2018 52nd Annual Conference on Information Sciences and Systems (CISS)

An ADMM Approach to Dynamic Sharing Problems

Xuanyu Cao and K. J. Ray Liu

***Abstract*—In this paper, we study a dynamic version of the sharing problem, in which a dynamic system cost function composed of time-variant local costs of subsystems and a shared time-variant cost of the whole system is minimized. A dynamic alternating direction method of multipliers (ADMM) is proposed to track the varying optimal points of the dynamic optimization problem in an online manner. We analyze the convergence properties of the dynamic ADMM and show that, under several standard technical assumptions, the iterations of the dynamic ADMM converge linearly to some neighborhoods of the time-varying optimal points. The sizes of these neighborhoods depend on the drifts of the dynamic objective functions: the more drastically the dynamic objective function evolves across time, the larger the sizes of these neighborhoods. We also investigate the impact of the drifts on the steady state convergence behaviors of the dynamic ADMM. Finally, numerical results are presented to corroborate the effectiveness of the proposed dynamic ADMM.**

***Index Terms*—Dynamic optimization, the sharing problem, alternating direction method of multipliers**

I. Introduction

Many signal processing and resource allocation problems can be posed as an optimization problem which aims at minimizing a system cost consisting of local costs of subsystems and a shared cost of the whole system. For instance, consider a power system divided into multiple subsystems [1]. On one hand, if a subsystem receives some amount of power supplies, the consumption or storage of these supplies enables the subsystem to gain some utility. On the other hand, the generation of the total power supplies of all the subsystems incurs some cost for the whole power system due to factors such as the consumption of resources and the pollution. The goal of the designer or controller of the power system is to minimize the overall system cost including the negative of the total utilities of all the subsystems and the power generation cost of the whole system. This structure of local costs plus shared common cost arises in many applications such as smart grids, communication networks, or more generally, resource allocation in multi-agent systems. Optimization problems with such structure are called the sharing problems [2].

One implicit assumption of the conventional sharing problem is that both the local cost functions and the shared cost function are static, i.e., they do not vary with time. However, in practice, the cost or utility functions of many applications are intrinsically time-varying. For example, in power grids, the utility functions of the subsystems vary across time as the power users' demands evolve, e.g., the demands climax during evening and decline between midnight and early morning. The generation cost of the power system also varies with time owing to the changing and somewhat unpredictable renewable energy sources (e.g., wind and solar energy) as well as the fluctuation of the market prices of the traditional energy. Therefore, we are motivated to study a dynamic version of the sharing problem in this paper.

In the literature, dynamic optimization problems arise in various research fields and have been studied from different perspectives, such as adaptive signal processing [3], online convex optimization (OCO) [4]–[9] and online learning [10]–[15]. To solve the dynamic sharing problem in an online manner, in this paper, we present a dynamic ADMM algorithm. As a primal-dual method, the ADMM is superior to its primal domain counterparts such as the gradient descent method in terms of convergence speed. Due to its broad applicability, the ADMM has been exploited in various signal processing and control problems [16]–[20]. In addition, A few recent studies have investigated the performance of ADMM in a dynamic scenario. When the time-varying objective functions are unknown a-priori, an online ADMM algorithm is proposed in [21] to generate solutions with low regrets compared to the optimal static offline solution. This online ADMM is not directly applicable to many dynamic sharing problems in which the goal is to track the time-varying optimal points and a static offline benchmark is not very meaningful. A more closely related work is [22], in which a dynamic ADMM algorithm is applied to the consensus optimization problems. However, the convergence analysis of the dynamic ADMM in [22] significantly relies on the special structure of the consensus optimization problems, in which all agents share the same decision variable. This leaves the performance of the dynamic ADMM in other optimization scenarios unknown.

Our goal in this work is to investigate the convergence behaviors of the dynamic ADMM for the dynamic sharing problem both theoretically and empirically. Specifically, A dynamic ADMM algorithm (Algorithm 1) is proposed for a more general dynamic optimization problem (Problem (6)), which encompasses the dynamic sharing problem as a special case. The dynamic ADMM can adapt to the time-varying cost functions and track the optimal points in an online manner. We analyze the convergence properties of the proposed dynamic ADMM algorithm. We show that, under standard technical assumptions, the dynamic ADMM converges linearly to some neighborhoods of the time-varying optimal points. The sizes of the neighborhoods are related to the drifts of the dynamic optimization problem: the more drastically the dynamic problem evolves with time, the larger the sizes of the neighborhoods. We also study the impact of the drifts on the steady state convergence behaviors of the dynamic ADMM. Furthermore, numerical experiments are conducted to validate the effectiveness of the dynamic ADMM algorithm.

The remaining part of this paper is organized as follows. In Section II, the dynamic sharing problem is formally defined and a dynamic ADMM algorithm is proposed. In Section III, theoretical analysis of the convergence properties of the dynamic ADMM is presented. Two numerical examples are shown in Section IV, following which we conclude this work in Section V.

II. Problem Statement and Algorithm Development

In this section, we first formally state the dynamic sharing problem and give some examples and motivations for it. Then, we develop a dynamic ADMM algorithm for a more general dynamic optimization problem, which encompasses the dynamic sharing problem as a special case.

A. The Statement of the Problem

Consider the sharing problem [2]:

$$\text{Minimize} \quad \sum_{i=1}^{n} f^{(i)}\left(\mathbf{x}^{(i)}\right) + g\left(\sum_{i=1}^{n} \mathbf{x}^{(i)}\right), \tag{1}$$

with variables $\mathbf{x}^{(i)} \in \mathbb{R}^p$, $i = 1, ..., n$, where $f^{(i)} : \mathbb{R}^p \mapsto \mathbb{R}$ is the local cost function of subsystem i and $g : \mathbb{R}^p \mapsto \mathbb{R}$ is the global cost function of some commonly shared objective of all subsystems. The global cost function g takes the sum of all $\mathbf{x}^{(i)}$ as its input argument. The sharing problem (1) is a canonical problem with broad applications in resource allocation and signal processing [2]. One limitation of the problem formulation in (1) and its solution methods is that all the cost functions are static, i.e., they do not vary over time. This can be a major obstacle when the application is inherently time-variant and real-time, in which the cost functions change with time and online processing/optimization is imperative. In such circumstances, dynamic algorithms adaptive for the variation of the cost functions are more favorable. This motivates us to study a dynamic version of the sharing problem:

$$\text{Minimize} \quad \sum_{i=1}^{n} f_k^{(i)}\left(\mathbf{x}^{(i)}\right) + g_k\left(\sum_{i=1}^{n} \mathbf{x}^{(i)}\right), \tag{2}$$

where k is the time index. $f_k^{(i)} : \mathbb{R}^p \mapsto \mathbb{R}$ is the local cost function of subsystem i at time k and $g_k : \mathbb{R}^p \mapsto \mathbb{R}$ is the global cost function of the shared objective at time k.

The dynamic sharing problem in (2) can be applied to many dynamic resource allocation problems. For example, consider a power grid which is divided into n power subsystems according to either geographical locations or power line connections. If subsystem i receives $\mathbf{x}^{(i)}$ amount of power supplies at time k, then it gains a utility of $-f_k^{(i)}\left(\mathbf{x}^{(i)}\right)$ by either consuming or storing the supplies. In other words, $f_k^{(i)}$ is the negative of the utility function of power subsystem i at time k. The utility function is time-variant because users often have different power demands at different time, e.g., 6-11pm may be the peak demand period while 2-6am may be a low demand period. On the other hand, the generation of the total power supplies of $\sum_{i=1}^{n} \mathbf{x}^{(i)}$ can incur a cost of $g_k\left(\sum_{i=1}^{n} \mathbf{x}^{(i)}\right)$ for the power generator due to resource consumptions, human efforts and pollution. The generation cost function g_k also varies across time owing to factors such as the changing and somewhat unpredictable renewable energy sources and the variant prices of the traditional energy sources. Thus, the overall social welfare maximization problem can be posed as a dynamic sharing problem as in (2).

A well-known method to decouple the local cost functions $f^{(i)}$ and the global cost function g in the sharing problem (1) is the ADMM [2]. As a primal-dual optimization method, the ADMM has faster convergence than primal domain alternatives such as the gradient descent algorithm. This inspires us to develop and analyze a dynamic ADMM algorithm to solve the dynamic sharing problem in (2) in this work.

B. Development of the Dynamic ADMM

Define $\mathbf{x} = \left[\mathbf{x}^{(1)\mathsf{T}}, ..., \mathbf{x}^{(n)\mathsf{T}}\right]^{\mathsf{T}} \in \mathbb{R}^{np}$, $\mathbf{A} = [\mathbf{I}_p, ..., \mathbf{I}_p] \in \mathbb{R}^{p \times np}$ and

$$f_k(\mathbf{x}) = \sum_{i=1}^{n} f_k^{(i)}\left(\mathbf{x}^{(i)}\right). \tag{3}$$

Then, the dynamic sharing problem can be reformulated as:

$$\text{Minimize}_{\mathbf{x}\in\mathbb{R}^{np}, \mathbf{z}\in\mathbb{R}^{p}} \quad f_k(\mathbf{x}) + g_k(\mathbf{z}) \tag{4}$$

$$\text{s.t.} \quad \mathbf{A}\mathbf{x} - \mathbf{z} = \mathbf{0}. \tag{5}$$

In the remaining part of this paper, we study the following more general dynamic optimization problem:

$$\text{Minimize}_{\mathbf{x}\in\mathbb{R}^{N}, \mathbf{z}\in\mathbb{R}^{M}} \quad f_k(\mathbf{x}) + g_k(\mathbf{z}) \tag{6}$$

$$\text{s.t.} \quad \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{c}, \tag{7}$$

where $f_k : \mathbb{R}^N \mapsto \mathbb{R}$ and $g_k : \mathbb{R}^M \mapsto \mathbb{R}$ are two functions and $\mathbf{A} \in \mathbb{R}^{M\times N}, \mathbf{B} \in \mathbb{R}^{M\times M}$ are two matrices. The problem (4) is clearly a special case of the problem (6) by taking $N = np, M = p, \mathbf{B} = -\mathbf{I}, \mathbf{c} = \mathbf{0}$ and f_k decomposable as in (3). To apply the ADMM, we form the augmented Lagrangian of the problem (6):

$$\begin{aligned}\mathfrak{L}_{\rho,k}(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}) =& f_k(\mathbf{x}) + g_k(\mathbf{z}) + \boldsymbol{\lambda}^{\mathsf{T}}(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}) \\ &+ \frac{\rho}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}\|_2^2,\end{aligned} \tag{8}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^M$ is the Lagrange multiplier and $\rho > 0$ is some positive constant. Thus, applying the traditional ADMM to the dynamic augmented Lagrangian $\mathfrak{L}_{\rho,k}$, we propose a dynamic ADMM algorithm, as specified in Algorithm 1. The main difference between the dynamic ADMM in Algorithm 1 and the traditional static ADMM is that the functions f_k and g_k varies across iterations of the ADMM. The aim of this paper is to study the impact of these varying functions on the ADMM algorithm. Lastly, we introduce the following two linear convergence concepts which shall be used later.

Definition 1. *A sequence $\mathbf{s}_k$ is said to converge* <u>*Q-linearly*</u> *to $\mathbf{s}^*$ if there exists some constant $\theta \in (0,1)$ such that $\|\mathbf{s}_{k+1} - \mathbf{s}^*\|_2 \leq \theta \|\mathbf{s}_k - \mathbf{s}^*\|_2$ for any positive integer k.*

Definition 2. *A sequence $\mathbf{v}_k$ is said to converge* <u>*R-linearly*</u> *to $\mathbf{v}^*$ if there exists a positive constant $\tau > 0$ and some sequence $\mathbf{s}_k$ Q-linearly converging to some point $\mathbf{s}^*$ such that $\|\mathbf{v}_k - \mathbf{v}^*\|_2 \leq \tau \|\mathbf{s}_k - \mathbf{s}^*\|_2$ for every positive integer k.*

Algorithm 1 The dynamic ADMM algorithm for the dynamic problem (6)

1: Initialize $\mathbf{x}_0 = \mathbf{0}, \mathbf{z}_0 = \boldsymbol{\lambda}_0 = \mathbf{0}, k = 0$
2: **Repeat**:
3: $k \leftarrow k + 1$
4: Update $\mathbf{x}$ according to:

$$\mathbf{x}_k = \arg\min_{\mathbf{x}} f_k(\mathbf{x}) + \boldsymbol{\lambda}_{k-1}^\mathsf{T} \mathbf{A}\mathbf{x} + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}_{k-1} - \mathbf{c}\|_2^2. \tag{9}$$

5: Update $\mathbf{z}$ according to:

$$\mathbf{z}_k = \arg\min_{\mathbf{z}} g_k(\mathbf{z}) + \boldsymbol{\lambda}_{k-1}^\mathsf{T} \mathbf{B}\mathbf{z} + \frac{\rho}{2}\|\mathbf{B}\mathbf{z} + \mathbf{A}\mathbf{x}_k - \mathbf{c}\|_2^2. \tag{10}$$

6: Update $\boldsymbol{\lambda}$ according to:

$$\boldsymbol{\lambda}_k = \boldsymbol{\lambda}_{k-1} + \rho(\mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{z}_k - \mathbf{c}). \tag{11}$$

III. Convergence Analysis

In this section, convergence analysis for the dynamic ADMM algorithm, i.e., Algorithm 1, is conducted. We first make several standard assumptions for algorithm analysis. Then, we show that the iterations of the dynamic ADMM converge linearly (either Q-linearly or R-linearly) to some neighborhoods of their respective optimal points (Theorem 1 and 2). The sizes of these neighborhoods depend on the *drift* (to be formally defined later) of the dynamic optimization problem (6). Finally, we demonstrate the impact of the drift of the dynamic optimization problem (6) on the steady state convergence properties of the dynamic ADMM.

A. Assumptions

Throughout the convergence analysis, we make the following assumptions on the functions f_k and g_k, all of which are standard in the analysis of optimization algorithms [17], [23], [24].

Assumption 1. *For any positive integer k, g_k is strongly convex with constant $m > 0$ (m is independent of k), i.e., for any positive integer k:*

$$\left(\nabla g_k(\mathbf{z}) - \nabla g_k(\mathbf{z}')\right)^\mathsf{T} (\mathbf{z} - \mathbf{z}') \geq m\|\mathbf{z} - \mathbf{z}'\|_2^2, \quad \forall \mathbf{z}, \mathbf{z}' \in \mathbb{R}^M. \tag{12}$$

Assumption 2. *For any positive integer k, f_k is strongly convex with constant $\widetilde{m} > 0$ ($\widetilde{m}$ is independent of k), i.e., for any positive integer k:*

$$\left(\nabla f_k(\mathbf{x}) - \nabla f_k(\mathbf{x}')\right)^\mathsf{T} (\mathbf{x} - \mathbf{x}') \geq \widetilde{m}\|\mathbf{x} - \mathbf{x}'\|_2^2, \quad \forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^N. \tag{13}$$

Assumption 3. *For any positive integer k, ∇g_k is Lipschitz continuous with constant $L > 0$ (L is independent of k), i.e., for any positive integer k and any $\mathbf{z}, \mathbf{z}' \in \mathbb{R}^M$:*

$$\|\nabla g_k(\mathbf{z}) - \nabla g_k(\mathbf{z}')\|_2 \leq L\|\mathbf{z} - \mathbf{z}'\|_2. \tag{14}$$

Assumption 4. $\mathbf{B}$ *is nonsingular.*

B. Convergence Analysis

In this subsection, we study the convergence behavior of the proposed dynamic ADMM algorithm under the Assumptions 1-4. Due to the strong convexity assumption in Assumptions 1 and 2, there is a unique primal/dual optimal point pair $(\mathbf{x}_k^*, \mathbf{z}_k^*, \boldsymbol{\lambda}_k^*)$ for the dynamic optimization problem (6) at time k. Denote $\mathbf{u}_k = \left[\mathbf{z}_k^\mathsf{T}, \boldsymbol{\lambda}_k^\mathsf{T}\right]^\mathsf{T}$ and $\mathbf{u}_k^* = \left[\mathbf{z}_k^{*\mathsf{T}}, \boldsymbol{\lambda}_k^{*\mathsf{T}}\right]^\mathsf{T}$. Since $\mathbf{B}$ is a square matrix, the eigenvalues of $\mathbf{B}\mathbf{B}^\mathsf{T}$ are the same as those of $\mathbf{B}^\mathsf{T}\mathbf{B}$. Denote the smallest eigenvalue of $\mathbf{B}\mathbf{B}^\mathsf{T}$, which is also the smallest eigenvalue of $\mathbf{B}^\mathsf{T}\mathbf{B}$, as α. According to Assumption 4, $\mathbf{B}$ is nonsingular, so $\mathbf{B}\mathbf{B}^\mathsf{T}$ and $\mathbf{B}^\mathsf{T}\mathbf{B}$ are positive definite and $\alpha > 0$. Define matrix $\mathbf{C} \in \mathbb{R}^{2M \times 2M}$ to be:

$$\mathbf{C} = \begin{bmatrix} \frac{\rho}{2}\mathbf{B}^\mathsf{T}\mathbf{B} & \\ & \frac{1}{2\rho}\mathbf{I}_M \end{bmatrix} \tag{15}$$

Since $\mathbf{B}$ is nonsingular (Assumption 4), we know that $\mathbf{C}$ is positive definite. Therefore, we can define a norm on $\mathbb{R}^{2M}$ as $\|\mathbf{u}\|_\mathbf{C} = \sqrt{\mathbf{u}^\mathsf{T}\mathbf{C}\mathbf{u}}$. Define t to be any arbitrary number within the interval $(0,1)$. A positive constant $\delta > 0$ is defined as:

$$\delta = \min\left\{\frac{2mt}{\rho\|\mathbf{B}\|_2^2}, \frac{2\alpha\rho(1-t)}{L}\right\}, \tag{16}$$

where $\|\mathbf{B}\|_2$ is the spectral norm, i.e., the maximum singular value, of $\mathbf{B}$. Now, we are ready to show the first intermediate result, whose proof is omitted due to space limitation.

Proposition 1. *For any positive integer k, we have:*

$$\|\mathbf{u}_k - \mathbf{u}_k^*\|_\mathbf{C} \leq \frac{1}{\sqrt{1+\delta}}\|\mathbf{u}_{k-1} - \mathbf{u}_k^*\|_\mathbf{C}. \tag{17}$$

Remark 1. *Proposition 1 states that $\mathbf{u}_k$ is closer to $\mathbf{u}_k^*$ than $\mathbf{u}_{k-1}$ with a shrinkage factor of δ. The bigger the δ, the stronger the shrinkage. Note that there is an arbitrary factor $t \in (0,1)$ in the definition of δ in (16). By choosing $t = \frac{\alpha\rho^2\|\mathbf{B}\|_2^2}{mL+\alpha\rho^2\|\mathbf{B}\|_2^2}$, we get the maximum δ as $\delta_{\max} = \frac{2m\alpha\rho}{mL+\alpha\rho^2\|\mathbf{B}\|_2^2}$. In the expression of $\delta_{\max}$, only ρ is a tunable algorithm parameter while all other parameters are given by the optimization problem. The fact that $\delta_{\max}$ increases with ρ may partially justify the need of a relatively large ρ for good convergence behaviors of the dynamic ADMM. We will investigate the impact of ρ on algorithm performance empirically later.*

Proposition 1 establishes a relation between $\|\mathbf{u}_k - \mathbf{u}_k^*\|_\mathbf{C}$ and $\|\mathbf{u}_{k-1} - \mathbf{u}_k^*\|_\mathbf{C}$. However, to describe the convergence behavior of the dynamic ADMM algorithm, what we really want is the relation between $\|\mathbf{u}_k - \mathbf{u}_k^*\|_\mathbf{C}$ and $\|\mathbf{u}_{k-1} - \mathbf{u}_{k-1}^*\|_\mathbf{C}$. This is accomplished by the following theorem.

Theorem 1. *Define the drift d_k of the dynamic problem (6) to be:*

$$\begin{aligned} d_k = &\sqrt{\frac{\rho}{2}}\|\mathbf{B}\|_2\|\mathbf{z}_{k-1}^* - \mathbf{z}_k^*\|_2 \\ &+ \frac{1}{\sqrt{2\rho\alpha}}\|\nabla g_{k-1}(\mathbf{z}_{k-1}^*) - \nabla g_k(\mathbf{z}_k^*)\|_2. \end{aligned} \tag{18}$$

Then, for any integer $k \geq 2$, we have:

$$\|\mathbf{u}_k - \mathbf{u}_k^*\|_{\mathbf{C}} \leq \frac{1}{\sqrt{1+\delta}}(\|\mathbf{u}_{k-1} - \mathbf{u}_{k-1}^*\|_{\mathbf{C}} + d_k). \tag{19}$$

Proof. According to KKT optimality conditions, we have:

$$\nabla g_k(\mathbf{z}_k^*) + \mathbf{B}^\mathsf{T}\boldsymbol{\lambda}_k^* = \mathbf{0}, \tag{20}$$

$$\nabla g_{k-1}(\mathbf{z}_{k-1}^*) + \mathbf{B}^\mathsf{T}\boldsymbol{\lambda}_{k-1}^* = \mathbf{0}. \tag{21}$$

Substraction of (21) from (20) yields:

$$\mathbf{B}^\mathsf{T}(\boldsymbol{\lambda}_k^* - \boldsymbol{\lambda}_{k-1}^*) = -\nabla g_k(\mathbf{z}_k^*) + \nabla g_{k-1}(\mathbf{z}_{k-1}^*). \tag{22}$$

Hence,

$$\begin{aligned} &\|\nabla g_{k-1}(\mathbf{z}_{k-1}^*) - \nabla g_k(\mathbf{z}_k^*)\|_2^2 && (23)\\ &= \|\mathbf{B}^\mathsf{T}(\boldsymbol{\lambda}_{k-1}^* - \boldsymbol{\lambda}_k^*)\|_2^2 && (24)\\ &= (\boldsymbol{\lambda}_{k-1}^* - \boldsymbol{\lambda}_k^*)^\mathsf{T}\mathbf{B}\mathbf{B}^\mathsf{T}(\boldsymbol{\lambda}_{k-1}^* - \boldsymbol{\lambda}_k^*) && (25)\\ &\geq \alpha\|\boldsymbol{\lambda}_{k-1}^* - \boldsymbol{\lambda}_k^*\|_2^2. && (26) \end{aligned}$$

On the other hand,

$$(\mathbf{z}_{k-1}^* - \mathbf{z}_k^*)^\mathsf{T}\mathbf{B}^\mathsf{T}\mathbf{B}(\mathbf{z}_{k-1}^* - \mathbf{z}_k^*) \leq \|\mathbf{B}\|_2^2\|\mathbf{z}_{k-1}^* - \mathbf{z}_k^*\|_2^2. \tag{27}$$

Combining (26) and (27), we get:

$$\begin{aligned} &\|\mathbf{u}_{k-1}^* - \mathbf{u}_k^*\|_{\mathbf{C}}^2 && (28)\\ &= \frac{\rho}{2}(\mathbf{z}_{k-1}^* - \mathbf{z}_k^*)^\mathsf{T}\mathbf{B}^\mathsf{T}\mathbf{B}(\mathbf{z}_{k-1}^* - \mathbf{z}_k^*) + \frac{1}{2\rho}\|\boldsymbol{\lambda}_{k-1}^* - \boldsymbol{\lambda}_k^*\|_2^2 && (29)\\ &\leq \frac{\rho}{2}\|\mathbf{B}\|_2^2\|\mathbf{z}_{k-1}^* - \mathbf{z}_k^*\|_2^2 + \frac{1}{2\rho\alpha}\|\nabla g_{k-1}(\mathbf{z}_{k-1}^*) - \nabla g_k(\mathbf{z}_k^*)\|_2^2 && (30)\\ &\leq \Bigg(\sqrt{\frac{\rho}{2}}\|\mathbf{B}\|_2\|\mathbf{z}_{k-1}^* - \mathbf{z}_k^*\|_2 \\ &\quad + \frac{1}{\sqrt{2\rho\alpha}}\|\nabla g_{k-1}(\mathbf{z}_{k-1}^*) - \nabla g_k(\mathbf{z}_k^*)\|_2\Bigg)^2 && (31)\\ &= d_k^2. && (32) \end{aligned}$$

Thus, $\|\mathbf{u}_{k-1}^* - \mathbf{u}_k^*\|_{\mathbf{C}} \leq d_k$ and:

$$\begin{aligned} \|\mathbf{u}_{k-1} - \mathbf{u}_k^*\|_{\mathbf{C}} &\leq \|\mathbf{u}_{k-1} - \mathbf{u}_{k-1}^*\|_{\mathbf{C}} + \|\mathbf{u}_{k-1}^* - \mathbf{u}_k^*\|_{\mathbf{C}} \\ &\leq \|\mathbf{u}_{k-1} - \mathbf{u}_{k-1}^*\|_{\mathbf{C}} + d_k. \end{aligned} \tag{33}$$

Combining (33) and (17) in Proposition 1 gives:

$$\|\mathbf{u}_k - \mathbf{u}_k^*\|_{\mathbf{C}} \leq \frac{1}{\sqrt{1+\delta}}(\|\mathbf{u}_{k-1} - \mathbf{u}_{k-1}^*\|_{\mathbf{C}} + d_k). \tag{34}$$

□

Remark 2. *Theorem 1 means that $\mathbf{u}_k$ converges Q-linearly (with contraction factor $\sqrt{1+\delta}$) to some neighborhood of the optimal point $\mathbf{u}_k^*$. The size of the neighborhood is characterized by d_k, the drift of the dynamic problem (6), which is determined by the problem formulation instead of the algorithm. The more drastically the dynamic problem (6) varies across time, the bigger the drift d_k, and the larger the size of that neighborhood. When the dynamic problem (6) degenerates to its static counterpart, i.e., f_k and g_k does not vary with k, d_k becomes zero. In such a case, Theorem 1 degenerates to the linear convergence result of static ADMM in [24].*

Q-linear convergence of $\mathbf{u}_k$ to some neighborhood of the optimal point $\mathbf{u}_k^*$ is established in Theorem 1. A more meaningful result will be about the convergence properties of $\mathbf{x}_k, \mathbf{z}_k, \boldsymbol{\lambda}_k$. To this end, we want to link the quantities $\|\mathbf{x}_k - \mathbf{x}_k^*\|_2$, $\|\mathbf{z}_k - \mathbf{z}_k^*\|_2$, $\|\boldsymbol{\lambda}_k - \boldsymbol{\lambda}_k^*\|_2$ with $\|\mathbf{u}_k - \mathbf{u}_k^*\|_{\mathbf{C}}$. This is accomplished by the following theorem, the proof of which is omitted.

Theorem 2. *For any integer $k \geq 2$, we have:*

$$\begin{aligned} &\|\mathbf{x}_k - \mathbf{x}_k^*\|_2 \\ &\leq \frac{1}{\widetilde{m}}\|\mathbf{A}\|_2\Bigg[\left(\sqrt{2\rho} + \|\mathbf{B}\|_2\sqrt{\frac{2\rho}{\alpha}}\right)\|\mathbf{u}_k - \mathbf{u}_k^*\|_{\mathbf{C}} \\ &\quad + \|\mathbf{B}\|_2\sqrt{\frac{2\rho}{\alpha}}\|\mathbf{u}_{k-1} - \mathbf{u}_{k-1}^*\|_{\mathbf{C}} + \sqrt{2\rho}d_k\Bigg], \end{aligned} \tag{35}$$

where $\|\mathbf{A}\|_2$ is the spectral norm, i.e., the largest singular value, of $\mathbf{A}$. Furthermore, for any positive integer k, we have:

$$\|\mathbf{z}_k - \mathbf{z}_k^*\|_2 \leq \sqrt{\frac{2}{\alpha\rho}}\|\mathbf{u}_k - \mathbf{u}_k^*\|_{\mathbf{C}}, \tag{36}$$

$$\|\boldsymbol{\lambda}_k - \boldsymbol{\lambda}_k^*\|_2 \leq \sqrt{2\rho}\|\mathbf{u}_k - \mathbf{u}_k^*\|_{\mathbf{C}}. \tag{37}$$

Remark 3. *Sine $\mathbf{u}_k$ converges Q-linearly to some neighborhood of $\mathbf{u}_k^*$ (Theorem 1), Theorem 2 indicates that $\mathbf{x}_k, \mathbf{z}_k, \boldsymbol{\lambda}_k$ converge R-linearly to some neighborhoods of $\mathbf{x}_k^*, \mathbf{z}_k^*, \boldsymbol{\lambda}_k^*$, respectively. When the dynamic optimization problem (6) degenerates to its static version, i.e., f_k and g_k does not vary with k, Theorem 2 also degenerate to its static counterpart in [17], [24].*

To see the impact of the drift d_k (and thus the difference between the dynamic ADMM and the static ADMM) on the steady state convergence behaviors, we present the following result, the proof of which is omitted.

Theorem 3. *Suppose the drift defined in (18) satisfies $d_k \leq d, \forall k$, for some $d \in \mathbb{R}$. Then, we have:*

$$\limsup_{k\to\infty} \|\mathbf{u}_k - \mathbf{u}_k^*\|_{\mathbf{C}} \leq \frac{d}{\sqrt{1+\delta} - 1}, \tag{38}$$

$$\limsup_{k\to\infty} \|\mathbf{x}_k - \mathbf{x}_k^*\|_2 \leq \frac{\|\mathbf{A}\|_2}{\widetilde{m}}\left[\frac{\sqrt{2\rho} + \|\mathbf{B}\|_2\sqrt{\frac{8\rho}{\alpha}}}{\sqrt{1+\delta} - 1} + \sqrt{2\rho}\right]d, \tag{39}$$

$$\limsup_{k\to\infty} \|\mathbf{z}_k - \mathbf{z}_k^*\|_2 \leq \sqrt{\frac{2}{\alpha\rho}}\frac{d}{\sqrt{1+\delta} - 1}, \tag{40}$$

$$\limsup_{k\to\infty} \|\boldsymbol{\lambda}_k - \boldsymbol{\lambda}_k^*\|_2 \leq \sqrt{2\rho}\frac{d}{\sqrt{1+\delta} - 1}. \tag{41}$$

IV. Numerical Examples

In this section, numerical experiments are carried out to validate the effectiveness of the proposed dynamic ADMM algorithm, Algorithm 1. Specifically, we consider the following dynamic sharing problem:

$$\text{Minimize}_{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(n)}\in\mathbb{R}^p} \quad \sum_{i=1}^{n}\left(\mathbf{x}^{(i)}-\boldsymbol{\theta}_k^{(i)}\right)^{\mathsf{T}}\boldsymbol{\Phi}_k^{(i)}\left(\mathbf{x}^{(i)}-\boldsymbol{\theta}_k^{(i)}\right) + \gamma\left\|\sum_{i=1}^{n}\mathbf{x}^{(i)}\right\|_1, \tag{42}$$

where $\boldsymbol{\theta}_k^{(i)} \in \mathbb{R}^p$, $\boldsymbol{\Phi}_k^{(i)} \in \mathbb{R}^{p\times p}$ positive definite, $\gamma > 0$ are given problem data. The problem (42) is clearly in the form of (2) with:

$$f_k^{(i)}\left(\mathbf{x}^{(i)}\right) = \left(\mathbf{x}^{(i)}-\boldsymbol{\theta}_k^{(i)}\right)^{\mathsf{T}}\boldsymbol{\Phi}_k^{(i)}\left(\mathbf{x}^{(i)}-\boldsymbol{\theta}_k^{(i)}\right), \tag{43}$$

$$g_k(\mathbf{z}) = \gamma\|\mathbf{z}\|_1. \tag{44}$$

Define $\mathbf{x} = \left[\mathbf{x}^{(1)\mathsf{T}},\ldots,\mathbf{x}^{(n)\mathsf{T}}\right]^{\mathsf{T}}$, $\boldsymbol{\theta}_k = \left[\boldsymbol{\theta}_k^{(1)\mathsf{T}},\ldots,\boldsymbol{\theta}_k^{(n)\mathsf{T}}\right]^{\mathsf{T}}$ and $\boldsymbol{\Phi}_k = \operatorname{diag}\left(\boldsymbol{\Phi}_k^{(1)},\ldots,\boldsymbol{\Phi}_k^{(n)}\right)$. Thus, in terms of problem (4), we have:

$$f_k(\mathbf{x}) = \left(\mathbf{x}-\boldsymbol{\theta}_k\right)^{\mathsf{T}}\boldsymbol{\Phi}_k\left(\mathbf{x}-\boldsymbol{\theta}_k\right). \tag{45}$$

Applying the dynamic ADMM algorithm, i.e., Algorithm 1, to this dynamic sharing problem, we obtain Algorithm 2. The soft-threshold function $\mathcal{S}$ is defined for $a \in \mathbb{R}, \kappa > 0$ as follows:

$$\mathcal{S}_\kappa(a) = \begin{cases} a-\kappa, & \text{if } a > \kappa, \\ 0, & \text{if } |a| \le \kappa, \\ a+\kappa, & \text{if } a < \kappa. \end{cases} \tag{46}$$

In (48), an entrywise extension of the soft-threshold function to vector input is used.

Algorithm 2 The dynamic ADMM algorithm for the dynamic sharing problem (42)

1: Initialize $\mathbf{x}_0 = \mathbf{0}, \mathbf{z}_0 = \boldsymbol{\lambda}_0 = \mathbf{0}, k = 0$
2: **Repeat**:
3: $k \leftarrow k+1$
4: Update $\mathbf{x}$ according to:

$$\mathbf{x}_k = \left(2\boldsymbol{\Phi}_k + \rho\mathbf{A}^{\mathsf{T}}\mathbf{A}\right)^{-1}\left(2\boldsymbol{\Phi}_k\boldsymbol{\theta}_k - \mathbf{A}^{\mathsf{T}}\boldsymbol{\lambda}_{k-1} + \rho\mathbf{A}^{\mathsf{T}}\mathbf{z}_{k-1}\right). \tag{47}$$

5: Update $\mathbf{z}$ according to:

$$\mathbf{z}_k = \mathcal{S}_{\frac{\gamma}{\rho}}\left(\mathbf{A}\mathbf{x}_k + \frac{\boldsymbol{\lambda}_{k-1}}{\rho}\right). \tag{48}$$

6: Update $\boldsymbol{\lambda}$ according to:

$$\boldsymbol{\lambda}_k = \boldsymbol{\lambda}_{k-1} + \rho(\mathbf{A}\mathbf{x}_k - \mathbf{z}_k). \tag{49}$$

We generate the problem data $\boldsymbol{\Phi}_k^{(i)}$ and $\boldsymbol{\theta}_k^{(i)}$ recursively as follows. Given $\boldsymbol{\Phi}_{k-1}^{(i)}$ $(k \ge 1)$, we first generate $\widetilde{\boldsymbol{\Phi}}_k^{(i)}$ according to $\widetilde{\boldsymbol{\Phi}}_k^{(i)} = \boldsymbol{\Phi}_{k-1}^{(i)} + \eta_k^{(i)}\mathbf{E}_k^{(i)}$, where $\eta_k^{(i)}$ is some small positive number and $\mathbf{E}_k^{(i)}$ is a random symmetric matrix with entries uniformly distributed on $[-1, 1]$. Then, we construct the matrix $\boldsymbol{\Phi}_k^{(i)}$ as:

$$\boldsymbol{\Phi}_k^{(i)} = \begin{cases} \widetilde{\boldsymbol{\Phi}}_k^{(i)}, & \text{if } \lambda_{\min}\left(\widetilde{\boldsymbol{\Phi}}_k^{(i)}\right) \ge \epsilon, \text{ i.e., } \widetilde{\boldsymbol{\Phi}}_k^{(i)} \succeq \epsilon\mathbf{I}, \\ \widetilde{\boldsymbol{\Phi}}_k^{(i)} + \left[\epsilon - \lambda_{\min}\left(\widetilde{\boldsymbol{\Phi}}_k^{(i)}\right)\right]\mathbf{I}, & \text{otherwise}, \end{cases} \tag{50}$$

where $\lambda_{\min}(\cdot)$ denotes the smallest eigenvalue and $\epsilon > 0$ is some positive constant. Through this construction, we ensure that $\boldsymbol{\Phi}_k^{(i)} \succeq \epsilon\mathbf{I}, k = 1, 2, \ldots$. In addition, $\boldsymbol{\Phi}_0$ is a random symmetric matrix whose entries are uniformly distributed on $[-1, 1]$.

Given $\boldsymbol{\theta}_{k-1}^{(i)}$ $(k \ge 1)$, we generate $\boldsymbol{\theta}_k^{(i)}$ according to:

$$\boldsymbol{\theta}_k^{(i)} = \boldsymbol{\theta}_{k-1}^{(i)} + \eta_k^{(i)}\mathbf{h}_k^{(i)}, \tag{51}$$

where $\mathbf{h}_k^{(i)}$ is a random p-dimensional vector whose entries are uniformly distributed on $[-1, 1]$. $\boldsymbol{\theta}_0^{(i)}$ is also a random p-dimensional vector with entries uniformly distributed on $[-1, 1]$.

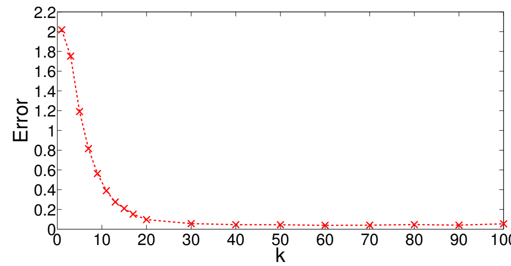

Fig. 1: The convergence curve of $\|\mathbf{x}_k - \mathbf{x}_k^*\|_2$. $\mathbf{x}_k^*$ is the optimal point of the dynamic sharing problem (42) at time k computed by an offline optimizor. $\mathbf{x}_k$ is the online solution given by the proposed dynamic ADMM algorithm, i.e., Algorithm 2.

In the first simulation, we set the parameters as $\eta = 0.2, \epsilon = 1, \gamma = 1, \rho = 1, p = 5, n = 20$. We use the CVX package [25] to compute the optimal point $\mathbf{x}_k^*$ of the instance of the dynamic sharing problem (42) at time k in an offline manner. The convergence curve of $\|\mathbf{x}_k - \mathbf{x}_k^*\|_2$ ($\mathbf{x}_k$ is the online solution given by the proposed dynamic ADMM algorithm, i.e., Algorithm 2) is shown in Fig. 1. The result is the average of 100 independent trials. We observe that $\mathbf{x}_k$ can converge to some neighborhood of $\mathbf{x}_k^*$ after about 30 iterations. This corroborates the theoretical results (Theorem 2 and Theorem 3) and the effectiveness of the proposed dynamic ADMM algorithm.

In the second simulation, we investigate the impact of the algorithm parameter ρ on the convergence performance of the dynamic ADMM. We consider three different values for ρ: $0.01, 0.1, 1$. The corresponding convergence curves ($\|\mathbf{x}_k - \mathbf{x}_k^*\|_2$) are shown in Fig. 2. We find that $\rho = 0.1$ yields the best convergence performance among the three circumstances. This indicates that the importance of an appropriate value of ρ,

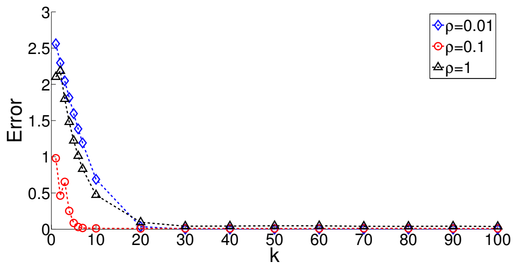

Fig. 2: The impact of the algorithm parameter ρ on the convergence behaviors ($\|\mathbf{x}_k - \mathbf{x}_k^*\|_2$) of the dynamic ADMM.

which should be neither too large nor too small. We note that similar observations have been made in the traditional static ADMM [2].

V. Conclusion

In this paper, motivated by the dynamic sharing problem, we propose and study a dynamic ADMM algorithm, which can adapt to the time-varying optimization problems in an online manner. Theoretical analysis is presented to show that the dynamic ADMM converges linearly to some neighborhood of the optimal point. The size of the neighborhood depends on the inherent evolution speed, i.e., the drift, of the dynamic optimization problem across time. The impact of the drift on the steady state convergence behaviors of the dynamic ADMM is also investigated. Finally, numerical results are presented to corroborate the effectiveness of the proposed dynamic ADMM.

References

[1] L. Gan, U. Topcu, and S. H. Low, "Optimal decentralized protocol for electric vehicle charging," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 940–951, 2013.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[3] S. Haykin, *Adaptive Filter Theory (3rd Ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996.

[4] E. Hazan *et al.*, "Introduction to online convex optimization," *Foundations and Trends® in Optimization*, vol. 2, no. 3-4, pp. 157–325, 2016.

[5] E. C. Hall and R. M. Willett, "Online convex optimization in dynamic environments," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 647–662, 2015.

[6] M. Mahdavi, R. Jin, and T. Yang, "Trading regret for efficiency: online convex optimization with long term constraints," *Journal of Machine Learning Research*, vol. 13, no. Sep, pp. 2503–2528, 2012.

[7] A. Koppel, F. Y. Jakubiec, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5149–5164, 2015.

[8] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: gradient descent without a gradient," in *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 385–394, Society for Industrial and Applied Mathematics, 2005.

[9] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," 2003.

[10] N. Littlestone and M. K. Warmuth, "The weighted majority algorithm," in *Foundations of Computer Science, 1989., 30th Annual Symposium on*, pp. 256–261, IEEE, 1989.

[11] S. Arora, E. Hazan, and S. Kale, "The multiplicative weights update method: a meta-algorithm and applications.," *Theory of Computing*, vol. 8, no. 1, pp. 121–164, 2012.

[12] J. Kivinen and M. K. Warmuth, "Exponentiated gradient versus gradient descent for linear predictors," *Information and Computation*, vol. 132, no. 1, pp. 1–63, 1997.

[13] C. Tekin, J. Yoon, and M. van der Schaar, "Adaptive ensemble learning with confidence bounds," *IEEE Transactions on Signal Processing*, vol. 65, no. 4, pp. 888–903, 2016.

[14] C. Tekin and M. van der Schaar, "Distributed online learning via cooperative contextual bandits," *IEEE Transactions on Signal Processing*, vol. 63, no. 14, pp. 3700–3714, 2015.

[15] C. Tekin, S. Zhang, and M. van der Schaar, "Distributed online learning in social recommender systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 4, pp. 638–652, 2014.

[16] Q. Ling and Z. Tian, "Decentralized sparse signal recovery for compressive sleeping wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 58, no. 7, pp. 3816–3827, 2010.

[17] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the admm in decentralized consensus optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1750–1761, 2014.

[18] C. Shen, T.-H. Chang, K.-Y. Wang, Z. Qiu, and C.-Y. Chi, "Distributed robust multicell coordinated beamforming with imperfect csi: An admm approach," *IEEE Transactions on signal processing*, vol. 60, no. 6, pp. 2988–3003, 2012.

[19] J. Zhang, S. Nabavi, A. Chakrabortty, and Y. Xin, "Admm optimization strategies for wide-area oscillation monitoring in power systems under asynchronous communication delays," *IEEE Transactions on Smart Grid*, vol. 7, no. 4, pp. 2123–2133, 2016.

[20] T.-H. Chang, "A proximal dual consensus admm method for multi-agent constrained optimization," *IEEE Transactions on Signal Processing*, vol. 64, no. 14, pp. 3719–3734, 2014.

[21] S. Hosseini, A. Chapman, and M. Mesbahi, "Online distributed admm via dual averaging," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pp. 904–909, IEEE, 2014.

[22] Q. Ling and A. Ribeiro, "Decentralized dynamic optimization through the alternating direction method of multipliers," *IEEE Transactions on Signal Processing*, vol. 62, no. 5, pp. 1185–1197, 2014.

[23] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[24] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," *Journal of Scientific Computing*, vol. 66, no. 3, pp. 889–916, 2016.

[25] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1." http://cvxr.com/cvx, mar 2014.