}_1^\top\mathbf{B}_1,\ldots,\mathbf{B}_N^\top\mathbf{B}_N\right)\\
\mathbf{Q} &:= \rho G_{\mathbf{A}}+\beta G_{\mathbf{B}}-\rho\mathbf{A}^\top\mathbf{A}\\
\rho_F &:= L_f+L_g.
\end{aligned}$$

Proof of Proposition 1. We defer the proof to Appendix A.

Let $\mathbb{L}_\rho^+(\mathbf{x},\boldsymbol{\lambda}) := \mathbb{L}_\rho(\mathbf{x},\boldsymbol{\lambda}) - \frac{\tau}{2\rho}\|\boldsymbol{\lambda}\|^2$ be the regularized AL function. We have the subsequent proposition to quantify the change of regularized AL function over the successive iterations.

Proposition 2. *For the sequences $\{\mathbf{x}^k\}_{k\in\mathbf{K}}$ and $\{\boldsymbol{\lambda}^k\}_{k\in\mathbf{K}}$ generated by Algorithm 1, we have*

$$\begin{aligned}
&\mathbb{L}_\rho^+(\mathbf{x}^{k+1},\boldsymbol{\lambda}^{k+1})-\mathbb{L}_\rho^+(\mathbf{x}^k,\boldsymbol{\lambda}^k)\\
&\le-\|\mathbf{x}^{k+1}-\mathbf{x}^k\|_{\mathbf{Q}}^2+\frac{\rho_F}{2}\|\mathbf{x}^{k+1}-\mathbf{x}^k\|^2\\
&\quad-\frac{\rho}{2}\|\mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^k)\|^2+\frac{2-\tau}{2\rho}\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\|^2.
\end{aligned}$$

Proof of Proposition 2. We defer the proof to Appendix B.

In the literature, the AL function is often used as the Lyapunov function if the sufficiently decreasing property can be established (see Guo et al. (2017), Li and Pong (2015), Wang et al. (2019) and Yang et al. (2017) for examples). However, this is not the case for Algorithm 1. From Proposition 2, we note that the sufficiently decreasing property of the (regularized) AL function can be established if and only if the dual updates $\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\|^2$ can be bounded by the primal updates $\|\mathbf{x}^{k+1}-\mathbf{x}^k\|^2$ (see the definition (2)). This is difficult (if not impossible) due to the lack of a well-behaved last block (i.e., unconstrained and Lipschitz differentiable) as discussed.

However, by combining Propositions 1 and 2, we indeed can identify a sufficiently decreasing Lyapunov function. Specifically, from Proposition 2, we have the (regularized) AL function $\mathbb{L}_\rho^+(\mathbf{x}^{k+1},\boldsymbol{\lambda}^{k+1})$ is ascending in $\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\|^2$ and descending in $\|\mathbf{x}^{k+1}-\mathbf{x}^k\|^2$. This is exactly opposite to the descending and ascending properties of the term $\frac{1-2\tau^2}{2\rho}\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\|^2+\frac{1}{2}\|\mathbf{x}^{k+1}-\mathbf{x}^k\|_{\mathbf{Q}}^2$ stated in Proposition 1. Note that this is attributed to the imposed discounted factor $\tau > 0$, otherwise the term $\tau(1-\tau)/\rho\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\|^2$ in Proposition 1 would be *zero*. We therefore build the Lyapunov function as

$$\begin{aligned}
T_c(\mathbf{x}^{k+1},\boldsymbol{\lambda}^{k+1};\mathbf{x}^k,\boldsymbol{\lambda}^k) &= \mathbb{L}_\rho^+(\mathbf{x}^{k+1},\boldsymbol{\lambda}^{k+1})\\
&\quad+c\Bigg(\frac{1-2\tau^2}{2\rho}\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\|^2+\frac{1}{2}\|\mathbf{x}^{k+1}-\mathbf{x}^k\|_{\mathbf{Q}}^2\\
&\quad+\frac{L_g}{2}\|\mathbf{x}^k-\mathbf{x}^{k-1}\|^2\Bigg) \qquad (8)
\end{aligned}$$

where c is a constant parameter to be determined for ensuring the sufficiently decreasing and lower boundness property of the Lyapunov function.

Let $T_c^{k+1} := T_c(\mathbf{x}^{k+1},\boldsymbol{\lambda}^{k+1};\mathbf{x}^k,\boldsymbol{\lambda}^k)$ be the Lyapunov function at iteration k, we have the following proposition regarding the sufficiently decreasing property.

Proposition 3. *For the sequences $\{\mathbf{x}^k\}_{k\in\mathbf{K}}$ and $\{\boldsymbol{\lambda}^k\}_{k\in\mathbf{K}}$ generated by Algorithm 1, we have*

$$T_c^{k+1}-T_c^k \le -a_{\mathbf{x}}\|\mathbf{x}^{k+1}-\mathbf{x}^k\|^2-a_{\boldsymbol{\lambda}}\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\|^2-\frac{c}{2}\|\mathbf{w}^k\|^2$$

where we have $\rho_F = L_f + L_g$ and

$$\begin{aligned}
a_{\mathbf{x}} &:= \frac{2\rho G_{\mathbf{A}}+2\beta G_{\mathbf{B}}-\rho\mathbf{A}^\top\mathbf{A}-(2c+1)\rho_F\mathbf{I}_N}{2}\\
a_{\boldsymbol{\lambda}} &:= \frac{2c\tau(1+\tau)-(2-\tau)}{2\rho}.
\end{aligned}$$

Proof of Proposition 3. Based on Propositions 1 and 2, we have

$$\begin{aligned}
&T_c^{k+1}-T_c^k = -\|\mathbf{x}^{k+1}-\mathbf{x}^k\|_{\mathbf{Q}}^2+\frac{\rho_F}{2}\|\mathbf{x}^{k+1}-\mathbf{x}^k\|^2\\
&\quad-\frac{\rho}{2}\|\mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^k)\|^2+\frac{2-\tau}{2\rho}\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\|^2\\
&\quad+c\Big(\rho_F\left\|\mathbf{x}^{k+1}-\mathbf{x}^k\right\|^2-\tau(1+\tau)/\rho\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\|^2\\
&\quad-1/2\|\mathbf{w}^k\|_{\mathbf{Q}}^2\Big)\\
&\le-a_{\mathbf{x}}\|\mathbf{x}^{k+1}-\mathbf{x}^k\|^2-a_{\boldsymbol{\lambda}}\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\|^2-\frac{c}{2}\|\mathbf{w}^k\|_{\mathbf{Q}}^2
\end{aligned}$$

where the inequality is directly derived by rearranging the terms. We therefore close the proof.

Remark 1. Proposition 3 implies that we would have the sufficiently decreasing property hold by the constructed Lyapunov function T_c^k if we have $a_{\mathbf{x}} > 0$, $a_{\boldsymbol{\lambda}} > 0$, $c \ge 0$ and $\mathbf{Q} \ge 0$. Actually, this can be achieved by setting the tuple $(\tau, \rho, \beta, \mathbf{B}_i, c)$ properly for Algorithm 1, which will be discussed shortly.

As discussed, another key step to draw the convergence is to establish the lower boundness property of the Lyapunov function.

To this end, we first prove the lower boundness property of Lagrangian multipliers resulting from the discounted dual update scheme.

Proposition 4. *Let $\Delta^k := \|\mathbf{A}\mathbf{x}^k - \mathbf{b}\|$ be the constraints residual at iteration k, $\Delta^{\max} := \max_{\mathbf{x}\in\mathbf{X}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|$ denote the maximal constraints residual over the closed feasible set $\mathbf{X}$, and Algorithm 1 start with any given initial dual variable $\boldsymbol{\lambda}^0$, we have $\|\boldsymbol{\lambda}^k\|$ is bounded, i.e.,*

$$\begin{aligned} &\|\boldsymbol{\lambda}^k\| \le \|\boldsymbol{\lambda}^0\| + \tau^{-1}\rho\Delta^{\max} \\ \text{or}\ \ &\|\boldsymbol{\lambda}^k\|^2 \le 2\|\boldsymbol{\lambda}^0\|^2 + 2\tau^{-2}\rho^2(\Delta^{\max})^2. \end{aligned} \tag{9}$$

Proof of Proposition 4. Recall the dual update scheme in (4), we have

$$\begin{aligned} \|\boldsymbol{\lambda}^k\| &= \|(1-\tau)^{k+1}\boldsymbol{\lambda}^0 + \sum_{\ell=0}^{k}\rho(1-\tau)^{k-\ell}\Delta^{\ell+1}\| \\ &\le \|(1-\tau)^{k+1}\boldsymbol{\lambda}^0\| + \sum_{\ell=0}^{k}\|\rho(1-\tau)^{k-\ell}\Delta^{\ell+1}\| \\ &\le \|(1-\tau)^{k+1}\boldsymbol{\lambda}^0\| + \rho\Delta^{\max}\frac{1-(1-\tau)^k}{\tau} \\ &\le \|\boldsymbol{\lambda}^0\| + \tau^{-1}\rho\Delta^{\max} \end{aligned}$$

where the first inequality is by the triangle inequality of norm, the second inequality infers from $\Delta^k \le \Delta^{\max}, \forall k$, and the last inequality holds because of $\tau \in (0,1)$.

Further based on $\|\mathbf{a}+\mathbf{b}\|^2 \le 2\|\mathbf{a}\|^2 + 2\|\mathbf{b}\|^2$, we directly have $\|\boldsymbol{\lambda}^k\|^2 \le 2\|\boldsymbol{\lambda}^0\|^2 + 2\tau^{-2}\rho^2(\Delta^{\max})^2$. We therefore complete the proof.

Based on Proposition 4, we are able to establish the lower boundness property of Lyapunov function as below.

Proposition 5. *For the sequences $\{\mathbf{x}^k\}_{k\in\mathbf{K}}$ and $\{\boldsymbol{\lambda}^k\}_{k\in\mathbf{K}}$ generated by Algorithm 1, we have*

$$T_c^{k+1} > -\infty, \forall k \in \mathbf{K}. \tag{10}$$

Proof of Proposition 5. By examining the terms of T_c^{k+1} in (8), we only require to establish the lower boundness property of $\mathbb{L}_\rho^+(\mathbf{x}^{k+1}, \boldsymbol{\lambda}^{k+1}) = \mathbb{L}_\rho(\mathbf{x}^{k+1}, \boldsymbol{\lambda}^{k+1}) - \frac{\tau}{2\rho}\|\boldsymbol{\lambda}^{k+1}\|^2$ for the other terms are all non-negative. Based on Proposition 4, we directly have $-\frac{\tau}{2\rho}\|\boldsymbol{\lambda}^{k+1}\|^2$ lower bounded since $\|\boldsymbol{\lambda}^{k+1}\|^2$ is upper bounded. We therefore only need to prove that $\mathbb{L}_\rho(\mathbf{x}^{k+1}, \boldsymbol{\lambda}^{k+1}) = f(\mathbf{x}^{k+1}) + \langle\boldsymbol{\lambda}^{k+1}, \mathbf{A}\mathbf{x}^{k+1} - b\rangle + \rho/2\left\|\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}\right\|^2$ is lower bounded. Note that we have $f(\mathbf{x}^{k+1}) > -\infty$ over the compact set $\mathbf{X}$ (see A2) and the quadratic term non-negative. This infers we only need to prove the lower boundness for the second term $\langle\boldsymbol{\lambda}^{k+1}, \mathbf{A}\mathbf{x}^{k+1} - b\rangle$. Based on the dual update (7), we have

$$\begin{aligned} \langle\boldsymbol{\lambda}^{k+1}, \mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}\rangle &= \left\langle\boldsymbol{\lambda}^{k+1}, \frac{\boldsymbol{\lambda}^{k+1} - (1-\tau)\boldsymbol{\lambda}^k}{\rho}\right\rangle \\ &= \left\langle\boldsymbol{\lambda}^{k+1}, \frac{1-\tau}{\rho}(\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k) + \frac{\tau}{\rho}\boldsymbol{\lambda}^{k+1}\right\rangle \\ &= \frac{\tau}{\rho}\|\boldsymbol{\lambda}^{k+1}\|^2 + \frac{1-\tau}{\rho}\langle\boldsymbol{\lambda}^{k+1}, \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\rangle \\ &= \frac{\tau}{\rho}\|\boldsymbol{\lambda}^{k+1}\|^2 + \frac{1-\tau}{2\rho}\left(\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2 + \|\boldsymbol{\lambda}^{k+1}\|^2 - \|\boldsymbol{\lambda}^k\|^2\right) \end{aligned} \tag{11}$$

Since we have $\|\boldsymbol{\lambda}^k\|^2$ is upper bounded (see Proposition 4), we therefore have $\mathbb{L}_\rho(\mathbf{x}^{k+1}, \boldsymbol{\lambda}^{k+1})$ lower bounded for the other terms of (11) are all non-negative. We thus complete the proof.

To present the main results regarding the convergence of Algorithm 1, we first give the definition on **Approximate stationary solution**.

Definition 1 (*Approximate Stationary Solution*)**.** For any given ϵ, we say a tuple $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ is an ϵ-stationary solution of problem (P), if we have

$$\operatorname{dist}\left(\nabla F(\mathbf{x}^*) + \mathbf{A}^\top\boldsymbol{\lambda}^* + N_{\mathbf{X}}(\mathbf{x}^*), \mathbf{0}\right) + \|\mathbf{A}\mathbf{x}^* - \mathbf{b}\| \le \epsilon.$$

where $\nabla F(\mathbf{x}^*) = \nabla f(\mathbf{x}^*) + \nabla g(\mathbf{x}^*)$.

In terms of the convergence of Algorithm 1 for problem (P), we have the following main results.

Theorem 1. *For Algorithm 1 with the tuple $(\tau, \rho, \beta, \mathbf{B}_i, c)$ selected by*

$$\begin{aligned} &\tau : \tau \in (0,1) \\ &c : c > \frac{2-\tau}{2\tau(1+\tau)} \\ &(\rho, \beta, \mathbf{B}_i) : \begin{cases} 2\rho G_{\mathbf{A}} + 2\beta G_{\mathbf{B}} - \rho\mathbf{A}^\top\mathbf{A} \ge (2c+1)\rho_F\mathbf{I}_N \\ \mathbf{Q} := \rho G_{\mathbf{A}} + \beta G_{\mathbf{B}} - \rho\mathbf{A}^\top\mathbf{A} \ge 0 \end{cases} \end{aligned} \tag{C1}$$

(a) The generated sequence $\{\mathbf{x}^k\}_{k\in\mathbf{K}}$ and $\{\boldsymbol{\lambda}^k\}_{k\in\mathbf{K}}$ are bounded and convergent, i.e.,

$$\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k \to 0, \quad \mathbf{x}^{k+1} - \mathbf{x}^k \to 0.$$

(b) Suppose we have the limit tuple $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$, then $(\mathbf{x}^*, \hat{\boldsymbol{\lambda}}^*)$ with $\hat{\boldsymbol{\lambda}}^* = (1+\tau\boldsymbol{\lambda}^*)$ is $\tau\rho^{-1}\|\boldsymbol{\lambda}^*\|$-stationary solution of problem (P).

Proof of Theorem 1. (a) Recall Proposition 3, we have

$$\begin{aligned} \sum_{k=1}^{K}\left(T_c^k - T_c^{k+1}\right) \ge\ & a_{\mathbf{x}}\sum_{k=1}^{K}\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 \\ &+ a_{\boldsymbol{\lambda}}\sum_{k=1}^{K}\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2 + \frac{c}{2}\sum_{k=1}^{K}\|\mathbf{w}^k\|^2 \end{aligned}$$

By assuming $K \to \infty$, we have

$$\begin{aligned} T_c^1 - \lim_{K\to\infty} T_c^{k+1} \ge\ & a_{\mathbf{x}}\sum_{k=1}^{\infty}\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 \\ &+ a_{\boldsymbol{\lambda}}\sum_{k=1}^{\infty}\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2 + \frac{c}{2}\sum_{k=1}^{\infty}\|\mathbf{w}^k\|^2 \end{aligned}$$

Since we have $T_c^{k+1} > -\infty$ (see Proposition 5), we thus have

$$\infty \ge a_{\mathbf{x}}\sum_{k=1}^{\infty}\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 + a_{\boldsymbol{\lambda}}\sum_{k=1}^{\infty}\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2 + \frac{c}{2}\sum_{k=1}^{\infty}\|\mathbf{w}^k\|^2.$$

We therefore conclude

$$\begin{aligned} &\|\mathbf{x}^{k+1} - \mathbf{x}^k\| \to 0, \quad \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\| \to 0, \\ &\|\mathbf{w}^k\| = \|(\mathbf{x}^{k+1} - \mathbf{x}^k) - (\mathbf{x}^k - \mathbf{x}^{k-1})\| \to 0. \end{aligned}$$

(b) According to (a), we have the sequences $\{\mathbf{x}^k\}_{k\in\mathbf{K}}$ and $\{\boldsymbol{\lambda}^k\}_{k\in\mathbf{K}}$ converge to some limit tuple $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$, i.e., if $k \to \infty$, we have $\mathbf{x}^{k+1} \to \mathbf{x}^*$, $\boldsymbol{\lambda}^{k+1} \to \boldsymbol{\lambda}^*$ and $\mathbf{x}^{k+1} \to \mathbf{x}^k$ and $\boldsymbol{\lambda}^{k+1} \to \boldsymbol{\lambda}^k$.

Based on the dual update procedure (7), we have the stationary tuple $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ satisfy

$$\mathbf{A}\mathbf{x}^* - \mathbf{b} = \tau\rho^{-1}\boldsymbol{\lambda}^*. \tag{12}$$

Since we have $\hat{\boldsymbol{\lambda}}^k = \boldsymbol{\lambda}^k + \rho(\mathbf{A}\mathbf{x}^k - \mathbf{b})$, we thus have $\hat{\boldsymbol{\lambda}}^k \to (1+\tau)\boldsymbol{\lambda}^*$. Let $\hat{\boldsymbol{\lambda}}^* = (1+\tau)\boldsymbol{\lambda}^*$, we have $\hat{\boldsymbol{\lambda}}^k \to \hat{\boldsymbol{\lambda}}^*$.

Recall the first-order optimality condition (A.4) and assume $k \to \infty$ that the stationary point $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ is reached, we would have

$$\langle \nabla f(\mathbf{x}^*) + \nabla g(\mathbf{x}^*) + \mathbf{A}^\top \hat{\boldsymbol{\lambda}}^*, \mathbf{x}^* - \mathbf{x} \rangle \leq 0, \forall \mathbf{x} \in \mathbf{X}.$$

This implies that

$$\mathbf{0} \in \nabla f(\mathbf{x}^*) + \nabla g(\mathbf{x}^*) + \mathbf{A}^\top \hat{\boldsymbol{\lambda}}^* + N_{\mathbf{X}}(\mathbf{x}^*).$$

We further have

$$\operatorname{dist}\big(\nabla f(\mathbf{x}^*) + \nabla g(\mathbf{x}^*) + \mathbf{A}^\top \hat{\boldsymbol{\lambda}}^* + N_{\mathbf{X}}(\mathbf{x}^*), 0\big) = 0 \tag{13}$$

By combining (12) and (13), we therefore conclude

$$\operatorname{dist}\big(\nabla f(\mathbf{x}^*) + \nabla g(\mathbf{x}^*) + \mathbf{A}^\top \hat{\boldsymbol{\lambda}}^* + N_{\mathbf{X}}(\mathbf{x}^*), 0\big) + \|\mathbf{A}\mathbf{x}^* - \mathbf{b}\| \leq \tau \rho^{-1} \|\boldsymbol{\lambda}^*\|,$$

which closes the proof.

From Theorem 1, we note that if the convergent $\boldsymbol{\lambda}^*$ does not depend on τ and ρ, we could decrease τ or increase ρ to achieve any sub-optimality. If that is not the case, we give the following corollary to show that this still can be achieved by properly setting the initial point and parameters.

Corollary 1. *For any given $\epsilon > 0$, if Algorithm 1 starts with $\boldsymbol{\lambda}^0 = 0$ and $\mathbf{A}\mathbf{x}^0 = \mathbf{b}$, $\tau \in (0, 1)$, and the penalty parameter ρ is selected that*

$$\rho \geq \epsilon^{-2}\tau\big(4 + c(1 - 2\tau^2) + c/2\big)d_F + \epsilon^{-2} cL_g/2 \|\mathbf{x}^0\|^2 + \epsilon^{-2}\tau cL_g/2 \|\mathbf{x}^0\|^2 + \epsilon^{-2}\tau c\rho_F/4\, d_{\mathbf{x}},$$

we have the limit tuples $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ and $(\mathbf{x}^*, \hat{\boldsymbol{\lambda}}^*)$ with $\hat{\boldsymbol{\lambda}}^* = (1 + \tau)\boldsymbol{\lambda}^*$ is ϵ-stationary solution of problem (P). where we have $d_F = \max_{\mathbf{x}\in\mathbf{X}} f(\mathbf{x}) + g(\mathbf{x})$, $d_{\mathbf{x}} = \max_{\mathbf{x},\mathbf{y}\in\mathbf{X}} \|\mathbf{x} - \mathbf{y}\|^2$, and we assume $f(\mathbf{x}) \geq 0$, $g(\mathbf{x}) \geq 0$ without losing any generality.

Proof of Corollary 1. We only give the sketch of the proof and defer the details to Appendix C. The proof is structured by two parts which include: (i) proving $\|\boldsymbol{\lambda}^*\|^2 \leq \rho\tau^{-1}T_c^0$, and (ii) proving $T_c^0 \leq \big(4 + c(1 - 2\tau^2) + c/2\big)d_F + cL_g/2\|\mathbf{x}^0\|^2 + c\rho_F/4\, d_{\mathbf{x}}$. Based on (i) and (ii), we have $\tau^2\rho^{-2}\|\boldsymbol{\lambda}^*\|^2 \leq \epsilon^2$. We then directly draw the conclusion based on Theorem 1.

4. Numerical experiments

4.1. A numerical example

We first consider a numerical example with $N = 2$ agents given by

$$\begin{aligned} \min_{x_1, x_2} \quad & 0.1x_1^3 + 0.1x_2^3 + 0.1x_1x_2 \qquad \textbf{(P1)} \\ \text{s.t.} \quad & x_1 + x_2 = 1 \\ & -1 \leq x_1 \leq 1 \\ & -1 \leq x_2 \leq 1 \end{aligned}$$

For this example, we have $f_1(x_1) = 0.1x_1^3$, $f_2(x_2) = 0.1x_2^3$, and $g(x_1, x_2) = 0.1x_1x_2$. The Lipschitz continuous gradient modulus for f and g are $L_f = 0.6$ and $L_g = 0.2$. Besides, we have $\mathbf{A}_1 = 1$, $\mathbf{A}_2 = 1$, $\mathbf{A} = (1\ 1)$. The stationary point of the problem is $x_1^\star = 0.5, x_2^\star = 0.5$.

To our best knowledge, there is no distributed solution methods for solving problem (P1) with theoretical convergence guarantee. In the following, we apply the proximal ADMM to solve this problem and verify the solution quality. We consider four different parameter settings for Algorithm 1:

(S1) $\tau = 0.1$, $\rho = 10$, $\beta = 10$, $c = 8.7$
(S2) $\tau = 0.1$, $\rho = 20$, $\beta = 20$, $c = 8.7$
(S3) $\tau = 0.05$, $\rho = 5$, $\beta = 16$, $c = 18.6$
(S4) $\tau = 0.05$, $\rho = 10$, $\beta = 16$, $c = 18.6$

The other parameters are set as $\mathbf{B}_1 = \mathbf{B}_2 = 1$, $\tau = 0.1$, $x_1^0 = 0.2$, $x_2^0 = 0.8$, $\lambda^0 = 0$ and kept the same for S1–S4. Note that we have $\tau/\rho = 0.01$ for S1/S3 and $\tau/\rho = 0.005$ for S2/S4. We make such settings for comparisons as we have the suboptimality of the method related to the ratio τ/ρ as stated in Theorem 1. We therefore study how the ratio τ/ρ will affect the convergence rate and the solution quality of the method.

Before running the algorithm, we first can easily verify the convergence condition (C1) stated in Theorem 1 for S1–S4. We use the `interior-point` method embedded in the `fmincon` solver of MATLAB to solve subproblems (6). We run Algorithm 1 sufficiently long (i.e., $K = 2000$ iterations when the Lyapunov function does not change apparently) for the settings S1–S4. We first examine the convergence of the method indicated by the Lyapunov function. Fig. 1(a) shows the evolution of the Lyapunov function w.r.t. the iterations with S1–S4. We observe that for all the settings S1–S4, the Lyapunov functions strictly decrease w.r.t. the iterations and finally stabilize at some value that is close to the optima $f^\star = 0.1x_1^\star + 0.1x_2^\star + 0.1x_1^\star x_2^\star = 0.05$. By further examining the results, we note that a larger ratio τ/ρ yields faster convergence rate as with S1/S3 ($\tau/\rho = 0.01$) compared with S2/S4 ($\tau/\rho = 0.005$). This is caused by the relatively smaller penalty factor ρ and proximal factor β required to ensure the convergence condition (C1) for a larger τ/ρ. Note that the penalty factor ρ and the proximal factor β can be interpreted as some means to slow down the primal updates as they have an effect in penalizing the deviation from current update $\mathbf{x}^k$. Oppositely, a smaller ratio τ/ρ generally yields higher solution quality (i.e., smaller suboptimality gap) as with S2/S4 ($\tau/\rho = 0.005$) compared with S1/S3 ($\tau/\rho = 0.01$). This is in line with Theorem 1.

To further examine the solution quality, we report the detailed results with the four settings S1–S4 (`Prox-ADMM-Sx`, x = 1, 2, 3, 4) and the centralized method (using the `interior-point` method embedded in the `fmincon` solver of MATLAB) in Table 2. Note that the convergent solution $\hat{x}_1$ and $\hat{x}_2$ with proximal ADMM under the settings S1–S4 are quite close to the optimal solution $x_1^\star = 0.5$ and $x_2^\star = 0.5$ obtained from the centralized method. More specifically, by measuring the sub-optimality by $\|\hat{\mathbf{x}} - \mathbf{x}^\star\|/\|\mathbf{x}^\star\|$ where $\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2)$ and $\mathbf{x}^\star = (x_1^\star, x_2^\star)$ are the convergent and optimal solution, we have the sub-optimality of proximal ADMM is around 1.2E−3 with S1/S3 ($\tau/\rho = 0.01$) and 5.9E−4 with S2/S4 ($\tau/\rho = 0.005$). We therefore imply that a smaller ratio τ/ρ can achieve higher solution quality but generally at the cost of slower convergence rate as observed in Fig. 1(a). This implies that a trade-off in terms of the solution quality and the convergence speed is necessary while configuring the algorithm (i.e., the ratio of τ/ρ) for specific applications. For this example, considering both the solution quality and convergence rate, we have S4 a preferred option. We therefore display the convergence of primal variables x_1 and x_2 with S4 in Fig. 1(b). Note that x_1 and x_2 gradually approach the optimal solution $\mathbf{x}_1^\star = 0.5$ and $x_2^\star = 0.5$.

4.2. Application: multi-zone HVAC control

To showcase the performance of proximal ADMM in applications, we apply it to the multi-zone heating, ventilation, and air conditioning (HVAC) control arising from smart buildings. The goal is to optimize the HVAC operation to provide the comfortable temperature with minimal electricity bill. Due to the thermal capacity of buildings, the evolution of indoor temperature is a slow

Table 2
Performance of proximal ADMM under the settings S1–S4 vs. Centralized method.

Method	τ/ρ	$\hat{x}_1$	$\hat{x}_2$	Sub-optimality	Convergence rate
Centralized	–	0.5	0.5	–	–
Prox-ADMM-S1	0.01	0.4994	0.4994	1.1E−3	No. 1
Prox-ADMM-S2	0.005	0.4997	0.4997	5.7E−4	No. 4
Prox-ADMM-S3	0.01	0.4994	0.4994	1.2E−3	No .2
Prox-ADMM-S4	0.005	0.4997	0.4997	5.9E−4	No. 3

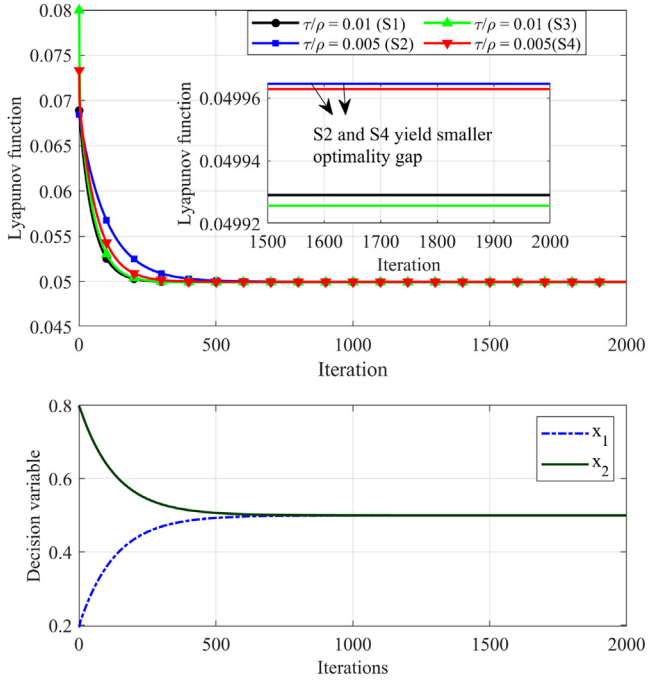

Fig. 1. (a) The evolution of Lyapunov function T_c^k with S1–S4. (b) The evolution of primal variables x_1 and x_2 with S4.

process affected both by the dynamic indoor occupancy (thermal loads) and the HVAC operation (cooling loads). The general solution is to design a model predictive controller for optimizing HVAC operation (i.e., zone mass flow and zone temperature trajectories) to minimize the overall electricity cost while respecting the comfortable temperature ranges based on the predicted information (i.e., indoor occupancy, outdoor temperature, electricity price, etc.). The general problem formulation is presented below.

$$\min_{\mathbf{m}^z,\mathbf{T}} \sum_t c_t \Big\{ c_p(1-d_r) \sum_i m_t^{zi}(T_t^o - T^c) + c_p \eta d_r \sum_i m_t^{zi}(T_t^i - T^c) + \kappa_f \Big(\sum_i m_t^{zi}\Big)^2 \Big\} \Delta_t \tag{P2}$$

$$\text{s.t. } T_{t+1}^i = A_{ii} T_t^i + \sum_{j\in N_i} A_{ij} T_t^j + C_{ii} m_t^{zi}(T_t^i - T^c) + D_t^i, \ \forall i, t. \tag{14a}$$

$$T_{\min}^i \le T_t^i \le T_{\max}^i, \quad \forall i, t. \tag{14b}$$

$$m_{\min}^{zi} \le m_t^{zi} \le m_{\max}^{zi}, \ \forall i, t. \tag{14c}$$

$$\sum_i m_t^{zi} \le \overline{m}, \ \forall t. \tag{14d}$$

where i and t are zone and time indices, $\mathbf{T} = (T_t^i)_{\forall i,t}$ and $\mathbf{m}^z = (m_t^{zi})_{\forall i,t}$ are zone temperature and the supplied zone mass flow

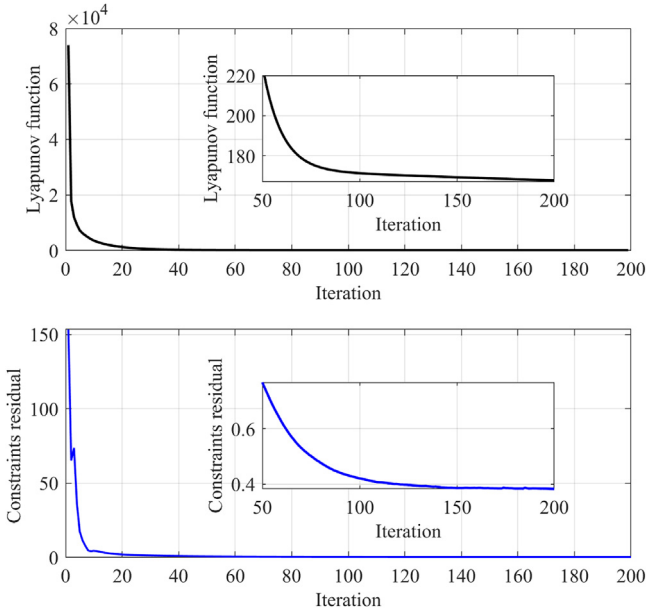

Fig. 2. (a) The evolution of Lyapunov function. (b) The evolution of the norm of constraints residual.

rates, which are decision variables. Note that we have the HVAC system serves N zones with thermal couplings (i.e., heat transfer) within a building. The other notations are parameters. For example, $[T_{\min}^i, T_{\max}^i]$ represent the comfortable temperature range of zone i. The problem is subject to the constraints including zone thermal couplings (14a), comfortable temperature margins (14b), zone mass flow rate limits (14c), and total zone mass flow rate limits (14d).

Note that problem (**P2**) is generally in large scale for a commercial building due to the large number of zones and rooms. This problem represents one of the major challenging problems with smart buildings. In this part, we show how the proximal ADMM can be applied to solve problem (**P2**) in a distributed manner and thus overcome the computation burden. Note that problem (**P2**) can be transformed to the form of (**P**). We refer the readers to our extended version (Yang, Jia, Xu, Guan, & Spanos, 2021b) for details (see problem **P**3). We consider a case study with $N = 10$ zones and the predicted horizon $T = 48$ time slots (a whole day with a sampling interval of 30 min). We set the lower and upper comfortable temperature bounds as $T_{\min}^i = 24$ °C and $T_{\max}^i = 26$ °C. The specifications for HVAC system can refer to Yang et al. (2020) and Yang, Srinivasan et al. (2021a). The algorithm of proximal ADMM is configured by $\rho = 2.0$, $\tau = 0.1$, $\beta = 3.0$, $\mathbf{B}_i = \mathbf{I}$ (suitable sizes), and $c = 8.7$. We run the algorithm suitably long ($K = 200$ iterations when both the residual and Lyapunov function do not change apparently). We first examine the convergence of the algorithm measured by the `Lyapunov function` and the norm of (coupled) `constraints residual`. We visualize the `Lyapunov function` and `constraints residual` in Fig. 2. Note that the `Lyapunov function` strictly declines along the iterations, which is consistent with our theoretical analysis. Besides, the `constraints residual` almost strictly decreases with the iterations as well and finally approaches *zero*. We have the overall norm of the `constraints residual` at the end of iterations is about 0.38, which is quite small considering the problem scale $T \cdot N = 480$. This justifies the convergence property of proximal ADMM for the smart building application.

We next evaluate the solution quality measured by the HVAC electricity cost and human comfort. We randomly pick 3 zones

Table 3
Prox-ADMM vs. Centralized for HVAC control in smart buildings ($N = 10$ zones).

Method	Electricity cost (s$)	Human comfort	Constraints residual	Computing time
`Centralized`	153.12	Y	0	$\geq$ 10 h
`Prox-ADMM`	160.54	Y	0.38	50 min

(zone 1, 3, 7) and display the predicted `zone occupancy` (inputs), the zone mass flow rates (`zone MFR`, control variables), and the zone temperature (`zone temp.`, control variables) over the 48 time slots in Fig. 3. Note that the variations of `zone MFR` are almost consistent with the `zone occupancy`. This is reasonable as the `zone occupancy` determines the thermal loads which need to be balanced by the zone mass flow rates. We besides see that the `zone temp.` are all maintained within the comfortable range [24, 26] °C. This infers the satisfaction of `human comfort`. To further evaluate the solution quality and computation efficiency, we compare the proximal ADMM (`Prox-ADMM`) with centralized method (`Centralized`). Specifically, we use the `interior-point` embedded in the `fmincon` solver of MATLAB to solve both the subproblems (6) with `Prox-ADMM` and problem (**P2**) with `Centralized`. For the `Centralized`, we run the solver sufficiently long without considering the time with the objective to approach the best possible optimal solution. We compare the two methods in three folds, i.e., `electricity cost`, the norm of `constraints residual`, and `computation time` as reported in Table 3. We see that `electricity cost` with `Prox-ADMM` is about 160.20 (s$) versus 153.12 (s$) yield by `Centralized`. We imply the sub-optimality of `Prox-ADMM` in terms of the objective is about 5.0%. However, the `Prox-ADMM` obviously outperforms the `Centralized` in computation efficiency. The average `computing time` for each zone is about 50 min with `Prox-ADMM` (parallel computation) while the `Centralized` takes more than 10 h. Note that we have picked $T = 48$ time slots (a whole day) as the predicted horizon, the computing time could be largely sharpened in practice with a much smaller prediction horizon, say $T = 10$ time slots (5 h).

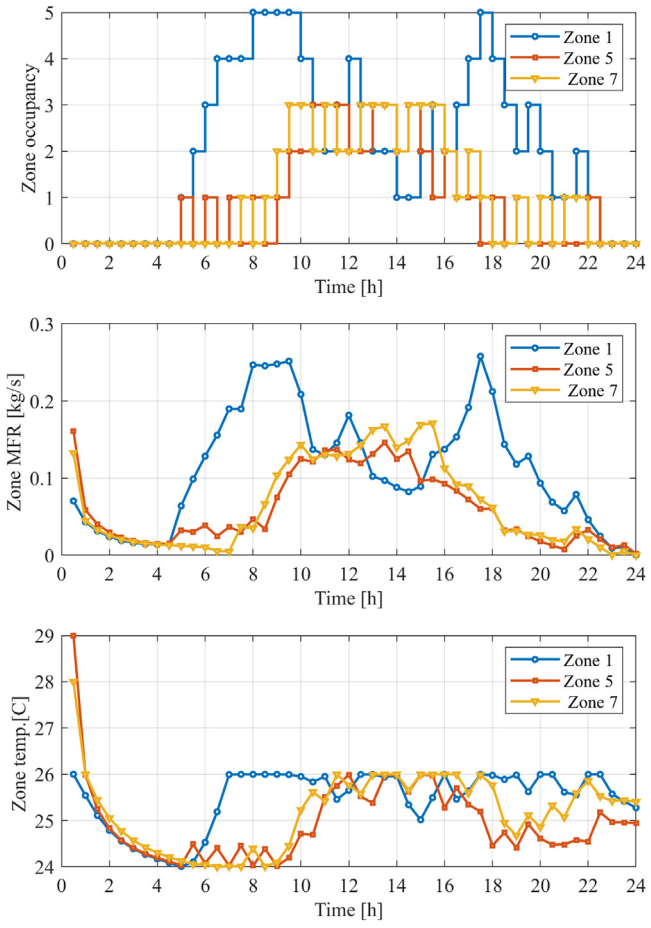

Fig. 3. (a) Zone occupancy. (b) Zone mass flow rate (Zone MFR). (c) Zone temperature (Zone temp.).

5. Conclusion and future work

This paper focused on developing a distributed algorithm for a class of nonconvex and nonsmooth problems with convergence guarantee. The problems are featured by (i) a possibly nonconvex objective composed of both separate and composite components, (ii) local bounded convex constraints, and (iii) global coupled linear constraints. This class of problems is broad in application but lacks distributed methods with convergence guarantee. We turned to the powerful alternating direction method of multiplier (ADMM) for constrained optimization but faced the challenge to establish convergence. Noting that the underlying obstacle is to assume the boundness of dual updates, we revised the classic ADMM and proposed to update the dual variables in a distributed manner. This leads to a proximal ADMM with the convergence guarantee towards the approximate stationary points of the problem. We demonstrated the convergence and solution quality of the distributed method by a numerical example and a concrete application to the multi-zone heating, ventilation, and air-condition (HVAC) control arising from smart buildings.

This paper proposed the discounted dual update scheme in conjunction with ADMM for a class of nonconvex and nonsmooth problems, some interesting future work includes studying whether the discounted dual update scheme can be explored to develop distributed methods for more broad classes of problems both in convex and nonconvex settings.

Acknowledgments

This work is supported by the Republic of Singapore's National Research Foundation through a grant to the Berkeley Education Alliance for Research in Singapore (BEARS) for the Singapore-Berkeley Building Efficiency and Sustainability in the Tropics (SinBerBEST) Program. This work is also supported in part by National Natural Science Foundation of China (62192752, 62192750, 62125304, 62073182), 111 International Collaboration Project (BP2018006), and Tsinghua University Initiative Scientific Research Program.

Appendix A. Proof of Proposition 1

Proposition 1 is established based on the first-order optimality condition of subproblems (6) and the Lipschitz continuous gradient property of f and g.

We first establish the following equality and notation.

$$\begin{aligned}&\mathbf{A}_i\mathbf{x}_i^{k+1} + \sum_{j\neq i}\mathbf{A}_j\mathbf{x}_j^k - \mathbf{b} \qquad (A.1)\\ &= \mathbf{A}\mathbf{x}^k - \mathbf{b} + \mathbf{A}_i(\mathbf{x}_i^{k+1} - \mathbf{x}_i^k).\\ &= \mathbf{A}\mathbf{x}^{k+1} - \mathbf{b} + \mathbf{A}(\mathbf{x}^k - \mathbf{x}^{k+1}) + \mathbf{A}_i(\mathbf{x}_i^{k+1} - \mathbf{x}_i^k).\end{aligned}$$

$$\hat{\boldsymbol{\lambda}}^k := \boldsymbol{\lambda}^k + \rho(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}). \qquad (A.2)$$

For subproblems (6), the first-order optimality condition states that there exists $\nu_i^{k+1} \in N_{\mathbf{X}_i}(\mathbf{x}_i^{k+1})$ that

$$\begin{aligned}
0 = \ & \nabla f_i(\mathbf{x}_i^{k+1}) + \nabla_i g(\mathbf{x}^k) + \mathbf{A}_i^\top \boldsymbol{\lambda}^k \\
& + \rho \mathbf{A}_i^\top (\mathbf{A}_i \mathbf{x}_i^{k+1} + \sum_{j \neq i} \mathbf{A}_j \mathbf{x}_j^k - \mathbf{b}) \\
& + \beta \mathbf{B}_i^\top \mathbf{B}_i (\mathbf{x}_i^{k+1} - \mathbf{x}_i^k) + \nu_i^{k+1} \\
= \ & \nabla f_i(\mathbf{x}_i^{k+1}) + \nabla_i g(\mathbf{x}^k) + \mathbf{A}_i^\top \big(\boldsymbol{\lambda}^k + \rho(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b})\big) \\
& + \rho \mathbf{A}_i^\top \mathbf{A}(\mathbf{x}^k - \mathbf{x}^{k+1}) + \rho \mathbf{A}_i^\top \mathbf{A}_i (\mathbf{x}_i^{k+1} - \mathbf{x}_i^k) \\
& + \beta \mathbf{B}_i^\top \mathbf{B}_i (\mathbf{x}_i^{k+1} - \mathbf{x}_i^k) + \nu_i^{k+1} \qquad \text{by (A.1)} \\
= \ & \nabla f_i(\mathbf{x}_i^{k+1}) + \nabla_i g(\mathbf{x}^k) + \mathbf{A}_i^\top \hat{\boldsymbol{\lambda}}^k + \rho \mathbf{A}_i^\top \mathbf{A}(\mathbf{x}^k - \mathbf{x}^{k+1}) \\
& + \rho \mathbf{A}_i^\top \mathbf{A}_i (\mathbf{x}_i^{k+1} - \mathbf{x}_i^k) \\
& + \beta \mathbf{B}_i^\top \mathbf{B}_i \big(\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\big) + \nu_i^{k+1} \qquad \text{by (A.2).}
\end{aligned}$$

Multiplying by $(\mathbf{x}_i^{k+1} - \mathbf{x}_i)$ in both sides, we have

$$\begin{aligned}
& \langle \nabla f_i(\mathbf{x}_i^{k+1}), \mathbf{x}_i^{k+1} - \mathbf{x}_i \rangle + \langle \nabla_i g(\mathbf{x}^k), \mathbf{x}_i^{k+1} - \mathbf{x}_i \rangle \\
& + \langle \hat{\boldsymbol{\lambda}}^k, \mathbf{A}_i(\mathbf{x}_i^{k+1} - \mathbf{x}_i) \rangle \\
& + \rho \langle \mathbf{A}(\mathbf{x}^k - \mathbf{x}^{k+1}), \mathbf{A}_i(\mathbf{x}_i^{k+1} - \mathbf{x}_i) \rangle \\
& + \rho \langle \mathbf{A}_i(\mathbf{x}_i^{k+1} - \mathbf{x}_i^k), \mathbf{A}_i(\mathbf{x}_i^{k+1} - \mathbf{x}_i) \rangle \\
& + \beta \langle \mathbf{B}_i(\mathbf{x}_i^{k+1} - \mathbf{x}_i^k), \mathbf{B}_i(\mathbf{x}_i^{k+1} - \mathbf{x}_i) \rangle \\
= & - \langle \nu_i^{k+1}, \mathbf{x}_i^{k+1} - \mathbf{x}_i \rangle \leq 0, \quad \forall \mathbf{x}_i \in \mathbf{X}_i.
\end{aligned} \tag{A.3}$$

Summing up (A.3) over i, we have $\forall \mathbf{x}_i \in \mathbf{X}_i$,

$$\begin{aligned}
& \langle \nabla f(\mathbf{x}^{k+1}), \mathbf{x}^{k+1} - \mathbf{x} \rangle + \langle \nabla g(\mathbf{x}^k), \mathbf{x}^{k+1} - \mathbf{x} \rangle \\
& + \langle \hat{\boldsymbol{\lambda}}^k, \mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}) \rangle + (\mathbf{x}^{k+1} - \mathbf{x}) \rho \mathbf{A}^\top \mathbf{A}(\mathbf{x}^k - \mathbf{x}^{k+1}) \\
& + \sum_i (\mathbf{x}_i^{k+1} - \mathbf{x}_i)^\top (\rho \mathbf{A}_i^\top \mathbf{A}_i + \beta \mathbf{B}_i^\top \mathbf{B}_i)(\mathbf{x}_i^{k+1} - \mathbf{x}_i^k) \leq 0.
\end{aligned}$$

Plugging in $\mathbf{Q} := \rho G_{\mathbf{A}} + \beta G_{\mathbf{B}} - \rho \mathbf{A}^\top \mathbf{A}$, we have

$$\begin{aligned}
& \langle \nabla f(\mathbf{x}^{k+1}), \mathbf{x}^{k+1} - \mathbf{x} \rangle + \langle \nabla g(\mathbf{x}^k), \mathbf{x}^{k+1} - \mathbf{x} \rangle \\
& + \langle \hat{\boldsymbol{\lambda}}^k, \mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}) \rangle \\
& + (\mathbf{x}^{k+1} - \mathbf{x})^\top \mathbf{Q}(\mathbf{x}^{k+1} - \mathbf{x}^k) \leq 0, \quad \forall \mathbf{x} \in \mathbf{X}.
\end{aligned} \tag{A.4}$$

By induction, we have

$$\begin{aligned}
& \langle \nabla f(\mathbf{x}^k), \mathbf{x}^k - \mathbf{x} \rangle + \langle \nabla g(\mathbf{x}^{k-1}), \mathbf{x}^k - \mathbf{x} \rangle \\
& + \langle \hat{\boldsymbol{\lambda}}^{k-1}, \mathbf{A}(\mathbf{x}^k - \mathbf{x}) \rangle \\
& + (\mathbf{x}^k - \mathbf{x})^\top \mathbf{Q}(\mathbf{x}^k - \mathbf{x}^{k-1}) \leq 0, \quad \forall \mathbf{x} \in \mathbf{X}.
\end{aligned} \tag{A.5}$$

By setting $\mathbf{x} = \mathbf{x}^k$ and $\mathbf{x} = \mathbf{x}^{k+1}$ with (A.4) and (A.5), we have

$$\begin{aligned}
& \langle \nabla f(\mathbf{x}^{k+1}), \mathbf{x}^{k+1} - \mathbf{x}^k \rangle + \langle \nabla g(\mathbf{x}^k), \mathbf{x}^{k+1} - \mathbf{x}^k \rangle \\
& + \langle \hat{\boldsymbol{\lambda}}^k, \mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k) \rangle \\
& + (\mathbf{x}^{k+1} - \mathbf{x}^k)^\top \mathbf{Q}(\mathbf{x}^{k+1} - \mathbf{x}^k) \leq 0.
\end{aligned} \tag{A.6}$$

$$\begin{aligned}
& \langle \nabla f(\mathbf{x}^k), \mathbf{x}^k - \mathbf{x}^{k+1} \rangle + \langle \nabla g(\mathbf{x}^{k-1}), \mathbf{x}^k - \mathbf{x}^{k+1} \rangle \\
& + \langle \hat{\boldsymbol{\lambda}}^{k-1}, \mathbf{A}(\mathbf{x}^k - \mathbf{x}^{k+1}) \rangle \\
& + (\mathbf{x}^k - \mathbf{x}^{k+1})^\top \mathbf{Q}(\mathbf{x}^k - \mathbf{x}^{k-1}) \leq 0.
\end{aligned} \tag{A.7}$$

Summing up (A.6) and (A.7) and plugging in $\mathbf{w}^k := (\mathbf{x}^{k+1} - \mathbf{x}^k) - (\mathbf{x}^k - \mathbf{x}^{k-1})$, we have

$$\begin{aligned}
& \langle \nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k), \mathbf{x}^{k+1} - \mathbf{x}^k \rangle \\
& + \langle \nabla g(\mathbf{x}^k) - \nabla g(\mathbf{x}^{k-1}), \mathbf{x}^{k+1} - \mathbf{x}^k \rangle \\
& + \langle \hat{\boldsymbol{\lambda}}^k - \hat{\boldsymbol{\lambda}}^{k-1}, \mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k) \rangle \\
& + (\mathbf{x}^{k+1} - \mathbf{x}^k)^\top \mathbf{Q} \mathbf{w}^k \leq 0.
\end{aligned} \tag{A.8}$$

Based on the Lipschitz continuous gradient property of f over the compact set $\mathbf{x} \in \mathbf{X}$, we have

$$\langle \nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k), \mathbf{x}^{k+1} - \mathbf{x}^k \rangle \geq -L_f \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2. \tag{A.9}$$

We also have

$$\begin{aligned}
& \langle \nabla g(\mathbf{x}^k) - \nabla g(\mathbf{x}^{k-1}), \mathbf{x}^{k+1} - \mathbf{x}^k \rangle \\
& = \langle \frac{\nabla g(\mathbf{x}^k) - \nabla g(\mathbf{x}^{k-1})}{\sqrt{L_g}}, \sqrt{L_g}(\mathbf{x}^{k+1} - \mathbf{x}^k) \rangle \\
& \geq -\frac{1}{2L_g} \|\nabla g(\mathbf{x}^k) - \nabla g(\mathbf{x}^{k-1})\|^2 - \frac{L_g}{2} \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 \\
& \geq -\frac{L_g}{2} \|\mathbf{x}^k - \mathbf{x}^{k-1}\|^2 - \frac{L_g}{2} \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2
\end{aligned}$$

where the last equality is based on the Lipschitz continuous gradient property of g.

Besides, we have

$$\begin{aligned}
& \langle \hat{\boldsymbol{\lambda}}^k - \hat{\boldsymbol{\lambda}}^{k-1}, \mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k) \rangle \\
& = \big\langle \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k + \tau(\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}), \mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k) \big\rangle \\
& = \Big\langle \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k + \tau(\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}), \\
& \qquad \frac{\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k}{\rho} - \frac{(1-\tau)}{\rho}(\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}) \Big\rangle \\
& = \frac{\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2}{\rho} - \frac{(1-2\tau)}{\rho} \langle \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k, \boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1} \rangle \\
& \quad - \frac{\tau(1-\tau)}{\rho} \|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}\|^2 \\
& \geq \frac{\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2}{\rho} - \frac{1-2\tau}{2\rho} \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2 \\
& \quad - \frac{1-2\tau}{2\rho} \|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}\|^2 - \frac{\tau(1-\tau)}{\rho} \|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}\|^2 \\
& = \frac{1-2\tau^2}{2\rho} \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2 - \frac{1-2\tau^2}{2\rho} \|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}\|^2 \\
& \quad + \tau(\tau+1)/\rho \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2
\end{aligned} \tag{A.10}$$

where the inequality is by $\langle \mathbf{a}, \mathbf{b} \rangle \leq \frac{1}{2}(\|\mathbf{a}\|^2 + \|\mathbf{b}\|^2)$.

Based on the inequality $\mathbf{b}^\top \mathbf{M}(\mathbf{b} - \mathbf{c}) = \frac{1}{2}(\|\mathbf{b} - \mathbf{c}\|_{\mathbf{M}}^2 + \|\mathbf{b}\|_{\mathbf{M}}^2 - \|\mathbf{c}\|_{\mathbf{M}}^2)$, and by setting $\mathbf{M} = \mathbf{Q}$, $\mathbf{b} = \mathbf{x}^{k+1} - \mathbf{x}^k$, and $\mathbf{c} = \mathbf{x}^k - \mathbf{x}^{k-1}$, we have

$$\begin{aligned}
(\mathbf{x}^{k+1} - \mathbf{x}^k)^\top \mathbf{Q} \mathbf{w}^k = & \frac{1}{2}(\|\mathbf{w}^k\|_{\mathbf{Q}}^2 + \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{Q}}^2 \\
& - \|\mathbf{x}^k - \mathbf{x}^{k-1}\|_{\mathbf{Q}}^2).
\end{aligned} \tag{A.11}$$

Plugging (A.9), (A.10), (A.11) into (A.8), we have

$$\begin{aligned}
& \frac{1-2\tau^2}{2\rho} \big\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\big\|^2 + \frac{1}{2} \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{Q}}^2 \\
& + \frac{L_g}{2} \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 + \frac{1}{2} \|\mathbf{w}^k\|_{\mathbf{Q}}^2 \\
& \leq \frac{1-2\tau^2}{2\rho} \big\|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}\big\|^2 + \frac{1}{2} \|\mathbf{x}^k - \mathbf{x}^{k-1}\|_{\mathbf{Q}}^2 \\
& + \frac{L_g}{2} \|\mathbf{x}^k - \mathbf{x}^{k-1}\|^2 + (L_g + L_f) \big\|\mathbf{x}^{k+1} - \mathbf{x}^k\big\|^2 \\
& - \tau(1+\tau)/\rho \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2.
\end{aligned}$$

We therefore complete the proof.

Appendix B. Proof of Proposition 2

Before starting the proof, we first establish the following inequalities to be used. Based on the Lipschitz continuous gradient

property of $f:\mathbf{R}^n\to\mathbf{R}$ over $\mathbf{x}\in\mathbf{X}$ (see (A1)), we have (Guo et al., 2017)

$$f(\mathbf{x}^{k+1})-f(\mathbf{x}^k)\le\ \langle\nabla f(\mathbf{x}^{k+1}),\mathbf{x}^{k+1}-\mathbf{x}^k\rangle + L_f/2\|\mathbf{x}^k-\mathbf{x}^{k+1}\|^2. \tag{B.1}$$

Similarly, for $g:\mathbf{R}^n\to\mathbf{R}$ with Lipschitz continuous gradient over $\mathbf{x}\in\mathbf{X}$ (see (A1)), we have (Guo et al., 2017)

$$g(\mathbf{x}^{k+1})-g(\mathbf{x}^k)\le\ \langle\nabla g(\mathbf{x}^k),\mathbf{x}^{k+1}-\mathbf{x}^k\rangle + L_g/2\|\mathbf{x}^k-\mathbf{x}^{k+1}\|^2. \tag{B.2}$$

Besides, we have

$$\begin{aligned}&\frac{\rho}{2}\|\mathbf{A}\mathbf{x}^{k+1}-\mathbf{b}\|^2-\frac{\rho}{2}\|\mathbf{A}\mathbf{x}^k-\mathbf{b}\|^2\\ =\ &\frac{\rho}{2}\left\langle\mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^k),\mathbf{A}\mathbf{x}^{k+1}+\mathbf{A}\mathbf{x}^k-2\mathbf{b}\right\rangle\\ =\ &\frac{\rho}{2}\left\langle\mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^k),\mathbf{A}(\mathbf{x}^k-\mathbf{x}^{k+1})+2(\mathbf{A}\mathbf{x}^k-\mathbf{b})\right\rangle\\ =&-\frac{\rho}{2}\|\mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^k)\|^2+\left\langle\mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^k),\rho(\mathbf{A}\mathbf{x}^{k+1}-\mathbf{b})\right\rangle.\end{aligned} \tag{B.3}$$

We next quantify the decrease of $\mathbb{L}_\rho(\mathbf{x},\boldsymbol{\lambda})$ with respect to (w.r.t.) the primal updates. We have

$$\begin{aligned}&\mathbb{L}_\rho(\mathbf{x}^{k+1},\boldsymbol{\lambda}^k)-\mathbb{L}_\rho(\mathbf{x}^k,\boldsymbol{\lambda}^k)\\ =&f(\mathbf{x}^{k+1})-f(\mathbf{x}^k)+g(\mathbf{x}^{k+1})-g(\mathbf{x}^k)+\langle\boldsymbol{\lambda}^k,\mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^k)\rangle\\ &+\frac{\rho}{2}\|\mathbf{A}\mathbf{x}^{k+1}-\mathbf{b}\|^2-\frac{\rho}{2}\|\mathbf{A}\mathbf{x}^k-\mathbf{b}\|^2\\ \le&\langle\nabla f(\mathbf{x}^{k+1})+\nabla g(\mathbf{x}^k),\mathbf{x}^{k+1}-\mathbf{x}^k\rangle+\rho_F/2\|\mathbf{x}^k-\mathbf{x}^{k+1}\|^2\\ &+\left\langle\boldsymbol{\lambda}^k,\mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^k)\right\rangle-\frac{\rho}{2}\|\mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^k)\|^2\\ &+\left\langle\mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^k),\rho(\mathbf{A}\mathbf{x}^{k+1}-\mathbf{b})\right\rangle\ \text{by (B.1), (B.2), (B.3)}\\ =&\langle\nabla f(\mathbf{x}^{k+1})+\nabla g(\mathbf{x}^k)+\mathbf{A}^\top\hat{\boldsymbol{\lambda}}^k,\mathbf{x}^{k+1}-\mathbf{x}^k\rangle\\ &+\rho_F/2\|\mathbf{x}^k-\mathbf{x}^{k+1}\|^2-\rho/2\|\mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^k)\|^2\ \text{by (A.2)}\\ \le&-\|\mathbf{x}^{k+1}-\mathbf{x}^k\|_{\mathbf{Q}}^2+\rho_F/2\|\mathbf{x}^k-\mathbf{x}^{k+1}\|^2\\ &-\rho/2\ \|\mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^k)\|^2\ \text{by (A.6)}.\end{aligned} \tag{B.4}$$

We next quantify the change of $\mathbb{L}_\rho(\mathbf{x},\boldsymbol{\lambda})$ w.r.t. dual update. We have

$$\begin{aligned}&\mathbb{L}_\rho(\mathbf{x}^{k+1},\boldsymbol{\lambda}^{k+1})-\mathbb{L}_\rho(\mathbf{x}^{k+1},\boldsymbol{\lambda}^k)\\ =&\left\langle\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k,\mathbf{A}\mathbf{x}^{k+1}-\mathbf{b}\right\rangle\\ =&\left\langle\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k,\frac{\boldsymbol{\lambda}^{k+1}-(1-\tau)\boldsymbol{\lambda}^k}{\rho}\right\rangle\\ =&\left\langle\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k,\frac{1-\tau}{\rho}(\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k)+\frac{\tau}{\rho}\boldsymbol{\lambda}^{k+1}\right\rangle\\ =&\frac{(1-\tau)}{\rho}\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\|^2+\frac{\tau}{2\rho}\Big(\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\|^2\\ &+\|\boldsymbol{\lambda}^{k+1}\|^2-\|\boldsymbol{\lambda}^k\|^2\Big)\\ =&\frac{2-\tau}{2\rho}\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\|^2+\frac{\tau}{2\rho}\|\boldsymbol{\lambda}^{k+1}\|^2-\frac{\tau}{2\rho}\|\boldsymbol{\lambda}^k\|^2.\end{aligned} \tag{B.5}$$

Combining (B.4) and (B.5), we have

$$\begin{aligned}&\mathbb{L}_\rho(\mathbf{x}^{k+1},\boldsymbol{\lambda}^{k+1})-\frac{\tau}{2\rho}\|\boldsymbol{\lambda}^{k+1}\|^2-\Big(\mathbb{L}_\rho(\mathbf{x}^k,\boldsymbol{\lambda}^k)-\frac{\tau}{2\rho}\|\boldsymbol{\lambda}^k\|^2\Big)\\ \le&-\|\mathbf{x}^{k+1}-\mathbf{x}^k\|_{\mathbf{Q}}^2+\frac{\rho_F}{2}\|\mathbf{x}^k-\mathbf{x}^{k+1}\|^2\\ &-\frac{\rho}{2}\|\mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^k)\|^2+\frac{2-\tau}{2\rho}\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\|^2.\end{aligned}$$

We have $\mathbb{L}_\rho^+(\mathbf{x},\boldsymbol{\lambda})=\mathbb{L}_\rho(\mathbf{x},\boldsymbol{\lambda})-\frac{\tau}{2\rho}\|\boldsymbol{\lambda}\|^2$, we therefore close the proof.

Appendix C. Proof of Corollary 1

(i) Prove $\|\boldsymbol{\lambda}^*\|^2\le\rho\tau^{-1}T_c^0$: Based on the sufficiently decreasing property of T_c^{k+1} (see Proposition 3), we have

$$T_c^{k+1}\le T_c^0 \tag{C.1}$$

Recalling the definition of the Lyapunov function in (8) and invoking (11), we have

$$\begin{aligned}T_c^{k+1}=&f(\mathbf{x}^{k+1})+g(\mathbf{x}^{k+1})+\frac{\tau}{\rho}\|\boldsymbol{\lambda}^{k+1}\|^2\\ &+\frac{1-\tau}{2\rho}\big(\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\|^2+\|\boldsymbol{\lambda}^{k+1}\|^2-\|\boldsymbol{\lambda}^k\|^2\big)\\ &+\frac{\rho}{2}\|\mathbf{A}\mathbf{x}^{k+1}-\mathbf{b}\|^2+c\Big(\frac{1-2\tau^2}{2\rho}\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\|^2\\ &+1/2\|\mathbf{x}^{k+1}-\mathbf{x}^k\|_{\mathbf{Q}}^2+L_g/2\|\mathbf{x}^k-\mathbf{x}^{k-1}\|^2\Big)\end{aligned} \tag{C.2}$$

By combining (C.1) and (C.2), we have

$$\frac{1-\tau}{2\rho}\big(\|\boldsymbol{\lambda}^{k+1}\|^2-\|\boldsymbol{\lambda}^k\|^2\big)+\frac{\tau}{\rho}\|\boldsymbol{\lambda}^{k+1}\|^2\le T_c^0. \tag{C.3}$$

The above holds because we have $f(\mathbf{x})\ge 0$, $g(\mathbf{x})\ge 0$ over $\mathbf{X}$ and the other terms are all non-negative.

We next prove $\frac{\tau}{\rho}\|\boldsymbol{\lambda}^{k+1}\|^2\le T_c^0$ by induction. For $k=0$, we can properly pick the initial point to satisfy the inequality. For iteration k, we assume $\frac{\tau}{\rho}\|\boldsymbol{\lambda}^k\|^2\le T_c^0$. We consider the two possible cases for iteration $k+1$, i.e., if $\|\boldsymbol{\lambda}^{k+1}\|^2\le\|\boldsymbol{\lambda}^k\|^2$, we straightforwardly have $\frac{\tau}{\rho}\|\boldsymbol{\lambda}^{k+1}\|^2\le\frac{\tau}{\rho}\|\boldsymbol{\lambda}^k\|^2\le T_c^0$, and else if $\|\boldsymbol{\lambda}^{k+1}\|^2\ge\|\boldsymbol{\lambda}^k\|^2$, we also have $\frac{\tau}{\rho}\|\boldsymbol{\lambda}^{k+1}\|^2\le T_c^0$ by (C.3). We therefore conclude $\|\boldsymbol{\lambda}^*\|^2\le\rho\tau^{-1}T_c^0$.

(ii) Prove $T_c^0\le\big(4+c(1-2\tau^2)+c/2\big)d_F+cL_g/2\|\mathbf{x}^0\|^2+c\rho_F/4\,d_{\mathbf{x}}$: Invoke Proposition 2 and set $k=0$, we have

$$\begin{aligned}&\mathbb{L}_\rho(\mathbf{x}^1,\boldsymbol{\lambda}^1)-\frac{\tau}{2\rho}\|\boldsymbol{\lambda}^1\|^2\le\mathbb{L}_\rho(\mathbf{x}^0,\boldsymbol{\lambda}^0)-\frac{\tau}{2\rho}\|\boldsymbol{\lambda}^0\|^2\\ &-\|\mathbf{x}^1-\mathbf{x}^0\|_{\mathbf{Q}}^2+\frac{\rho_F}{2}\|\mathbf{x}^1-\mathbf{x}^0\|^2-\frac{\rho}{2}\|\mathbf{A}(\mathbf{x}^1-\mathbf{x}^0)\|^2\\ &+\frac{2-\tau}{2\rho}\|\boldsymbol{\lambda}^1-\boldsymbol{\lambda}^0\|^2.\end{aligned}$$

By invoking (11) and setting $\boldsymbol{\lambda}^{-1}=0$, $\boldsymbol{\lambda}^0=0$, $\mathbf{A}\mathbf{x}^0=b$, $\mathbf{Q}:=\rho G_{\mathbf{A}}+\beta G_{\mathbf{B}}-\rho\mathbf{A}^\top\mathbf{A}$, we have

$$\begin{aligned}&\frac{\rho}{2}\|\mathbf{A}\mathbf{x}^1-\mathbf{b}\|^2+\frac{2\mathbf{Q}+\rho\mathbf{A}^\top\mathbf{A}-\rho_f\mathbf{I}_N}{2}\|\mathbf{x}^1-\mathbf{x}^0\|^2\\ \le&f(\mathbf{x}^0)+g(\mathbf{x}^0)-f(\mathbf{x}^1)-g(\mathbf{x}^1)\end{aligned}$$

Since we have $f(\mathbf{x})\ge 0$ and $f(\mathbf{x})\ge 0$ over the set $\mathbf{X}$, we have (the term $\frac{\rho\mathbf{A}^\top\mathbf{A}}{2}\|\mathbf{x}^1-\mathbf{x}^0\|^2$ is non-negative)

$$\frac{\rho}{2}\|\mathbf{A}\mathbf{x}^1-\mathbf{b}\|^2\le d_F. \tag{C.4}$$

$$\frac{2\mathbf{Q}-\rho_F\mathbf{I}_N}{2}\|\mathbf{x}^1-\mathbf{x}^0\|^2\le d_F$$

$$\|\mathbf{x}^1-\mathbf{x}^0\|_{\mathbf{Q}}^2\le d_F+\rho_F/2d_{\mathbf{x}}. \tag{C.5}$$

where the last inequality is by $d_{\mathbf{x}}:=\max_{\mathbf{x},\mathbf{y}}\|\mathbf{x}-\mathbf{y}\|^2$.

Further, based on the dual update, we have

$$\frac{1}{2\rho}\|\boldsymbol{\lambda}^1\|^2=\frac{\rho}{2}\|\mathbf{A}\mathbf{x}^1-\mathbf{b}\|^2\le d_F \tag{C.6}$$

Further, we have

$$\begin{aligned}T_c^0=&f(\mathbf{x}^1)+g(\mathbf{x}^1)+\frac{2+c(1-2\tau^2)}{2\rho}\|\boldsymbol{\lambda}^1\|^2+\frac{\rho}{2}\|\mathbf{A}\mathbf{x}^1-\mathbf{b}\|^2\\ &+\frac{c}{2}\|\mathbf{x}^1-\mathbf{x}^0\|_{\mathbf{Q}}^2+\frac{cL_g}{2}\|\mathbf{x}^0\|^2\ \text{by (C.2) and }\boldsymbol{\lambda}^0=0\end{aligned}$$

$$\leq\ d_F + (2 + c(1-2\tau^2))d_F + d_F + \frac{c}{2}d_F + \frac{c\rho_F}{4}d_{\mathbf{x}} + \frac{cL_g}{2}\|\mathbf{x}^0\|^2 \quad \text{by (C.4), (C.5), (C.6)}$$
$$=\ \big(4 + c(1-2\tau^2) + c/2\big)d_F + cL_g/2\|\mathbf{x}^0\|^2 + c\rho_F/4\ d_{\mathbf{x}}$$

By combining (i) and (ii), we have $\tau^2\rho^{-2}\|\boldsymbol{\lambda}^*\|^2 \leq \epsilon^2$. By invoking Theorem 1, we directly have

$$\operatorname{dist}\big(\nabla f(\mathbf{x}^*) + \nabla g(\mathbf{x}^*) + \mathbf{A}^\top\hat{\boldsymbol{\lambda}}^* + N_{\mathbf{X}}(\mathbf{x}^*), 0\big) + \|\mathbf{A}\mathbf{x}^* - \mathbf{b}\| \leq \tau\rho^{-1}\|\boldsymbol{\lambda}^*\| \leq \epsilon,$$

which closes the proof.

References

Ansere, J. A., Han, G., Liu, L., Peng, Y., & Kamal, M. (2020). Optimal resource allocation in energy-efficient internet-of-things networks with imperfect CSI. *IEEE Internet of Things Journal*, *7*(6), 5401–5411.

Arpanahi, M. K., Golshan, M. H., & Siano, P. (2020). A comprehensive and efficient decentralized framework for coordinated multiperiod economic dispatch of transmission and distribution systems. *IEEE Systems Journal*.

Aybat, N. S., Wang, Z., Lin, T., & Ma, S. (2017). Distributed linearized alternating direction method of multipliers for composite convex consensus optimization. *IEEE Transactions on Automatic Control*, *63*(1), 5–20.

Bai, J., Hager, W. W., & Zhang, H. (2022). An inexact accelerated stochastic ADMM for separable convex optimization. *Computational Optimization and Applications*, 1–40.

Bai, J., Li, J., Xu, F., & Zhang, H. (2018). Generalized symmetric ADMM for separable convex optimization. *Computational Optimization and Applications*, *70*(1), 129–170.

Boyd, S., Parikh, N., & Chu, E. (2011). *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc.

Cai, X., Han, D., & Yuan, X. (2017). On the convergence of the direct extension of ADMM for three-block separable convex minimization models with one strongly convex function. *Computational Optimization and Applications*, *66*(1), 39–73.

Chang, T.-H., Hong, M., & Wang, X. (2014). Multi-agent distributed optimization via inexact consensus ADMM. *IEEE Transactions on Signal Processing*, *63*(2), 482–497.

Chatzipanagiotis, N., & Zavlanos, M. M. (2017). On the convergence of a distributed augmented lagrangian method for nonconvex optimization. *IEEE Transactions on Automatic Control*, *62*(9), 4405–4420.

Deng, W., Lai, M.-J., Peng, Z., & Yin, W. (2017). Parallel multi-block ADMM with o (1/k) convergence. *Journal of Scientific Computing*, *71*(2), 712–736.

Falsone, A., Margellos, K., Garatti, S., & Prandini, M. (2017). Dual decomposition for multi-agent distributed optimization with coupling constraints. *Automatica*, *84*, 149–158.

Falsone, A., Notarnicola, I., Notarstefano, G., & Prandini, M. (2020). Tracking-ADMM for distributed constraint-coupled optimization. *Automatica*, *117*, Article 108962.

Guo, K., Han, D., & Wu, T.-T. (2017). Convergence of alternating direction method for minimizing sum of two nonconvex functions with linear constraints. *International Journal of Computational Methods*, *94*(8), 1653–1669.

Hashempour, S., Suratgar, A. A., & Afshar, A. (2021). Distributed nonconvex optimization for energy efficiency in mobile ad hoc networks. *IEEE Systems Journal*.

Hong, M., Luo, Z.-Q., & Razaviyayn, M. (2016). Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization*, *26*(1), 337–364.

Houska, B., Frasch, J., & Diehl, M. (2016). An augmented lagrangian based algorithm for distributed nonconvex optimization. *SIAM Journal on Optimization*, *26*(2), 1101–1127.

Jain, P., & Kar, P. (2017). Non-convex optimization for machine learning.

Li, X., Feng, G., & Xie, L. (2020). Distributed proximal algorithms for multiagent optimization with coupled inequality constraints. *IEEE Transactions on Automatic Control*, *66*(3), 1223–1230.

Li, H., & Lin, Z. (2015). Accelerated proximal gradient methods for nonconvex programming. *Advances in Neural Information Processing Systems*, 28.

Li, G., & Pong, T. K. (2015). Global convergence of splitting methods for nonconvex composite optimization. *SIAM Journal on Optimization*, *25*(4), 2434–2460.

Lin, T.-Y., Ma, S.-Q., & Zhang, S.-Z. (2015). On the sublinear convergence rate of multi-block ADMM. *Journal of the Operations Research Society of China*, *3*(3), 251–274.

Liu, Q., Shen, X., & Gu, Y. (2019). Linearized ADMM for nonconvex nonsmooth optimization with convergence analysis. *IEEE Access*, *7*, 76131–76144.

Lu, S., Lee, J. D., Razaviyayn, M., & Hong, M. (2021). Linearized ADMM converges to second-order stationary points for non-convex problems. *IEEE Transactions on Signal Processing*, *69*, 4859–4874.

Necoara, I., & Nedelcu, V. (2015). On linear convergence of a distributed dual gradient algorithm for linearly constrained separable convex problems. *Automatica*, *55*, 209–216.

Shi, W., Ling, Q., Yuan, K., Wu, G., & Yin, W. (2014). On the linear convergence of the ADMM in decentralized consensus optimization. *IEEE Transactions on Signal Processing*, *62*(7), 1750–1761.

Sun, K., & Sun, X. A. (2019). A two-level distributed algorithm for general constrained non-convex optimization with global convergence. arXiv preprint arXiv:1902.07654.

Sun, K., & Sun, X. A. (2021). A two-level ADMM algorithm for AC OPF with convergence guarantees. *IEEE Transactions on Power Systems*.

Wang, Y., Yin, W., & Zeng, J. (2019). Global convergence of ADMM in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, *78*(1), 29–63.

Yang, Y., Hu, G., & Spanos, C. J. (2020). HVAC energy cost optimization for a multizone building via a decentralized approach. *IEEE Transactions on Automation Science and Engineering*, *17*(4), 1950–1960.

Yang, Y., Hu, G., & Spanos, C. J. (2021). Optimal sharing and fair cost allocation of community energy storage. *IEEE Transactions on Smart Grid*, *12*(5), 4185–4194.

Yang, Y., Jia, Q.-S., Guan, X., Zhang, X., Qiu, Z., & Deconinck, G. (2018). Decentralized ev-based charging optimization with building integrated wind energy. *IEEE Transactions on Automation Science and Engineering*, *16*(3), 1002–1017.

Yang, Y., Jia, Q.-S., Xu, Z., Guan, X., & Spanos, C. J. (2021b). Proximal ADMM for Nonconvex and Nonsmooth Optimization. https://arxiv.org/pdf/2205.01951.pdf.

Yang, L., Pong, T. K., & Chen, X. (2017). Alternating direction method of multipliers for a class of nonconvex and nonsmooth problems with applications to background/foreground extraction. *SIAM Journal on Imaging Sciences*, *10*(1), 74–110.

Yang, Y., Srinivasan, S., Hu, G., & Spanos, C. J. (2021a). Distributed control of multizone HVAC systems considering indoor air quality. *IEEE Transactions on Control Systems Technology*.

Zhang, L., Kekatos, V., & Giannakis, G. B. (2016). Scalable electric vehicle charging protocols. *IEEE Transactions on Power Systems*, *32*(2), 1451–1462.

Yu Yang is with School of Automation Science and Engineering, Xi'an Jiaotong University, Shaanxi, China. Prior to that, she received the B.E. degree in control science and engineering from Huazhong University of Science and Technology, Wuhan, China, in 2013, and the Ph.D. degree from Department of Automation, Tsinghua University, Beijing, China, in 2018. During 2018 to 2021, she worked as a postdoctoral scholar with Berkeley Education Alliance for Research in Singapore (BEARS), University of California, Berkeley. Her research interests lie in the control, optimization and decision-making of Cyber-Physical Energy Systems (CPES) including smart buildings and smart grids.

Qing-Shan Jia received the B.S. degree in automation and the Ph.D. degree in control science and engineering from Tsinghua University, Beijing, China, in 2002 and 2006, respectively. He was a Visiting Scholar at Harvard University, Cambridge, MA, USA, Hong Kong University of Science and Technology, Hong Kong, and Massachusetts Institute of Technology, Cambridge, in 2006, 2010, and 2013, respectively. He is currently a Professor with the Center for Intelligent and Networked Systems (CFINS), Department of Automation, Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University. His research interests include theories and applications of discrete-event dynamic systems (DEDSs) and simulation-based optimization of cyber–physical systems.

Zhanbo Xu received the B.E. degree in electrical engineering and automation from the Harbin Institute of Technology University, China, in 2008, and the Ph.D. degree in control science and engineering from Xi'an Jiaotong University, Xi'an, China, in 2015. He was a Post-Doctoral Scholar with the Berkeley Education Alliance for Research in Singapore from 2015 to 2017. He is currently a Professor with the Systems Engineering Institute, Xi'an Jiaotong University. His research interests include networked energy systems including smart power grid, energy internet, planning and scheduling of cyber–physical energy systems, building energy management, and smart city.

Xiaohong Guan received the B.S. and M.S. degrees in control engineering from Tsinghua University, Beijing, China, in 1982 and 1985, respectively, and the Ph.D. degree in electrical and systems engineering from the University of Connecticut, Mansfield, CT, USA, in 1993. He is currently a Professor with the Systems Engineering Institute, Xi'an Jiaotong University, Xi'an, China. In 1999, he was appointed Cheung Kong Professor of systems engineering, and in 2008, the Dean of the Faculty of Electronic and Information Engineering. Since 2001, he has been the Director of the Center for Intelligent and Networked Systems, Tsinghua University, and from 2003 to 2008, was the Head of the Department of Automation. His research interests include economics and security of networked systems, optimization-based planning and scheduling of power and energy systems, manufacturing systems, and cyber–physical systems, including smart grid and sensor networks. He is the Member of Chinese Academy of Science and is the Editor of IEEE TRANSACTIONS ON SMART GRID.

Costas J. Spanos received the EE Diploma from the National Technical University of Athens, Greece, and the M.S. and Ph.D. degrees in ECE from Carnegie Mellon University. In 1988 he joined the department of Electrical Engineering and Computer Sciences (EECS) at the University of California, Berkeley, where he is now the Andrew S. Grove Distinguished Professor and the Director of the Center for Information Technology Research in the Interest of Society and the Banatao institute (CITRIS).

He is also the Founding Director and CEO of the Berkeley Education Alliance for Research in Singapore (BEARS), and the Lead Investigator of a large research program on smart buildings based in California and Singapore. Prior to that, he has been the Chair of EECS at UC Berkeley, the Associate Dean for Research in the College of Engineering at UC Berkeley, and the Director of the UC Berkeley Microfabrication Laboratory. His research focuses on Sensing, Data Analytics, Modeling and Machine Learning, with broad applications in semiconductor technologies, and cyber–physical systems.