Solving Non-smooth Constrained Programs with Lower Complexity than $O(1/\varepsilon)$: A Primal-Dual Homotopy Smoothing Approach

Xiaohan Wei  
Department of Electrical Engineering  
University of Southern California  
Los Angeles, CA, USA, 90089  
xiaohanw@usc.edu

Hao Yu  
Alibaba Group (U.S.) Inc.  
Bellevue, WA, USA, 98004  
hao.yu@alibaba-inc.com

Qing Ling  
School of Data and Computer Science  
Sun Yat-Sen University  
Guangzhou, China, 510006  
lingqing556@mail.sysu.edu.cn

Michael J. Neely  
Department of Electrical Engineering  
University of Southern California  
Los Angeles, CA, USA, 90089  
mikejneely@gmail.com

Abstract

We propose a new primal-dual homotopy smoothing algorithm for a linearly constrained convex program, where neither the primal nor the dual function has to be smooth or strongly convex. The best known iteration complexity solving such a non-smooth problem is $O(1/\varepsilon)$. In this paper, we show that by leveraging a local error bound condition on the dual function, the proposed algorithm can achieve a better primal convergence time of $O\left(\varepsilon^{-2/(2+\beta)}\log_2(1/\varepsilon)\right)$, where $\beta \in (0, 1]$ is a local error bound parameter. As an example application of the general algorithm, we show that the distributed geometric median problem, which can be formulated as a constrained convex program, has its dual function non-smooth but satisfying the aforementioned local error bound condition with $\beta = 1/2$, therefore enjoying a convergence time of $O\left(\varepsilon^{-4/5}\log_2(1/\varepsilon)\right)$. This result improves upon the $O(1/\varepsilon)$ convergence time bound achieved by existing distributed optimization algorithms. Simulation experiments also demonstrate the performance of our proposed algorithm.

1 Introduction

We consider the following linearly constrained convex optimization problem:

\[
\begin{align*}
\min_{x} & \quad f(x) \\
\text{s.t.} & \quad Ax - b = 0, \quad x \in \mathcal{X},
\end{align*}
\]

where $\mathcal{X} \subseteq \mathbb{R}^d$ is a compact convex set, $f : \mathbb{R}^d \to \mathbb{R}$ is a convex function, $A \in \mathbb{R}^{N \times d}$, $b \in \mathbb{R}^N$. Such an optimization problem has been studied in numerous works under various application scenarios such as machine learning (Yurtsever et al. (2015)), signal processing (Ling and Tian (2010)) and communication networks (Yu and Neely (2017a)). The goal of this work is to design new algorithms for (1) achieving an $\varepsilon$ approximation with better convergence time than $O(1/\varepsilon)$.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.
1.1 Optimization algorithms related to constrained convex program

Since enforcing the constraint $Ax - b = 0$ generally requires a significant amount of computation in large scale systems, the majority of the scalable algorithms solving problem (1-2) are of primal-dual type. Generally, the efficiency of these algorithms depends on two key properties of the dual function of (1-2), namely, the Lipschitz gradient and strong convexity. When the dual function of (1-2) is smooth, primal-dual type algorithms with Nesterov’s acceleration on the dual of (1-2) can achieve a convergence time of $O(1/\sqrt{\epsilon})$ (e.g. Yurtsever et al. (2015); Tran-Dinh et al. (2018)). When the dual function has both the Lipschitz continuous gradient and the strongly convex property, algorithms such as dual subgradient and ADMM enjoy a linear convergence $O(\log(1/\epsilon))$ (e.g. Yu and Neely (2017); Necoara (2018); Deng and Yin (2016)). However, when neither of the properties is assumed, the basic dual-subgradient type algorithm gives a relatively worse $O(1/\epsilon^2)$ convergence time (e.g. Wei et al. (2015); Wei and Neely (2018)), while its improved variants yield a convergence time of $O(1/\epsilon)$ (e.g. Lan and Monteiro (2013); Deng et al. (2017); Yu and Neely (2017b); Yurtsever et al. (2015); Gidel et al. (2018)).

More recently, several works seek to achieve a better convergence time than $O(1/\epsilon)$ under weaker assumptions than Lipschitz gradient and strong convexity of the dual function. Specifically, building upon the recent progress on the gradient type methods for optimization with Hölder continuous gradient (e.g. Nesterov (2013a,b)), the work Yurtsever et al. (2015) develops a primal-dual gradient method solving (1-2), which achieves a convergence time of $O(1/\epsilon^{2+\mu})$, where $\nu$ is the modulus of Hölder continuity on the gradient of the dual function of the formulation (1-2). On the other hand, the work Yu and Neely (2018) shows that when the dual function has Lipschitz continuous gradient and satisfies a locally quadratic property (i.e. a local error bound with $\beta = 1/2$, see Definition 2.1 for details), which is weaker than strong convexity, one can still obtain a linear convergence with a dual-subgradient algorithm. A similar result has also been proved for ADMM in Han et al. (2015).

In the current work, we aim to address the following question: Can one design a scalable algorithm with lower complexity than $O(1/\epsilon)$ solving (1-2), when both the primal and the dual functions are possibly non-smooth? More specifically, we look at a class of problems with dual functions satisfying only a local error bound, and show that indeed one is able to obtain a faster primal convergence via a primal-dual homotopy smoothing method under a local error bound condition on the dual function.

Homotopy methods were first developed in the statistics literature in relation to the model selection problem for LASSO, where, instead of computing a single solution for LASSO, one computes a complete solution path by varying the regularization parameter from large to small (e.g. Osborne et al. (2000); Xiao and Zhang (2013)). On the other hand, the smoothing technique for minimizing a non-smooth convex function of the following form was first considered in Nesterov (2005):

$$\Psi(x) = g(x) + h(x), \ x \in \Omega_1$$

(3)

where $\Omega_1 \subseteq \mathbb{R}^d$ is a closed convex set, $h(x)$ is a convex smooth function, and $g(x)$ can be explicitly written as

$$g(x) = \max_{u \in \Omega_2} \langle Ax, u \rangle - \phi(u),$$

(4)

where for any two vectors $a, b \in \mathbb{R}^d$, $\langle a, b \rangle = a^T b$, $\Omega_1 \subseteq \mathbb{R}^d$ is a closed convex set, and $\phi(u)$ is a convex function. By adding a strongly concave proximal function of $u$ with a smoothing parameter $\mu > 0$ into the definition of $g(x)$, one can obtain a smoothed approximation of $\Psi(x)$ with smooth modulus $\mu$. Then, Nesterov (2005) employs the accelerated gradient method on the smoothed approximation (which delivers a $O(1/\sqrt{\epsilon})$ convergence time for the approximation), and sets the parameter to be $\mu = O(\epsilon)$, which gives an overall convergence time of $O(1/\epsilon)$. An important follow-up question is that whether or not such a smoothing technique can also be applied to solve (1-2) with the same primal convergence time. This question is answered in subsequent works Necoara and Suykens (2008); Lu et al. (2016); Tran-Dinh et al. (2018), where they show that indeed one can also obtain an $O(1/\epsilon)$ primal convergence time for the problem (1-2) via smoothing.

---

1 The gradient of function $g(\cdot)$ is Hölder continuous with modulus $\nu \in (0, 1]$ on a set $X$ if $||\nabla g(x) - \nabla g(y)|| \leq L_\nu ||x - y||^\nu, \forall x, y \in X$, where $|| \cdot ||$ is the vector 2-norm and $L_\nu$ is a constant depending on $\nu$.

2 The word “homotopy”, which was adopted in Osborne et al. (2000), refers to the fact that the mapping from regularization parameters to the set of solutions of the LASSO problem is a continuous piece-wise linear function.
Combining the homotopy method with a smoothing technique to solve problems of the form \( (3) \) has been considered by a series of works including \( \text{Yang and Lin (2015)}, \text{Xu et al. (2016)} \) and \( \text{Xu et al. (2017)} \). Specifically, the works \( \text{Yang and Lin (2015)} \) and \( \text{Xu et al. (2016)} \) consider a multi-stage algorithm which starts from a large smoothing parameter \( \mu \) and then decreases this parameter over time. They show that when the function \( \Psi(x) \) satisfies a local error bound with parameter \( \beta \in (0, 1] \), such a combination gives an improved convergence time of \( \mathcal{O}(\log(1/\varepsilon)) \) minimizing the unconstrained problem \( (3) \). The work \( \text{Xu et al. (2017)} \) shows that the homotopy method can also be combined with ADMM to achieve a faster convergence solving problems of the form \( (1) \).

**Contributions:** In the current work, we show a multi-stage homotopy smoothing method enjoys a primal convergence time \( \mathcal{O}(\varepsilon^{-2/(2+\beta)} \log_2(\varepsilon^{-1})) \) solving \( (12) \) when the dual function satisfies a local error bound condition with \( \beta \in (0, 1] \). Our convergence time to achieve within \( \varepsilon \) of optimality is in terms of number of (unconstrained) maximization steps \( \arg \max_{x \in \mathcal{X}} |x^T(Ax - b) - f(x)|^{-\frac{\beta}{\beta+1}} \), where constants \( \lambda, A, \bar{x}, \mu \) are known, which is a standard measure of convergence time for Lagrangian-type algorithms that turn a constrained problem into a sequence of unconstrained problems. The algorithm essentially restarts a weighted primal averaging process at each stage using the last Lagrange multiplier computed. This result improves upon the earlier \( \mathcal{O}(1/\varepsilon) \) result by \( \text{Necoara and Suykens (2008)}, \text{Li et al. (2016)} \) and at the same time extends the scope of homotopy smoothing method to solve a new class of problems involving constraints \( (12) \). It is worth mentioning that a similar restarted smoothing strategy is proposed in a recent work \( \text{tran-Dinh et al. (2018)} \) to solve problems including \( (12) \), where they show that, empirically, restarting the algorithm from the Lagrange multiplier computed from the last stage improves the convergence time. Here, we give one theoretical justification of such an improvement.

### 1.2 The distributed geometric median problem

The geometric median problem, also known as the Fermat-Weber problem, has a long history (e.g. see \( \text{Weiszfeld and Plastria (2009)} \) for more details). Given a set of \( n \) points \( b_1, b_2, \ldots, b_n \in \mathbb{R}^d \), we aim to find one point \( x^* \in \mathbb{R}^d \) so as to minimize the sum of the Euclidean distance, i.e.

\[
x^* \in \arg\min_{x \in \mathbb{R}^d} \sum_{i=1}^{n} \|x - b_i\|, \tag{5}
\]

which is a non-smooth convex optimization problem. It can be shown that the solution to this problem is unique as long as \( b_1, b_2, \ldots, b_n \in \mathbb{R}^d \) are not co-linear. Linear convergence time algorithms solving \( (3) \) have also been developed in several works (e.g. \( \text{Xue and Ye (1997)}, \text{Parrilo and Sturmfels (2003)}, \text{Cohen et al. (2016)} \)). Our motivation of studying this problem is driven by its recent application in distributed statistical estimation, in which data are assumed to be randomly spreaded to multiple connected computational agents that produce intermediate estimators, and then, these intermediate estimators are aggregated in order to compute some statistics of the whole data set. Arguably one of the most widely used aggregation procedures is computing the geometric median of the local estimators (see, for example, \( \text{Duchi et al. (2014)}, \text{Minsker et al. (2014)}, \text{Minsker and Strawn (2017)}, \text{Yin et al. (2018)} \)). It can be shown that the geometric median is robust against arbitrary corruptions of local estimators in the sense that the final estimator is stable as long as at least half of the nodes in the system perform as expected.

**Contributions:** As an example application of our general algorithm, we look at the problem of computing the solution to \( (3) \) in a distributed scenario over a network of \( n \) agents without any central controller, where each agent holds a local vector \( b_i \). Remarkably, we show theoretically that such a problem, when formulated as \( (12) \), has its dual function non-smooth but locally quadratic.  

\footnote{The result in \( \text{Xu et al. (2017)} \) heavily depends on the assumption that the subgradient of \( \psi(\cdot) \) is defined everywhere over the set \( \Omega_1 \) and uniformly bound by some constant \( \rho \), which excludes the choice of indicator functions necessary to deal with constraints in the ADMM framework.}
Therefore, applying our proposed primal-dual homotopy smoothing method gives a convergence time of \( O\left(e^{-4/5 \log_2(\varepsilon^{-1})}\right) \). This result improves upon the performance bounds of the previously known decentralized optimization algorithms (e.g. PG-EXTRA \cite{Shi2015} and decentralized ADMM \cite{Shi2014}), which do not take into account the special structure of the problem and only obtain a convergence time of \( O(1/\varepsilon) \). Simulation experiments also demonstrate the superior ergodic convergence time of our algorithm compared to other algorithms.

# 2 Primal-dual Homotopy Smoothing

## 2.1 Preliminaries

The Lagrange dual function of \((1-2)\) is defined as follows:\(^4\)

\[
F(\lambda) := \max_{x \in X} \left\{ -\langle \lambda, Ax - b \rangle - f(x) \right\},
\]

where \(\lambda \in \mathbb{R}^N\) is the dual variable, \(X\) is a compact convex set and the minimum of the dual function is \(F^* := \min_{\lambda \in \mathbb{R}^N} F(\lambda)\). For any closed set \(K \subseteq \mathbb{R}^d\) and \(x \in \mathbb{R}^d\), define the distance function of \(x\) to the set \(K\) as

\[
\text{dist}(x, K) := \min_{y \in K} \|x - y\|,
\]

where \(\|x\| := \sqrt{\sum_{i=1}^{d} x_i^2}\). For a convex function \(F(\lambda)\), the \(\delta\)-sublevel set \(S_\delta\) is defined as

\[
S_\delta := \{\lambda \in \mathbb{R}^N : F(\lambda) - F^* \leq \delta\}.
\]

Furthermore, for any matrix \(A \in \mathbb{R}^{N \times d}\), we use \(\sigma_{\max}(A^T A)\) to denote the largest eigenvalue of \(A^T A\). Let

\[
\Lambda^* := \{\lambda^* \in \mathbb{R}^N : F(\lambda^*) \leq F(\lambda), \forall \lambda \in \mathbb{R}^N\}
\]

be the set of optimal Lagrange multipliers. Note that if the constraint \(Ax = b\) is feasible, then \(\lambda^* \in \Lambda^*\) implies \(\lambda^* + \nu \in \Lambda^*\) for any \(\nu\) that satisfies \(A^T \nu = 0\). The following definition introduces the notion of local error bound.

**Definition 2.1.** Let \(F(\lambda)\) be a convex function over \(\lambda \in \mathbb{R}^N\). Suppose \(\Lambda^*\) is non-empty. The function \(F(\lambda)\) is said to satisfy the local error bound with parameter \(\beta \in (0, 1]\) if there exists a positive constant \(C_\delta\) such that for any \(\lambda \in S_\delta\),

\[
\text{dist}(\lambda, \Lambda^*) \leq C_\delta (F(\lambda) - F^*)^\beta,
\]

where \(C_\delta\) is a positive constant possibly depending on \(\delta\). In particular, when \(\beta = 1/2\), \(F(\lambda)\) is said to be locally quadratic and when \(\beta = 1\), it is said to be locally linear.

**Remark 2.1.** Indeed, a wide range of popular optimization problems satisfy the local error bound condition. The work \cite{Tseng2010} shows that if \(X\) is a polyhedron, \(f(\cdot)\) has Lipschitz continuous gradient and is strongly convex, then the dual function of \((1-2)\) is locally linear. The work \cite{Burke1997, Yang2013} shows that when the objective is linear and \(X\) is a convex cone, the dual function is also locally linear. The values of \(\beta\) have also been computed for several other problems (e.g. \cite{Pang1997}).

**Definition 2.2.** Given an accuracy level \(\varepsilon > 0\), a vector \(x_0 \in X\) is said to achieve an \(\varepsilon\) approximate solution regarding problem \((1-2)\) if

\[
f(x_0) - f^* \leq O(\varepsilon), \quad \|Ax_0 - b\| \leq O(\varepsilon),
\]

where \(f^*\) is the optimal primal objective of \((1-2)\).

Throughout the paper, we adopt the following assumptions:

**Assumption 2.1.** (a) The feasible set \(\{x \in X : Ax - b = 0\}\) is nonempty and non-singleton.
(b) The set \(X\) is bounded, i.e. \(\sup_{x, y \in X} \|x - y\| \leq D\), for some positive constant \(D\). Furthermore, the function \(f(x)\) is also bounded, i.e. \(\max_{x \in X} |f(x)| \leq M\), for some positive constant \(M\).
(c) The dual function defined in \((6)\) satisfies the local error bound for some parameter \(\beta \in (0, 1]\) and some level \(\delta > 0\).
(d) Let \(P_A\) be the projection operator onto the column space of \(A\). There exists a unique vector \(\nu^* \in \mathbb{R}^N\) such that for any \(\lambda^* \in \Lambda^*\), \(P_A \lambda^* = \nu^*\), i.e. \(\Lambda^* = \{\lambda^* \in \mathbb{R}^N : P_A \lambda^* = \nu^*\}\).

\(^4\) Usually, the Lagrange dual is defined as \(\min_{x \in X} \langle \lambda, Ax - b \rangle + f(x)\). Here, we flip the sign and take the maximum for no reason other than being consistent with the form \((6)\).
Note that assumption (a) and (b) are very mild and quite standard. For most applications, it is enough to check (c) and (d). We will show, for example, in Section 4 that the distributed geometric median problem satisfies all the assumptions. Finally, we say a function $f : \mathcal{X} \to \mathbb{R}$ is smooth with modulus $L > 0$ if
\[ \|\nabla g(x) - \nabla g(y)\| \leq L\|x - y\|, \forall x, y \in \mathcal{X}. \]

2.2 Primal-dual homotopy smoothing algorithm

This section introduces our proposed algorithm for optimization problem (1-2) satisfying Assumption 2.1. The idea of smoothing is to introduce a smoothed Lagrange dual function $F_\mu(\lambda)$ that approximates the original possibly non-smooth dual function $F(\lambda)$ defined in (6).

For any constant $\mu > 0$, define
\[ f_\mu(x) = f(x) + \frac{\mu}{2} \|x - \bar{x}\|^2, \tag{10} \]
where $\bar{x}$ is an arbitrary fixed point in $\mathcal{X}$. For simplicity of notation, we drop the dependency on $\bar{x}$ in the definition of $f_\mu(x)$. Then, by the boundedness assumption of $\mathcal{X}$, we have $f(x) \leq f_\mu(x) \leq f(x) + \frac{\mu}{2} D^2, \forall x \in \mathcal{X}$. For any $\lambda \in \mathbb{R}^N$, define
\[ F_\mu(\lambda) = \max_{x \in \mathcal{X}} -\langle \lambda, Ax - b \rangle - f_\mu(x) \tag{11} \]
as the smoothed dual function. The fact that $F_\mu(\lambda)$ is indeed smooth with modulus $\mu$ follows from Lemma 6.1 in the Supplement. Thus, one is able to apply an accelerated gradient descent algorithm on this modified Lagrange dual function, which is detailed in Algorithm 1 below, starting from an initial primal-dual pair $(\bar{x}, \bar{\lambda}) \in \mathbb{R}^d \times \mathbb{R}^N$.

Algorithm 1 Primal-Dual Smoothing: PDS$\left(\lambda, \bar{x}, \mu, T\right)$

Let $\lambda_0 = \lambda_{-1} = \bar{\lambda}$ and $\theta_0 = \theta_{-1} = 1$.
For $t = 0$ to $T - 1$ do
\begin{itemize}
  \item Compute a tentative dual multiplier: $\hat{\lambda}_t = \lambda_t + \theta_t(\theta_{t-1}^{-1} - 1)(\lambda_t - \lambda_{t-1})$.
  \item Compute the primal update: $x(\hat{\lambda}_t) = \arg\max_{x \in \mathcal{X}} \hat{\lambda}_t - \langle \hat{\lambda}_t, Ax - b \rangle - f(x) - \frac{\mu}{2} \|x - \bar{x}\|^2$.
  \item Compute the dual update: $\lambda_{t+1} = \lambda_t + \mu(Ax(\hat{\lambda}_t) - b)$.
  \item Update the stepsize: $\theta_{t+1} = \frac{\sqrt{\theta_t^2 + 4\mu^2} - \theta_t}{2}$.
\end{itemize}
end for

Output: $x_T = \frac{1}{S_T} \sum_{t=0}^{T-1} \frac{1}{\theta_t} x(\hat{\lambda}_t)$ and $\lambda_T$, where $S_T = \sum_{t=0}^{T-1} \frac{1}{\theta_t}$.

Our proposed algorithm runs Algorithm 1 in multiple stages, which is detailed in Algorithm 2 below.

Algorithm 2 Homotopy Method

Let $\varepsilon_0$ be a fixed constant and $\varepsilon < \varepsilon_0$ be the desired accuracy. Set $\mu_0 = \frac{\varepsilon_0}{D^2}$, $\lambda^{(0)} = 0$, $\bar{x}^{(0)} \in \mathcal{X}$, the number of stages $K \geq \lceil \log_2(\varepsilon_0/\varepsilon) \rceil + 1$, and the time horizon during each stage $T \geq 1$.
For $k = 1$ to $K$ do
\begin{itemize}
  \item Let $\mu_k = \mu_{k-1}/2$.
  \item Run the primal-dual smoothing algorithm $(\lambda^{(k)}, \bar{x}^{(k)}) = \text{PDS}\left(\lambda^{(k-1)}, \bar{x}^{(k-1)}, \mu_k, T\right)$.
\end{itemize}
end for

Output: $\bar{x}^{(K)}$. 

5
3 Convergence Time Results

We start by defining the set of optimal Lagrange multipliers for the smoothed problem:

$$\Lambda^*_\mu := \{ \lambda^*_\mu \in \mathbb{R}^N : F^*_\mu(\lambda^*_\mu) \leq F^*_\mu(\lambda), \ \forall \lambda \in \mathbb{R}^N \}$$ (12)

Our convergence time analysis involves two steps. The first step is to derive a primal convergence time bound for Algorithm 1, which involves the location information of the initial Lagrange multiplier at the beginning of this stage. The details are given in Supplement 6.2

**Theorem 3.1.** Suppose Assumption 2.1 holds. For any $T \geq 1$ and any initial vector $(\tilde{x}, \tilde{\lambda}) \in \mathbb{R}^d \times \mathbb{R}^N$, we have the following performance bound regarding Algorithm 1,

$$f(\vec{x}_T) - f^* \leq \| P_A \tilde{\lambda}^* - \vec{b} \| + \frac{\sigma_{\max}(A^T A)}{2\mu S_T} \| \tilde{\lambda}^* - \tilde{\lambda} \|^2 + \frac{\mu D^2}{2},$$ (13)

$$\| A \vec{x}_T - \vec{b} \| \leq \frac{2\sigma_{\max}(A^T A)}{\mu S_T} \left( \| \tilde{\lambda}^* - \tilde{\lambda} \|^2 + \text{dist}(\lambda^*_\mu, \Lambda^*) \right),$$ (14)

where $\tilde{\lambda}^* \in \arg\min_{\lambda^* \in \Lambda^*} \| \lambda^* - \tilde{\lambda} \|$, $\vec{x}_T := \frac{1}{\mu T} \sum_{t=0}^{T-1} \vec{x}(\vec{\lambda}_t)$, $S_T = \sum_{t=0}^{T-1} \frac{1}{\mu_t}$ and $\lambda^*_\mu$ is any point in $\Lambda^*_\mu$ defined in (12).

An inductive argument shows that $\theta_t \leq 2/(t+2) \ \forall t \geq 0$. Thus, Theorem 3.1 already gives an $O(1/\varepsilon)$ convergence time by setting $\mu = \varepsilon$ and $T = 1/\varepsilon$. Note that this is the best trade-off we can get from Theorem 3.1 when simply bounding the terms $\| \tilde{\lambda}^* - \tilde{\lambda} \|$ and $\text{dist}(\lambda^*_\mu, \Lambda^*)$ by constants. To see how this bound leads to an improved convergence time when running in multiple rounds, suppose the computation from the last round gives a $\tilde{\lambda}$ that is close enough to the optimal set $\Lambda^*$, then, $\| \tilde{\lambda}^* - \tilde{\lambda} \|$ would be small. When the local error bound condition holds, one can show that $\text{dist}(\lambda^*_\mu, \Lambda^*) \leq O(\mu^3)$. As a consequence, one is able to choose $\mu$ smaller than $\varepsilon$ and get a better trade-off. Formally, we have the following overall performance bound. The proof is given in Supplement 6.3

**Theorem 3.2.** Suppose Assumption 2.1 holds, $\varepsilon_0 \geq \max\{2M, 1\}$, $0 < \varepsilon \leq \min\{\delta/2, 2M, 1\}$, $T \geq \frac{2DC \sqrt{\sigma_{\max}(A^T A)(2M)^{3/2}}}{\varepsilon^{2/(2+\beta)}}$. The proposed homotopy method achieves the following objective and constraint violation bound:

$$f(\vec{x}^{(K)}) - f^* \leq \left( \frac{24\| P_A \lambda_s \| (1 + C_\delta)}{C_\delta^2 (2M)^{2\beta}} + \frac{6}{C_\delta^2 (2M)^{2\beta}} + \frac{1}{4} \right) \varepsilon,$$

$$\| A \vec{x}^{(K)} - \vec{b} \| \leq \frac{24(1 + C_\delta)}{C_\delta^2 (2M)^{3\beta}} \varepsilon,$$

with running time $\frac{2DC \sqrt{\sigma_{\max}(A^T A)(2M)^{3/2}}}{\varepsilon^{2/(2+\beta)}} ([\log_2(\varepsilon_0/\varepsilon)] + 1)$, i.e. the algorithm achieves an $\varepsilon$ approximation with convergence time $O\left(\varepsilon^{-2/(2+\beta)} \log_2(\varepsilon^{-1})\right)$.

4 Distributed Geometric Median

Consider the problem of computing the geometric median over a connected network $(V, E)$, where $V = \{1, 2, \ldots, n\}$ is a set of $n$ nodes, $E = \{e_{ij}\}_{i,j \in V}$ is a collection of undirected edges, $e_{ij} = 1$ if there exists an undirected edge between node $i$ and node $j$, and $e_{ij} = 0$ otherwise. Furthermore, $e_{ii} = 1$, $\forall i \in \{1, 2, \ldots, n\}$. Furthermore, since the graph is undirected, we always have $e_{ij} = e_{ji}$, $\forall i, j \in \{1, 2, \ldots, n\}$. Two nodes $i$ and $j$ are said to be neighbors of each other if $e_{ij} = 1$. Each node $i$ holds a local vector $b_i \in \mathbb{R}^d$, and the goal is to compute the solution to 5 without having a central controller, i.e. each node can only communicate with its neighbors.

Computing geometric median over a network has been considered in several works previously and various distributed algorithms have been developed such as decentralized subgradient method (DSM, Nedic and Ozdaglar (2009), Yuan et al. (2016)), PG-EXTRA (Shi et al. (2015)) and ADMM (Shi

---

\*By Assumption 2.1(a) and Farkas’ Lemma, this is non-empty.
et al. (2014); Deng et al. (2017)). The best known convergence time for this problem is $O(1/\varepsilon)$. In this section, we will show that it can be written in the form of problem (12), has its Lagrange dual function locally quadratic and optimal Lagrange multiplier unique up to the null space of $\mathbf{A}$, thereby satisfying Assumption 2.1.

Throughout this section, we assume that $n \geq 3$, $b_1$, $b_2$, \cdots, $b_n \in \mathbb{R}^d$ are not co-linear and they are distinct (i.e. $b_i \neq b_j$ if $i \neq j$). We start by defining a mixing matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$ with respect to this network. The mixing matrix will have the following properties:

1. Decentralization: The $(i, j)$-th entry $\bar{w}_{ij} = 0$ if $e_{ij} = 0$.
2. Symmetry: $\mathbf{W} = \mathbf{W}^T$.
3. The null space of $\mathbf{I}_{n \times n} - \mathbf{W}$ satisfies $\mathcal{N}(\mathbf{I}_{n \times n} - \mathbf{W}) = \{c \mathbf{1}, \ c \in \mathbb{R}\}$, where $\mathbf{1}$ is an all 1 vector in $\mathbb{R}^n$.

These conditions are rather mild and satisfied by most doubly stochastic mixing matrices used in practice. Some specific examples are Markov transition matrices of max-degree chain and Metropolis-Hastings chain (see Boyd et al. (2004) for detailed discussions). Let $x_i \in \mathbb{R}^d$ be the local variable on the node $i$. Define

$$x := \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{nd}, \quad b := \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \in \mathbb{R}^{nd}, \quad \mathbf{A} = \begin{bmatrix} W_{11} & \cdots & W_{1n} \\ \vdots & \ddots & \vdots \\ W_{n1} & \cdots & W_{nn} \end{bmatrix} \in \mathbb{R}^{(nd) \times (nd)},$$

where

$$W_{ij} = \begin{cases} (1 - \bar{w}_{ij})\mathbf{I}_{d \times d}, & \text{if } i = j \\ -\bar{w}_{ij}\mathbf{I}_{d \times d}, & \text{if } i \neq j \end{cases},$$

and $\bar{w}_{ij}$ is $ij$-th entry of the mixing matrix $\bar{W}$. By the aforementioned null space property of the mixing matrix $\bar{W}$, it is easy to see that the null space of the matrix $\mathbf{A}$ is

$$\mathcal{N}(\mathbf{A}) = \{u \in \mathbb{R}^{nd} : u = [u_1^T, \cdots, u_n^T]^T, \ u_1 = u_2 = \cdots = u_n\},$$

(15)

Then, because of the null space property (15), one can equivalently write problem (5) in a “distributed fashion” as follows:

$$\min \sum_{i=1}^{n} \|x_i - b_i\| \quad \text{s.t.} \quad \mathbf{Ax} = 0, \|x_i - b_i\| \leq D, \ i = 1, 2, \cdots, n,$$

(16)\hspace{1cm}(17)

where we set the constant $D$ to be large enough so that the solution belongs to the set $\mathcal{X} := \{x \in \mathbb{R}^{nd} : \|x_i - b_i\| \leq D, \ i = 1, 2, \cdots, n\}$. This is in the same form as (12) with $\mathcal{X} := \{x \in \mathbb{R}^{nd} : \|x_i - b_i\| \leq D, \ i = 1, 2, \cdots, n\}$.

### 4.1 Distributed implementation

In this section, we show how to implement the proposed algorithm to solve (16)-(17) in a distributed way. Let $\lambda_t = [\lambda_{t,1}, \lambda_{t,2}, \cdots, \lambda_{t,n}] \in \mathbb{R}^{nd}$, $\hat{\lambda}_t = [\hat{\lambda}_{t,1}, \hat{\lambda}_{t,2}, \cdots, \hat{\lambda}_{t,n}] \in \mathbb{R}^{nd}$ be the vectors of Lagrange multipliers defined in Algorithm 1, where each $\lambda_{t,i}, \hat{\lambda}_{t,i} \in \mathbb{R}^d$. Then, each agent $i \in \{1, 2, \cdots, n\}$ in the network is responsible for updating the corresponding Lagrange multipliers $\lambda_{t,i}$ and $\hat{\lambda}_{t,i}$ according to Algorithm 1, which has the initial values $\lambda_{0,i} = \lambda_{-1,i} = \hat{\lambda}_i$. Note that the first, third and fourth steps in Algorithm 1 are naturally separable regarding each agent. It remains to check if the second step can be implemented in a distributed way.

Note that in the second step, we obtain the primal update $x(\hat{\lambda}_t) = [x_1(\hat{\lambda}_t)^T, \cdots, x_n(\hat{\lambda}_t)^T] \in \mathbb{R}^{nd}$ by solving the following problem:

$$x(\hat{\lambda}_t) = \arg\max_{x: \|x_i - b_i\| \leq D, \ i = 1, 2, \cdots, n} \left\{ \left\langle \hat{\lambda}_t, \mathbf{A}x \right\rangle - \sum_{i=1}^{n} \left( \|x_i - b_i\| + \frac{\mu}{2} \|x_i - \bar{x}_i\|^2 \right) \right\},$$

(18)
Furthermore, there exists a unique vector $\tilde{\lambda}_t$ and $\lambda^*$ under different network
where we used the fact that

**Theorem 4.1.** The proof of this theorem is given in Supplement 6.5.

The solution to such a subproblem has a closed form, as is shown in the following lemma (the proof
is given in Supplement 6.4):

**Lemma 4.1.** Let $a_i = \tilde{x}_i - \frac{1}{\mu} \sum_{j \in N_i} W_{ji} \tilde{\lambda}_{t,j}$, then, the solution to (18) has the following closed form:

$$x_i(\tilde{\lambda}_t) = \begin{cases} b_i, & \text{if } \|b_i - a_i\| \leq 1/\mu, \\ b_i - \frac{b_i - a_i}{\|b_i - a_i\|} \left(\|b_i - a_i\| - \frac{1}{\mu}\right), & \text{if } \frac{1}{\mu} < \|b_i - a_i\| \leq \frac{1}{\mu} + D, \\ b_i - \frac{b_i - a_i}{\|b_i - a_i\|} D, & \text{otherwise.} \end{cases}$$

### 4.2 Local error bound condition

The proof of this theorem is given in Supplement 6.5.

**Theorem 4.1.** The Lagrange dual function of (16)-(17) is non-smooth and given by the following

$$F(\lambda) = -\langle A^T \lambda, b \rangle + D \sum_{i=1}^{n} (\|A_{i}[\lambda]\| - 1) \cdot I \left(\|A_{i}[\lambda]\| > 1\right),$$

where $A_{i}[.] = [W_{1i}, W_{2i}, \ldots, W_{ni}]^T$ is the $i$-th column block of the matrix $A$, $I \left(\|A_{i}[\lambda]\| > 1\right)$ is
the indicator function which takes 1 if $\|A_{i}[\lambda]\| > 1$ and 0 otherwise. Let $\Lambda^*$ be the set of optimal
Lagrange multipliers defined according to (8). Suppose $D \geq 2n \cdot \max_{i,j \in V} \|b_i - b_j\|$, then, for any
$\delta > 0$, there exists a $C_\delta > 0$ such that

$$\text{dist}(\lambda, \Lambda^*) \leq C_\delta (F(\lambda) - F^*)^{1/2}, \forall \lambda \in S_\delta.$$  

Furthermore, there exists a unique vector $\nu^* \in \mathbb{R}^{nd}$ s.t. $\mathcal{P}_A \lambda^* = \nu^*, \forall \lambda^* \in \Lambda^*$, i.e. Assumption 2.1(d) holds. Thus, applying the proposed method gives the convergence time $O\left(\varepsilon^{-4/5} \log_2(\varepsilon^{-1})\right)$.

## 5 Simulation Experiments

In this section, we conduct simulation experiments on the distributed geometric median problem. Each vector $b_i \in \mathbb{R}^{100}$, $i \in \{1, 2, \ldots, n\}$ is sampled from the uniform distribution in $[0, 10]^{100}$, i.e. each entry of $b_i$ is independently sampled from uniform distribution on $[0, 10]$. We compare our
algorithm with DSM (Nedic and Ozdaglar (2009)), P-EXTRA (Shi et al. (2015)), Jacobian parallel
ADMM (Deng et al. (2017)) and Smoothing (Necoara and Suykens (2008)) under different network
sizes ($n = 20, 50, 100$). Each network is randomly generated with a particular connectivity ratio and the mixing matrix is chosen to be the Metropolis-Hastings Chain (Boyd et al. (2004)), which can be computed in a distributed manner. We use the relative error as the performance metric, which is defined as $\|x_t - x^\ast\|/\|x_0 - x^\ast\|$ for each iteration $t$. The vector $x_0 \in \mathbb{R}^n$ is the initial primal variable. The vector $x^\ast \in \mathbb{R}^n$ is the optimal solution computed by CVX (Grant et al. (2008)). For our proposed algorithm, $x_t$ is the restarted primal average up to the current iteration. For all other algorithms, $x_t$ is the primal average up to the current iteration. The results are shown below. We see in all cases, our proposed algorithm is much better than, if not comparable to, other algorithms. For detailed simulation setups and additional simulation results, see Supplement 6.6.

Acknowledgments

The authors thank Stanislav Minsker and Jason D. Lee for helpful discussions related to the geometric median problem. Qing Ling’s research is supported in part by the National Science Foundation China under Grant 61573331 and Guangdong IIE Grant 2017ZT07X355. Michael J. Neely’s research is supported in part by the National Science Foundation under Grant CCF-1718477.

References


---

The connectivity ratio is defined as the number of edges divided by the total number of possible edges $n(n + 1)/2$. 


