Gradient Primal-Dual Algorithm Converges to Second-Order Stationary Solution for Nonconvex Distributed Optimization Over Networks

Mingyi Hong [1] **Jason D. Lee** [2] **Meisam Razaviyayn** [3]

Abstract

In this work, we study two first-order primal-dual based algorithms, the Gradient Primal-Dual Algorithm (GPDA) and the Gradient Alternating Direction Method of Multipliers (GADMM), for solving a class of linearly constrained non-convex optimization problems. We show that with random initialization of the primal and dual variables, both algorithms are able to compute second-order stationary solutions (ss2) with probability one. This is the first result showing that primal-dual algorithm is capable of finding ss2 when only using first-order information; it also extends the existing results for first-order, but *primal-only* algorithms. An important implication of our result is that it also gives rise to the first global convergence result to the ss2, for two classes of unconstrained *distributed* non-convex learning problems over multi-agent networks.

1. Introduction

Consider the following linearly constrained problem:

$$\min_{x\in\mathbb{R}^N} \ f(x) \quad \text{s.t.} \quad Ax = b \tag{1}$$

where $f(x) : \mathbb{R}^N \to \mathbb{R}$ is a smooth function; $A \in \mathbb{R}^{M\times N}$ is not full column rank; $b \in \mathbb{R}^M$ is a known vector.

An important application of problem (1) is in the non-convex distributed optimization and learning – a problem that has gained considerable attention recently, and has found applications in training neural networks (Lian et al., 2017), distributed information processing and machine learning (Forero et al., 2011; Hong et al., 2016), and distributed signal processing (Lorenzo & Scutari, 2016). In distributed optimization and learning, the common setup is that a network consists of N distributed agents collectively optimize the following problem

$$\min_{v\in\mathbb{R}} \quad \sum_{i=1}^N f_i(v) + g(v), \tag{2}$$

where $f_i(v) : \mathbb{R} \to \mathbb{R}$ is a function local to agent i (note, for notational simplicity we assume that v is a scalar); $g(v)$ represents some smooth regularization function known to all agents. Below we present two problem formulations based on different topologies and application scenarios.

Scenario 1: The Global Consensus. Suppose that all the agents are connected to a single central node. The distributed agents can communicate with the controller, but they are not able to directly communicate among themselves. In this case problem (2) can be equivalently formulated into the following global consensus problem (Boyd et al., 2011; Hong et al., 2016)

$$\min_{\{x_i\}_{i=0}^N} \ \sum_{i=1}^N f_i(x_i) + g(x_0), \quad \text{s.t.} \quad x_i = x_0, \ \forall\, i. \tag{3}$$

The setting of the above global consensus problem is popular in applications such as parallel computing, in which the existence of central controller can orchestrate the activity of all agents; see (Li et al., 2013; Zhang & Lin, 2015). To cast the problem into the form of (1), define

$$f(x) = \sum_{i=1}^N f_i(x_i) + g(x_0),$$
$$A_1 = I_N, \ A_2 = 1_N, \ A = [A_1, -A_2], \ b = 0, \tag{4}$$

where $I_N \in \mathbb{R}^{N\times N}$ is the identity matrix; $1_N \in \mathbb{R}^N$ is the all one vector.

Scenario 2: Distributed Optimization Over Networks. Suppose that there is no central controller, and the N agents are connected by a network defined by an *undirected* graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, with $|\mathcal{V}| = N$ vertices and $|\mathcal{E}| = E$ edges. Each agent can only communicate with its immediate neighbors, and it can access one component function f_i. This problem has wide applications ranging from distributed communication networking (Liao et al., 2015), distributed and parallel machine learning (Forero et al., 2011; Mateos et al., 2010; Shalev-Shwartz & Zhang, 2013), to distributed signal processing (Schizas et al., 2008).

Define the node-edge incidence matrix $A \in \mathbb{R}^{E\times N}$ as following: if $e \in \mathcal{E}$ and it connects vertex i and j with $i > j$,

[1]Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55414, USA. [2]Department of Data Sciences and Operations, the University of Southern California, Los Angeles, CA 90089. [3]Department of Industrial and Systems Engineering, the University of Southern California.. Correspondence to: Mingyi hong <mhong@umn.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

then $A_{ev} = 1$ if $v = i$, $A_{ev} = -1$ if $v = j$ and $A_{ev} = 0$ otherwise. Introduce N local variables $x = [x_1, \cdots, x_N]^T$, and suppose the graph $\{\mathcal{V}, \mathcal{E}\}$ is connected. Then as long as as the graph is connected, the following formulation is equivalent to the global consensus problem, which is precisely problem (1)

$$\min_{x\in\mathbb{R}^N} \; f(x) := \sum_{i=1}^{N} \left(f_i(x_i) + \frac{1}{N} g(x_i) \right), \text{ s.t. } Ax = 0. \quad (5)$$

1.1. The objective of this work

The research question we attempt to address in this work is:

(Q) Can we design primal-dual algorithms capable of computing second-order stationary solutions for (1)?

Let us first analyze the first-order stationary (ss1) and second-order stationary (ss2) solutions for problem (1). For a general smooth nonlinear problem in the following form

$$\min_{x\in\mathbb{R}^N} \; g(x) \quad \text{s.t.} \quad h_i(x) = 0, \quad i = 1, \cdots, m, \quad (6)$$

the first-order necessary condition is given as

$$\nabla g(x^*) + \sum_{i=1}^{m} \langle \lambda_i^*, \nabla h_i(x^*) \rangle = 0, \quad h_i(x^*) = 0, \; \forall\, i. \quad (7)$$

The second-order necessary condition is given below [see Proposition 3.1.1 in (Bertsekas, 1999)]. Suppose x^* is regular, then

$$\begin{aligned} &\langle y, (\nabla^2 g(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla^2 h_i(x^*)) y \rangle \geq 0, \\ &\forall\, y \in \{ y \neq 0 \mid \langle \nabla h_i(x^*), y \rangle = 0, \; \forall\, i = 1, \cdots, m \}. \end{aligned} \quad (8)$$

Applying the above result to our problem, we obtain the following first- and second-order necessary condition for problem (1) [1]

$$\nabla f(x^*) + A^T \lambda^* = 0, \quad Ax^* = b. \quad (9a)$$

$$\langle y, \nabla^2 f(x^*) y \rangle \geq 0, \quad \forall \;\; y \in \{ y \mid Ay = 0 \}. \quad (9b)$$

In other words, the second-order necessary condition is equivalent to the condition that $\nabla^2 f(x^*)$ is positive semidefinite in the null space of A. Similarly, the sufficient condition for *strict* local minimizer is given by

$$\begin{aligned} &\nabla f(x^*) + A^T \lambda^* = 0, \quad Ax^* = b. \quad (10) \\ &\langle y, \nabla^2 f(x^*) y \rangle > 0, \quad \forall \;\; y \neq 0, \text{ and } y \in \{ y \mid Ay = 0 \}. \end{aligned}$$

To proceed, we need the following claim [see Lemma 3.2.1 in (Bertsekas, 1999)]

Claim 1. Let P and Q be two symmetric matrices. Assume that Q is positive semidefinite and P is positive definite on the null space of Q, that is, $x^T P x > 0$ for all $x \neq 0$ with $x^T Q x = 0$. Then there exists a scalar $\bar{c}$ such that

$$P + cQ \succ 0, \quad \forall\, c \geq \bar{c}. \quad (11)$$

Conversely, if there exists a scalar $\bar{c}$ such that (11) is true, then we have $x^T P x > 0$ for all $x \neq 0$ with $x^T Q x = 0$.

By Claim 1, the sufficient condition (10) can be equivalently written as:

$$\nabla f(x^*) + A^T \lambda^* = 0, \quad Ax^* = b. \quad (12)$$

$$\nabla^2 f(x^*) + \gamma A^T A \succ 0, \text{ for some } \gamma > 0. \quad (13)$$

It is worth mentioning that checking both of the above sufficient and necessary conditions can be done in polynomial time, but when there are inequality constraints, checking second-order conditions can be NP-hard; see (Murty & Kabadi, 1987). In the following we will refer to the condition (9a) as an ss1 solution and condition (9b) as an ss2 solution. According to the above definition, we define a *strict saddle* point to be the solution x^* such that

$$\begin{aligned} &\nabla f(x^*) + A^T \lambda^* = 0, \quad Ax^* = b, \\ &\exists\, y \in \{ y \mid Ay = 0, y \neq 0 \}, \text{ and } \sigma > 0 \\ &\quad \text{such that } \langle y, \nabla^2 f(x^*) y \rangle \leq -\sigma \|y\|^2. \end{aligned} \quad (14)$$

It is easy to verify using Claim 1 that the above condition implies that for the same $\sigma > 0$, the following is true

$$\begin{aligned} &\nabla f(x^*) + A^T \lambda^* = 0, \quad Ax^* = b, \\ &\sigma_{\min} \left(\gamma A^T A + \nabla^2 f(x^*) \right) \leq -\sigma, \quad \forall\, \gamma > 0 \end{aligned} \quad (15)$$

where $\sigma_{\min}$ denotes the smallest eigenvalue of a matrix. Clearly, if a ss1 solution x^* does not satisfy (14), i.e.,

$$\forall\, y, \text{ s.t. } Ay = 0, \quad \langle y, \nabla f^2(x^*) y \rangle \geq 0, \quad (16)$$

then (9b) is true. In this work, we will develop primal-dual algorithms that avoid converging to the strict saddles (14).

1.2. Existing literature

Many recent works have been focused on designing algorithms with convergence guarantees to local minimum points/ss2 for non-convex unconstrained problems. These include second-order methods such as trust region method (Conn et al., 2000), cubic regularized Newton's method (Nesterov & Polyak, 2006), and a hybrid of first-order and second-order methods (Reddi et al., 2017). When only gradient information is available, it has been shown that with random initialization, gradient descent (GD) converges to ss2 for unconstrained smooth problems with probability one (Lee et al., 2016a). Recently, a perturbed version of GD which occasionally adds noise to the iterates has been proposed (Jin et al., 2017), and such a method converges to the ss2 with faster convergence rate than the ordinary gradient descent algorithm with random initialization. When manifold constraints are present, it is shown in (Lee et al., 2017)

[1] Note that for linear constraints no further regularity is needed for the existence of multipliers

that manifold gradient descent converges to ss2, provided that each time the iterates are always feasible (ensured by performing a potentially expensive second-order retraction operation). However, there has been no work analyzing whether classical primal-dual gradient type methods based on Lagrangian relaxation are also capable of computing ss2.

The consensus problem (2) and (5) have been studied extensively in the literature when the objective functions are all convex; see for example (Aybat & Hamedani, 2016; Nedic & Olshevsky, 2015; Nedic & Ozdaglar, 2009; Shi et al., 2014). Primal methods such as distributed subgradient method (Nedic & Ozdaglar, 2009), the EXTRA method (Shi et al., 2014), as well as primal-dual based methods such as Alternating Direction Method of Multipliers (ADMM) (Boyd et al., 2011; Chang et al., 2015; Schizas et al., 2009) have been studied. On the contrary, only recently there have been some work addressing the more challenging problems without assuming convexity of f_i's; see recent developments in (Bianchi & Jakubowicz, 2013; Hong et al., 2016; 2017; Lorenzo & Scutari, 2016). In particular, reference (Hong et al., 2016) develops non-convex ADMM based methods (with global sublinear convergence rate) for solving the global consensus problem (3). Reference (Hong et al., 2017) proposes a primal-dual based method for unconstrained non-convex distributed optimization over a connected network (without a central controller), and derives the first global convergence rate for distributed non-convex optimization. In (Lorenzo & Scutari, 2016) the authors utilize certain gradient tracking idea to solve a constrained nonsmooth distributed problem over possibly time-varying networks. It is worth noting that the distributed algorithms proposed in all these works converge to ss1. There has been no distributed schemes that can provably converge to ss2 for smooth non-convex problem in the form of (2).

2. The Gradient Primal-Dual Algorithm

In this section, we introduce the gradient primal-dual algorithm (GPDA) for solving the non-convex problem (1). Let us introduce the augmented Lagrangian (AL) as

$$L(x,y) = f(x) + \langle \lambda, Ax - b\rangle + \frac{\rho}{2}\|Ax-b\|^2, \quad (17)$$

where $\lambda \in \mathbb{R}^M$ is the dual variable. The steps of the GPDA algorithm are described in the table below.

Each iteration of the GPDA performs a gradient descent step on the AL (with stepsize being $1/\beta$), followed by taking one step of approximate dual gradient ascent (with stepsize $\rho > 0$). The GPDA is closely related to the classical Uzawa primal-dual method (Uzawa, 1958), which has been utilized to solve *convex* saddle point problems and linearly constrained *convex* problems (Nedić & Ozdaglar, 2009). It is also related to the proximal method of multipliers (Prox-MM) first developed by Rockafellar in (Rockafellar, 1976), in which a proximal term has been added to the augmented Lagrangian in order to make it strongly convex in each iteration. The latter method has also been applied for example, in solving certain large-scale linear programs; see (J.Wright, 1990). However the theoretical results derived for Prox-MM in (J.Wright, 1990; Rockafellar, 1976) are only developed for convex problems. Further, such an algorithm requires that the proximal Lagrangian to be optimized with *increasing* accuracy as the algorithm progresses. Finally, we note that both step (18a) and (18b) can be decomposable over the variables, therefore they are easy to be implemented in a distributed manner (as will be explained shortly).

Algorithm 1. The gradient primal-dual algorithm

At iteration 0, initialize λ^0 and x^0.

At each iteration $r+1$, update variables by:

$$x^{r+1} = \arg\min \langle \nabla f(x^r) + A^T\lambda^r + \rho A^T(Ax^r - b), x - x^r\rangle + \frac{\beta}{2}\|x - x^r\|^2 \quad (18a)$$

$$\lambda^{r+1} = \lambda^r + \rho\left(Ax^{r+1} - b\right) \quad (18b)$$

2.1. Application in distributed optimization problem

To see how the GPDA can be specialized to the problem of distributed optimization over the network (5), let us begin by writing the optimality condition of (18a). We have

$$\nabla f(x^r) + A^T\lambda^r + \rho A^TAx^r + \beta(x^{r+1} - x^r) = 0. \quad (19)$$

Subtracting (19) with its counterpart at iteration r, we obtain

$$\nabla f(x^r) - \nabla f(x^{r-1}) + A^T(\lambda^r - \lambda^{r-1}) + \rho A^TA(x^r - x^{r-1}) + \beta w^{r+1} = 0$$

where we have defined $w^{r+1} = (x^{r+1} - x^r) - (x^r - x^{r-1})$. Rearrange, and use the fact that $A^TA = L_- \in \mathbb{R}^{N\times N}$ is the *signed Laplacian matrix*, and $b = 0$ in (5), we obtain

$$x^{r+1} = x^r + (x^r - x^{r-1}) + \frac{1}{\beta}\big(-\nabla f(x^r) + \nabla f(x^{r-1}) - \rho L_- x^r - \rho L_-(x^r - x^{r-1})\big). \quad (20)$$

Consider problem (5) (for simplicity assume that $g \equiv 0$), the above iteration can be implemented in a distributed manner, where each agent i performs

$$x_i^{r+1} = x_i^r + (x_i^r - x_i^{r-1}) + \frac{1}{\beta}\Bigg(-\nabla f_i(x_i^r) + \nabla f_i(x_i^{r-1}) - 2\rho\left(d_i x_i^r - \sum_{j\in\mathcal{N}_i} x_j^r\right) + \rho\big(d_i x_i^{r-1} - \sum_{j\in\mathcal{N}_i} x_j^{r-1}\big)\Bigg),$$

where $\mathcal{N}_i := \{j \mid j \neq i, (i,j) \in \mathcal{E}\}$ is the set of neighbors of node i; d_i is the degree for node i. Clearly, to implement this iteration each node only needs to know the information from the past two iterations about its immediate neighbors.

2.2. Convergence to ss1 solutions

We first state our main assumptions.

A1. The function $f(x)$ is smooth and has Lipschitz continuous gradient, as well as Lipschitz continuous Hessian:

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x-y\|,\ \forall\, x, y \in \mathbb{R}^N \quad (21)$$

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq M\|x-y\|,\ \forall\, x, y \in \mathbb{R}^N\ . \quad (22)$$

A2. The function $f(x)$ is lower bounded over $x \in \mathbb{R}^N$. Without loss of generality, assume that $f(x) \geq 0$.

A3. The constraint $Ax = b$ is feasible over $x \in X$. Further, A^TA is not full rank.

A4. The function $f(x) + \frac{\rho}{2}\|Ax-b\|^2$ is coercive.

A5. The function f is proper and it satisfies the Kurdyka-Łojasiewicz (KŁ) property. That is, at $\widehat{x} \in \mathbb{R}$ if there exist $\eta \in (0\ \infty]$, a neighborhood V of $\widehat{x}$ and a continuous concave function $\phi : [0,\ \eta) \to \mathbb{R}_+$ such that: 1) $\phi(0) = 0$ and ϕ is continuously differentiable on $[0,\ \eta]$ with positive derivatives; 2) for all $x \in \mathbb{R}^N$, satisfying $f(\widehat{x}) < f(x) < f(\widehat{x}) + \eta$, it holds that

$$\phi'(f(x) - f(\widehat{x}))\text{dist}(0, \partial f(x)) \geq 1 \quad (23)$$

where $\partial f(x)$ is the limiting subdifferential defined as

$$\partial f(x) = \Bigg\{ \{v \in \mathbb{R}^N : \exists x^t \to x, v^t \to v,$$
$$\text{with } \lim_{z \to x^t} \inf \frac{f(x) - f(x^t) - \langle v^t, z - x^t\rangle}{\|x - x^t\|} \geq 0, \forall\, t \Bigg\}$$

We comment that a wide class of functions enjoys the KŁ property, for example a semi-algebraic function is a KL function; for detailed discussions of the KŁ property we refer the readers to (Bolte et al., 2014; Li & Pong, 2014a).

Below we will use $\sigma_i(\cdot)$, $\sigma_{\max}(\cdot)$, $\sigma_{\min}(\cdot)$ and $\widetilde{\sigma}_{\min}(\cdot)$ to denote the ith, the maximum, the minimum, and the smallest *non-zero* eigenvalues of a matrix, respectively.

The convergence of GPDA to the ss1 is similar to Theorem 3.1 in (Hong, 2016) and Corollay 4.1 in (Hong, 2016). Algorithmically, the main difference is that the algorithms analyzed in (Hong, 2016) do not linearize the penalty term $\frac{\rho}{2}\|Ax-b\|^2$, and they make use of the same penalty and proximal parameters, that is, $\rho = \beta$. In this work, in order to show the convergence to ss2, we need to have the freedom of tuning β while fixing ρ, therefore β and ρ have to be chosen differently. However, in terms of analysis, there is no major difference between these versions. For completeness, we only outline the key proof steps in the Appendix.

Claim 2. Suppose Assumptions [A1] – [A5] are satisfied. For appropriate choices of ρ, and β satisfying (67) given in the appendix, and starting from any feasible point (x^0, λ^0), the GPDA converges to the set of ss1 solutions.

Further, if $L(x^r, \lambda^r)$ is a KŁ function, then (x^{r+1}, λ^{r+1}) converges globally to a unique point (x^*, λ^*).

2.3. Convergence to ss2

One can view Claim 2 as some variation of known results. On the contrary, in this section we show one of the main contributions of this work, which demonstrates that GPDA can converge to solutions beyond the ss1.

To this end, first let us rewrite the x update step using its first-order optimality condition as follows

$$x^{r+1} = x^r - \frac{1}{\beta}\left(\nabla f(x^r) + A^T\lambda^r + \rho A^T(Ax^r - b)\right).$$

Therefore the iteration can be written as

$$\begin{bmatrix} x^{r+1} \\ \lambda^{r+1} \end{bmatrix} = \begin{bmatrix} x^r - \frac{1}{\beta}\left(\nabla f(x^r) + A^T\lambda^r + \rho A^T(Ax^r - b)\right) \\ \lambda^r + \rho(Ax^{r+1} - b) \end{bmatrix}$$
$$= \begin{bmatrix} x^r - \frac{1}{\beta}\left(\nabla f(x^r) + A^T\lambda^r + \rho A^T(Ax^r - b)\right) \\ \lambda^r + \rho\left(A\left(x^r - \frac{1}{\beta}\left(\nabla f(x^r) + A^T\lambda^r + \rho A^T(Ax^r - b)\right)\right) - b\right) \end{bmatrix}$$

The compact way to write the above iteration is

$$\begin{bmatrix} I_N & 0_{N\times M} \\ -\rho A & I_M \end{bmatrix} \begin{bmatrix} x^{r+1} \\ \lambda^{r+1} \end{bmatrix} \quad (24)$$
$$= \begin{bmatrix} x^r - \frac{1}{\beta}\left(\nabla f(x^r) + A^T\lambda^r + \rho A^T(Ax^r - b)\right) \\ \lambda^r - \rho b \end{bmatrix},$$

where I_N denotes the N-by-N identity matrix $0_{N\times M}$ denotes the N-by-M all zero matrix.

Next let us consider approximating $\nabla f(x)$ near a first-order stationary solution x^*. Let us define

$$H := \nabla^2 f(x^*), \quad d^{r+1} := -x^* + x^{r+1}.$$

Claim 2 implies that when ρ, β are chosen appropriately, then $d^{r+1} \to 0$. Therefore for any given $\xi > 0$ there exists an iteration index $R(\xi) > 0$ such that the following holds

$$\|d^{r+1}\| \leq \xi, \quad \forall\, r - 1 \geq R(\xi). \quad (25)$$

Next let us approximate the gradients around $\nabla f(x^*)$:

$$\begin{aligned} &\nabla f(x^{r+1}) = \nabla f(x^* + d^{r+1}) \\ &= \nabla f(x^*) + \int_0^1 \nabla^2 f(x^* + td^{r+1})d^{r+1}dt \\ &= \nabla f(x^*) + \int_0^1 (\nabla^2 f(x^* + td^{r+1}) - H)d^{r+1}dt + Hd^{r+1} \\ &:= \nabla f(x^*) + \Delta^{r+1}d^{r+1} + Hd^{r+1}, \end{aligned} \quad (26)$$

where in the last inequality we have defined

$$\Delta^{r+1} := \int_0^1 (\nabla^2 f(x^* + td^{r+1}) - H)d^{r+1}dt. \quad (27)$$

From Assumption [A1] and (25) we have

$$\|\Delta^{r+1}\| \leq M\|d^{r+1}\| \leq M\xi, \quad \forall\, r \geq R(\xi).$$

Therefore we have

$$\lim_{r\to\infty} \|\Delta^{r+1}\| \to 0. \quad (28)$$

Using the approximation (26), we obtain

$$\nabla f(x^r) = \nabla f(x^*) + \Delta^r d^r + Hd^r. \quad (29)$$

Plugging (29) into (24), the iteration (24) can be written as

$$\begin{bmatrix} x^{r+1} \\ \lambda^{r+1} \end{bmatrix} = \begin{bmatrix} I_N & 0_{N\times M} \\ \rho A & I_M \end{bmatrix} \begin{bmatrix} I_N - \frac{1}{\beta}\left(H+\rho A^TA\right) & -\frac{1}{\beta}A^T \\ 0_{M\times N} & I_M \end{bmatrix} \begin{bmatrix} x^r \\ \lambda^r \end{bmatrix} + \begin{bmatrix} I_N & 0_{N\times M} \\ \rho A & I_M \end{bmatrix} \begin{bmatrix} \nabla f(x^*) + \Delta^r d^r - Hx^* \\ -\rho b \end{bmatrix}. \quad (30)$$

Then the above iteration can be compactly written as

$$z^{r+1} = Q^{-1}Tz^r + Q^{-1}c^r \quad (31)$$

for some appropriately defined vectors z^{r+1}, z^r, c^r and matrices M, T which are given below

$$T := \begin{bmatrix} I_N - \frac{1}{\beta}\left(H+\rho A^TA\right) & -\frac{1}{\beta}A^T \\ 0_{M\times N} & I_M \end{bmatrix} \in \mathbb{R}^{(N+M)\times(N+M)}$$

$$Q := \begin{bmatrix} I_N & 0_{N\times M} \\ -\rho A & I_M \end{bmatrix} \in \mathbb{R}^{(N+M)\times(N+M)} \quad (32)$$

$$c^r := \begin{bmatrix} \nabla f(x^*) + \Delta^r d^r - Hx^* \\ -\rho b \end{bmatrix}, \; z := \begin{bmatrix} x \\ \lambda \end{bmatrix} \quad (33)$$

It is clear that c^r is a bounded sequence. As a direct result of Claim 2, we can show that every fixed point of the above iteration is an ss1 solution for problem (1).

Corollary 3. Suppose that Assumptions [A1]–[A5] are satisfied, and the parameters are chosen according to (67). Then every fixed point of the mapping $g(z)$ defined below, is a first-order stationary solution for problem (1)

$$g(z) := g([z_1, z_2]) = \begin{bmatrix} I_N & 0_{N\times M} \\ \rho A & I_M \end{bmatrix} \begin{bmatrix} z_1 - \frac{1}{\beta}\left(\nabla f(z_1) + A^Tz_2 + \rho A^TAz_1\right) \\ z_2 - \rho b. \end{bmatrix}.$$

To proceed, we analyze the dynamics of the system (31). The following claim is a key result that characterizes the eigenvalues for the matrix $Q^{-1}T$. We refer the readers to the appendix for detailed proof.

Claim 4. Suppose Assumptions [A1] – [A5] hold, and that

$$\beta > \sigma_{\max}(H + \rho A^TA).$$

Let (x^*, λ^*) be an ss1 solution satisfying (7), and that x^* is a strict saddle (14). Let $\sigma_i(Q^{-1}T)$ be the ith eigenvalue for matrix $Q^{-1}T$. Then $Q^{-1}T$ is invertible, and there exists a real scalar $\delta^* > 0$ which is independent of iteration index r, such that the following holds:

$$\exists\, i \in [N], \;\; \text{s.t. } \sigma_i(Q^{-1}T) = 1 + \delta^*.$$

Theorem 5. Suppose that Assumptions [A1]–[A5] hold true, and that the following parameters are chosen

$$\beta > \sigma_{\max}(\rho A^TA) + L, \quad \text{and} \quad \beta, \rho \text{ satisfy (67).} \quad (34)$$

Suppose that (x^0, λ^0) are initialized randomly. Then with probability one, the iterates $\{(x^{r+1}, \lambda^{r+1})\}$ generated by the GPDA converges to an ss2 solution (9).

Proof. We utilize the stable manifold theorem (Lee et al., 2016b; Shub, 1987). We will verify the conditions given in Theorem 7 (Lee et al., 2016b) to show that the system (31) is not stable around strict saddle points.

Step 1. We will show that the mapping $g(z)$ defined in (34) is diffeomorphism.

First, suppose there exists $w_1 = (x_1, y_1)$, $w_2 = (x_2, y_2)$ such that $g(w_1) = g(w_2)$. Using the definition of g, and the fact that the matrix $[I\ 0; -\rho A\ I]$ is invertible, we obtain $y_2 = y_1$. Using the above two results, we obtain

$$\begin{aligned} &-x_1 + \frac{1}{\beta}(\rho A^TAx_1 + \nabla f(x_1)) \\ &= -x_2 + \frac{1}{\beta}(\rho A^TAx_2 + \nabla f(x_2)). \end{aligned}$$

Then we have

$$(x_1 - x_2) = \frac{1}{\beta}\left(\nabla f(x_1) - \nabla f(x_2)\right) + \frac{\rho}{\beta}A^TA(x_1 - x_2).$$

This implies that

$$\|x_1 - x_2\| \le \left(\frac{L}{\beta} + \frac{\rho}{\beta}\sigma_{\max}(A^TA)\right)\|x_1 - x_2\|.$$

Suppose that the following is true

$$\beta > \sigma_{\max}(\rho A^TA) + L. \quad (35)$$

Then we have $x_1 = x_2$, implying $y_1 = y_2$. This says that the mapping g is injective.

To show that the mapping is surjective, we see that for a given tuple (x^{r+1}, λ^{r+1}), the iterate x^r is given by

$$\ell(x^{r+1}, \lambda^{r+1}) = -x^r + \frac{1}{\beta}(\rho A^TAx^r + \nabla f(x^r))$$

where $\ell(x^{r+1}, \lambda^{r+1})$ is some function of (λ^{r+1}, x^{r+1}). It is clear that x^r is the unique solution to the following convex problem [with β satisfying (35)]

$$x^r = \arg\min_x \frac{1}{2}\|x - \ell(x^{r+1}, \lambda^{r+1})\|^2 - \frac{1}{\beta}\left(f(x) + \frac{\rho}{2}\|Ax\|^2\right).$$

Additionally, using the definition of the mapping g in (34), we have that the Jacobian matrix for the mapping g is given by

$$\begin{aligned} Dg(z) &= \begin{bmatrix} I_N & 0_{N\times M} \\ -\rho A & I_M \end{bmatrix} \begin{bmatrix} I - \frac{1}{\beta}\left(H+\rho A^TA\right) & -\frac{1}{\beta}A^T \\ 0_{M\times N} & I_M \end{bmatrix} \\ &= Q^{-1}T. \end{aligned} \quad (36)$$

Then it has been shown in Claim 4 that as long as the following is true

$$\beta > L + \rho\sigma_{\max}(A^TA) \quad (37)$$

the Jacobian matrix $Dg(z)$ is invertible. By applying the inverse function theorem, g^{-1} is continuously differentiable.

Step 2. We can show that at a strict saddle point x^*, for the Jacobian matrix $Dg(z^*)$ evaluated at $z^* = (x^*, \lambda^*)$, the span of the eigenvectors corresponding to the eigenvalues of magnitude less than or equal to 1 is not the full space. This is easily done since according to Claim 4, $Dg(z^*) = Q^{-1}T$ has one eigenvalue that is strictly greater than 1.

Step 3. Combining the previous two steps, and by utilizing Theorem 7 (Lee et al., 2016b), we conclude that with random initialization, the GPDA converges to the second-order stationary solutions with probability one. **Q.E.D.**

3. The Gradient ADMM Algorithm

In this section, we extend the argument in the previous section to an algorithm belonging to the class of method called alternating direction method of multipliers (ADMM). Although the main idea of the analysis extends those in the previous section, the presence of *two* blocks of primal variables instead of one significantly complicates the analysis.

Consider the following problem

$$\min\ f(x)+g(y) \quad \text{s.t.} \quad Ax+By=b \tag{38}$$

where $x\in\mathbb{R}^{N_1}$, $y\in\mathbb{R}^{N_2}$ and $N_1+N_2=N$; $b\in\mathbb{R}^M$. Clearly the global consensus problem (3) can be formulated into the above two-block problem, with the following identification: $x:=\{x_1,\cdots,x_N\}$, $y:=x_0$, $f(x):=\sum_{i=1}^N f_i(x_i)$, $g(y):=g(x_0)$, $A=I_N$, $B=-I_N$, $b=0$.

For this problem, the first- and second-order necessary conditions are given by [cf. (9)]

$$\nabla f(x^*)+(\lambda^*)^T A=0,\quad \nabla g(y^*)+(\lambda^*)^T B=0, \tag{39}$$
$$z^T\begin{bmatrix}\nabla^2 f(x^*) & 0\\ 0 & \nabla^2 g(x^*)\end{bmatrix} z\succeq 0, \forall y\in\left\{z\mid \begin{bmatrix}A^TA & A^TB\\ B^TA & B^TB\end{bmatrix} z=0\right\}.$$

Similarly as before, we will refer to solutions satisfying the first line as ss1 solutions, and those that satisfy both as ss2 solutions. Therefore, a strict saddle point is defined as a point (x^*,y^*,λ^*) that satisfies the following conditions

$$\begin{aligned}&\nabla f(x^*)+(\lambda^*)^T A=0,\quad \nabla g(y^*)+(\lambda^*)^T B=0,\\ &z^T\begin{bmatrix}\nabla^2 f(x^*) & 0\\ 0 & \nabla^2 g(y^*)\end{bmatrix} z\le -\sigma\|z\|^2,\\ &\text{for some } \sigma>0,\ z \text{ satisfying } \begin{bmatrix}A^TA & A^TB\\ B^TA & B^TB\end{bmatrix} z=0.\end{aligned} \tag{40}$$

Define the AL function as

$$L(x,y;\lambda)=f(x)+g(y)+\langle\lambda,Ax+By-b\rangle+\frac{\rho}{2}\|Ax+By-b\|^2.$$

The gradient ADMM (G-ADMM) algorithm that we propose is given below.

Algorithm 2. The gradient ADMM

At iteration 0, initialize λ^0 and x^0.
At each iteration $r+1$, update variables by:

$$\begin{aligned}x^{r+1}=&\arg\min_x\ \langle\nabla f(x^r)+A^T\lambda^r\\ &+\rho A^T(Ax^r+By^r-b),x-x^r\rangle+\frac{\beta}{2}\|x-x^r\|^2\end{aligned} \tag{41a}$$

$$\begin{aligned}y^{r+1}=&\arg\min_y\ \langle\nabla g(y^r)+B^T\lambda^r\\ &+\rho B^T(Ax^{r+1}+By^r-b),y-y^r\rangle+\frac{\beta}{2}\|y-y^r\|^2\end{aligned} \tag{41b}$$

$$\lambda^{r+1}=\lambda^r+\rho\left(Ax^{r+1}+By^{r+1}-b\right). \tag{41c}$$

We note that in the GADMM, the x and y steps perform gradient steps to optimize the AL, instead of performing the exact minimization as the original convex version of ADMM does (Boyd et al., 2011; Eckstein & Bertsekas, 1992). The reason is that the direct minimization may not be possible because the non-convexity of f and g makes the subproblem of minimizing the AL w.r.t. x and y also non-convex. Note that the gradient steps have been used in the primal updates of ADMM when dealing with convex problems, see (Gao et al., 2014), but their analyses do not extend to the non-convex setting.

It is also worth noting that the key difference between Algorithm 2 and 1 is that, in the y update step (41b) of Algorithm 2, the newly updated x^{r+1} is used. If in this step x^r is used instead of x^{r+1}, then Algorithm 2 is equivalent to Algorithm 1. Also there are quite a few recent works applying ADMM-type method to solve a number of non-convex problems; see, e.g., (Li & Pong, 2014b; Max L.N. Goncalves & Monteiro, 2017; Wang & W. Yin, 2015) and the references therein. However, to the best of our knowledge, these algorithms do not take exactly the same form as Algorithm 2 described above, despite the fact that their analyses all appear to be quite similar (i.e., some potential function based on the AL is shown to be descending at each iteration of the algorithm). In particular, in (Max L.N. Goncalves & Monteiro, 2017), both the x and y subproblems are solved using a proximal point method; In (B. Jiang & Zhang, 2016), the x-step is solved using the gradient step, while the y-step is solved using the conventional exact minimization. Of course, none of these works analyzed the convergence of these methods to ss2 solutions.

3.1. Application in global consensus problem

We discuss how Algorithm 2 can be applied to solve the global consensus (3). For this problem, the distributed nodes and the master node alternate between their updates:

$$\begin{aligned}x_i^{r+1}=&\arg\min_{x_i}\ \langle\nabla f_i(x_i^r)+\lambda_i^r+\rho(x_i^r-x_0^r),x_i-x_i^r\rangle\\ &+\frac{\beta}{2}\|x_i-x_i^r\|^2,\ \forall\, i\end{aligned}$$

$$\begin{aligned}x_0^{r+1}=&\arg\min_{x_0}\ \langle\nabla g(x_0)-\sum_{i=1}^N(\lambda_i^r+\rho(x_i^{r+1}-x_0^r)),x_0-x_0^r\rangle\\ &+\frac{\beta}{2}\|x_0-x_0^r\|^2.\end{aligned}$$

Clearly, for fixed x_0, the distributed nodes are able to perform their computation completely in parallel.

3.2. Convergence to first-order stationary solutions

First we make the following assumptions.

- B1. The function $f(x)$ and $g(y)$ are smooth and both have Lipschitz continuous gradient and Hessian, with constants L_f, L_g, M_f and M_g.
- B2. $f(x)$ and $g(y)$ are lower bounded over $\mathbb{R}^N$. Without loss of generality, assume $f(x)\ge 0, g(y)\ge 0$.
- B3. $Ax+By=b$ is feasible over $x\in\text{dom}\,(f)$ and $y\in\text{dom}\,(g)$; the matrix $[A;B]\in\mathbb{R}^{M\times N}$ is *not* full rank.
- B4. $f(x)+g(y)+\frac{\rho}{2}\|Ax+By-b\|^2$ is a coercive function.

B5. $f(x)+g(x)$ is a (KŁ) function given in [A5].

Based on the above assumptions, the convergence of Algorithm 2 to the ss1 solutions can be shown following similar line of arguments as in (B. Jiang & Zhang, 2016; Li & Pong, 2014b; Max L.N. Goncalves & Monteiro, 2017; Wang & W. Yin, 2015). However, since the exact form of this algorithm has not appeared before, for completeness we provide the proof outline in the appendix.

Claim 6. Suppose Assumptions [B1] – [B5] are satisfied. For appropriate choices of β, ρ [see (82) in the Appendix for the precise expression], and starting from any point (x^0, y^0, λ^0), Algorithm 2 converges to the set of ss1 points. Further, if $L(x^{r+1}, y^{r+1}, \lambda^{r+1})$ is a KŁ function, then Algorithm 2 converges globally to a unique point (x^*, y^*, λ^*).

3.3. Convergence to ss2 solutions

The optimality conditions for the (x, y) update is given as

$$\nabla f(x^r) + A^T\lambda^r + \rho A^T(Ax^r + By^r - b) + \beta(x^{r+1} - x^r) = 0$$
$$\nabla g(y^r) + B^T\lambda^r + \rho B^T(Ax^{r+1} + By^r - b) + \beta(y^{r+1} - y^r) = 0.$$

These conditions combined with the update rule of the dual variable give the following compact form of the algorithm

$$\begin{bmatrix} x^{r+1} \\ y^{r+1} \\ \lambda^{r+1} \end{bmatrix} = \begin{bmatrix} x^r - \frac{1}{\beta}\left(\nabla f(x^r) + A^T\lambda^r + \rho A^T(Ax^r + By^r - b)\right) \\ y^r - \frac{1}{\beta}\left(\nabla g(y^r) + B^T\lambda^r + \rho B^T(Ax^{r+1} + By^r - b)\right) \\ \lambda^r + \rho\left(Ax^{r+1} + By^{r+1} - b\right) \end{bmatrix}.$$

To compactly write the iterations in the form of a linear dynamic system, define

$$z^{r+1} := [x^{r+1}; y^{r+1}; \lambda^{r+1}] \in \mathbb{R}^{2N+M}.$$

Next we approximate the iteration around a stationary solution x^*. Suppose that $\nabla^2 f(x^*) = H$ and $\nabla^2 g(y^*) = G$. Then similarly as the derivation of (30), we can write

$$Pz^{r+1} = T^r z^r + d = (T + E^r)z^r + d^r$$

where we have defined

$$P := \begin{bmatrix} I_N & 0 & 0 \\ \frac{\rho}{\beta}B^TA & I_N & 0 \\ -\rho A & -\rho B & I_M \end{bmatrix},\ E^r := \begin{bmatrix} \Delta_H^r \\ \Delta_G^r \\ 0 \end{bmatrix} \tag{43a}$$

$$d := \begin{bmatrix} \frac{\rho}{\beta}A^Tb + \nabla f(x^*) - \Delta_H^r x^* - Hx^* \\ \frac{\rho}{\beta}B^Tb + \nabla g(y^*) - \Delta_G^r x^* - Gy^* \\ -\rho b \end{bmatrix} \tag{43b}$$

$$T := \begin{bmatrix} I_N - \frac{1}{\beta}H - \frac{\rho}{\beta}A^TA & -\frac{\rho}{\beta}A^TB & -\frac{1}{\beta}A^T \\ 0 & I_N - \frac{1}{\beta}G + \frac{\rho}{\beta}B^TB & -\frac{1}{\beta}B^T \\ 0 & 0 & I_M \end{bmatrix} \tag{43c}$$

with the following

$$\Delta_H^{r+1} := \int_0^1 (\nabla^2 f(x^* + td_x^{r+1}) - H)d_x^{r+1}dt$$
$$\Delta_G^{r+1} := \int_0^1 (\nabla^2 g(y^* + td_y^{r+1}) - G)d_y^{r+1}dt,$$
$$\text{with}\quad d_x^{r+1} := -x^* + x^{r+1},\quad d_y^{r+1} := -y^* + y^{r+1}.$$

By noting that P is an invertible matrix, we conclude that the new iteration z^{r+1} can be expressed as

$$z^{r+1} = P^{-1}(T + E^{r+1})z^r + P^{-1}d^r. \tag{44}$$

Now in order to analyze the stability at a point (x^*, y^*), similarly as before we need to analyze the eigenvalues of the matrix $P^{-1}T$ at a stationary solution.

We note that P is a lower triangular matrix and $\det P = 1$. This implies that $\det(P^{-1}T - \mu I) = \det(T - \mu P)$. We have the following characterization on the determinant of $T - \mu P$; please see Appendix for detailed proof.

Claim 7. We have the following for $\det[T - \mu P]$:

1) $\det[T - P] = 0$, i.e., 1 is an eigenvalue of $P^{-1}T$.

2) Suppose that the following condition is satisfied

$$\beta > \rho\sigma_{\max}(A^TA) + L_f,\quad \beta > \rho\sigma_{\max}(B^TB) + L_g,$$

Then $\det[T] \neq 0$, i.e., the matrix $P^{-1}T$ is invertible.

3) Define a $2N \times 2N$ matrix $U(\mu) = [U_{11}(\mu)\ U_{12}(\mu); U_{12}(\mu)\ U_{22}(\mu)]$, with

$$U_{11}(\mu) = -\mu\left(2I - \frac{2\rho}{\beta}A^TA - \frac{1}{\beta}H - \mu I\right) + I - \frac{\rho}{\beta}A^TA - \frac{1}{\beta}H \tag{45a}$$

$$U_{12}(\mu) = \mu\frac{2\rho}{\beta}A^TB - \frac{\rho}{\beta}A^TB = (2\mu - 1)\frac{\rho}{\beta}A^TB \tag{45b}$$

$$U_{21}(\mu) = \mu^2\frac{\rho}{\beta}B^TA \tag{45c}$$

$$U_{22}(\mu) = -\mu\left(2I - \frac{1}{\beta}G - \frac{2\rho}{\beta}B^TB - \mu I\right) + I - \frac{1}{\beta}G - \frac{\rho}{\beta}B^TB. \tag{45d}$$

Then we have $\det[U(\mu)] = \det[T - \mu P]$, and that for any $\delta \in \mathbb{R}_+$ the eigenvalues of $U(1+\delta)$ are the same as those of the following symmetric matrix

$$\begin{bmatrix} U_{11}(1+\delta) & (\delta+1)\sqrt{2\delta+1}\frac{\rho}{\beta}A^TB \\ (\delta+1)\sqrt{2\delta+1}\frac{\rho}{\beta}B^TA & U_{22}(1+\delta). \end{bmatrix} \tag{46}$$

Based on Claim 7, we will show that the matrix $P^{-1}T$ has a *real* eigenvalue $\mu = 1 + \delta$, with $\delta > 0$ being a positive number. To this end, plugging $\mu = 1 + \delta$ to the expression of the U matrix in (45a) we have

$$U_{11}(1+\delta) = \delta^2 I + \frac{\rho}{\beta}(1+2\delta)A^TA + \frac{\delta}{\beta}H$$
$$U_{21}(1+\delta) = (1+\delta)^2\frac{\rho}{\beta}B^TA,\ U_{12}(1+\delta) = (1+2\delta)\frac{\rho}{\beta}A^TB$$
$$U_{22}(1+\delta) = \delta^2 I + \frac{\rho}{\beta}(1+2\delta)B^TB + \frac{\delta}{\beta}G.$$

Therefore, in this case we can express $U(1+\delta)$ as

$$U(1+\delta) = (2\delta+1)U(1) + \frac{\delta}{\beta}\begin{bmatrix} H & 0 \\ 0 & G \end{bmatrix} + \delta^2\begin{bmatrix} I & 0 \\ \frac{\rho}{\beta}B^TA & I \end{bmatrix}.$$

It remains to show that there exists $\delta^* > 0$ such that the determinant of the above matrix is zero. To this end, we rewrite the above expression as follows

$$U(1+\delta) = \delta\left(\frac{2\delta+1}{\delta}U(1) + \frac{1}{\beta}\begin{bmatrix} H & 0 \\ 0 & G \end{bmatrix} + \delta\begin{bmatrix} I & 0 \\ \frac{\rho}{\beta}B^TA & I \end{bmatrix}\right) := \delta\left(F(\delta)+E(\delta)\right) \quad (47)$$

where for notational simplicity, we have defined

$$F(\delta) = \frac{(2\delta+1)}{\delta}U(1) + \frac{1}{\beta}\begin{bmatrix} H & 0 \\ 0 & G \end{bmatrix},\ E(\delta) = \delta\begin{bmatrix} I & 0 \\ \frac{\rho}{\beta}B^TA & I \end{bmatrix}.$$

Note that from (40), we know that at a strict saddle point, there exists y such that

$$U(1)y = 0, \quad y^T\begin{bmatrix} H & 0 \\ 0 & G \end{bmatrix}y \le -\sigma\|y\|^2, \quad (48)$$

which implies

$$y^T\left(\gamma U(1) + \begin{bmatrix} H & 0 \\ 0 & G \end{bmatrix}\right)y \le -\sigma\|y\|^2,\ \forall\,\gamma. \quad (49)$$

This further implies that the matrix $F(\delta)$ has eigenvalue no greater than $-\sigma/\beta$ for any δ.

Next we invoke a matrix perturbation result (Stewart & Sun, 1990) to argue that the matrix $F(\delta)+E(\delta)$ also has negative eigenvalue as long as the parameter $\delta > 0$ is small enough.

For a given matrix $\widetilde{F} = F + E \in \mathbb{R}^{N\times N}$, let us define the following quantity, which is referred to as the optimal matching distance between F and $\widetilde{F}$ [see Chapter 4, Section 1, Definition 1.2 in (Stewart & Sun, 1990)]

$$\mathrm{md}(F,\widetilde{F}) := \min_{\Pi}\max_{j\in[N]}|\widetilde{\sigma}_{\Pi(j)} - \sigma_j| \quad (50)$$

where Π is taken over all permutations of $[N]$, and σ_j (resp $\widetilde{\sigma}_j$) is the jth eigenvalue of F (resp. $\widetilde{F}$). We have the following results characterizing the matching distance of two matrices F and $\widetilde{F}$ (Stewart & Sun, 1990):

Claim 8. Suppose that F is diagonalizable, i.e., $X^{-1}FX = \Upsilon$. Then the following is true

$$\mathrm{md}(F,\widetilde{F}) \le (2N-1)\|X\|\|X^{-1}\|\|E\|. \quad (51)$$

Let us apply Claim 8 to the matrices $F(\delta)$ and $F(\delta)+E(\delta)$. Note that

$$\|E\|_2 = \delta\sigma_{\max}\left(\begin{bmatrix} I & \frac{\rho}{\beta}A^TB \\ \frac{\rho}{\beta}B^TA & \frac{\rho^2}{\beta^2}B^TAA^TB + I \end{bmatrix}\right) := \delta d$$

where d is a fixed number independent of δ. By applying Claim 8, and using the fact that $\|X\| = 1$, we obtain the following

$$\mathrm{md}(F(\delta), F(\delta)+E(\delta)) \le (2N-1)\delta d. \quad (52)$$

Clearly, we can pick $\delta = \frac{\sigma}{2d\beta(2N-1)}$, which implies that

$$\mathrm{md}(F(\delta), F(\delta)+E(\delta)) \le \frac{\sigma}{2\beta}. \quad (53)$$

This combined with the fact that $F(\delta)$ has an eigenvalue smaller or equal to $-\sigma/\beta$ regardless of the choice of δ, and that all the eigenvalues of $F(\delta)+E(\delta)$ are real (cf. Claim 7), we conclude that there exists an index $i \in [N]$ such that

$$\sigma_i(F(\delta)+E(\delta)) \le -\frac{\sigma}{2\beta}. \quad (54)$$

This implies that

$$\sigma_i(U(1+\delta)) \overset{(47)}{=} \delta\sigma_i(F(\delta)+E(\delta)) \le -\frac{\sigma\delta}{2\beta} = -\frac{\sigma^2}{4\beta^2(2N-1)}.$$

In conclusion, we have the following claim.

Claim 9. There exist $\widehat{\delta} > 0$ and $\widetilde{\delta} > 0$ such that

$$\sigma_{\min}(U(1+\widehat{\delta})) < 0,\ \sigma_i(U(1+\widetilde{\delta})) > 1, \quad \forall\, i. \quad (55)$$

Proof. The first claim comes directly form our above discussion. The second claim is also easy to see by analyzing the eigenvalues for the symmetric matrix in (46), for large positive δ. **Q.E.D.**

Using the results in Claim 7 and Claim 9, and using the fact that the eigenvalues for $U(1+\delta)$ are continuous functions of δ, we conclude that there exists $\delta^* > 0$ such that $\det[U(1+\delta^*)] = 0$. The result below summarizes the proceeding discussion.

Claim 10. Suppose Assumptions [B1] –[B5] hold true. Let (x^*, y^*, λ^*) be a first-order stationary solution satisfying (7), and that it is a strict saddle point satisfying (40). Let $\sigma_i(P^{-1}T)$ be the ith eigenvalue for matrix $P^{-1}T$. Then the following holds:

$$\exists\, i \in [N],\ \text{ s.t. } |\sigma_i(P^{-1}T)| > 1. \quad (56)$$

Further, when β satisfies

$$\beta > \rho\sigma_{\max}(A^TA) + L_f,\ \beta > \rho\sigma_{\max}(B^TB) + L_g, \quad (57)$$

the matrix $P^{-1}T$ is invertible.

The rest of the proof uses a similar argument as in Theorem 5. We have the following result for the GADMM algorithm.

Theorem 11. Suppose that Assumptions [B1] – [B5] hold, and β, ρ are chosen according to (57) and (82) in the Appendix. Suppose that (x^0, y^0, λ^0) are initialized randomly. Then with probability one, the iterates generated by the GADMM converge to an ss2 solution satisfying (39).

References

Aybat, N-S. and Hamedani, E-Y. A primal-dual method for conic constrained distributed optimization problems. *Advances in Neural Information Processing Systems*, 2016.

B. Jiang, T. Lin, S. Ma and Zhang, S. Structured nonconvex and nonsmooth optimization: algorithms and iteration complexity analysis. 2016. Preprint.

Bertsekas, D. P. *Nonlinear Programming, 2nd ed*. Athena Scientific, Belmont, MA, 1999.

Bianchi, P. and Jakubowicz, J. Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization. *IEEE Transactions on Automatic Control*, 58(2):391–405, 2013.

Bolte, J., Sabach, S., and Teboulle, M. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146, 2014.

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

Chang, T.-H., Hong, M., and Wang, X. Multi-agent distributed optimization via inexact consensus ADMM. *IEEE Transactions on Signal Processing*, 63(2):482–497, Jan 2015. ISSN 1053-587X. doi: 10.1109/TSP.2014.2367458.

Conn, Andrew R, Gould, Nicholas IM, and Toint, Philippe L. *Trust region methods*. SIAM, 2000.

Eckstein, J. and Bertsekas, D. P. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1):293–318, 1992.

Forero, P. A., Cano, A., and Giannakis, G. B. Distributed clustering using wireless sensor networks. *IEEE Journal of Selected Topics in Signal Processing*, 5(4):707–724, Aug 2011. ISSN 1932-4553. doi: 10.1109/JSTSP.2011.2114324.

Gao, X., Jiang, B., and Zhang, S. On the information-adaptive variants of the admm: An iteration complexity perspective. 2014. Preprint.

Hong, M. Decomposing nonconvex problems using a proximal primal-dual approach: Algorithms, convergence and applications. 2016. Preprint, available on arXiv, arXiv:1604.00543.

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

Hong, M., Hajinezhad, D., and Zhao, M.-M. Prox-pda: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks. In *the Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

Jin, Chi, Ge, Rong, Netrapalli, Praneeth, Kakade, Sham M, and Jordan, Michael I. How to escape saddle points efficiently. In *Proceedings of International Conference on Machine Learning (ICML)*, 2017.

J.Wright, S. Implementing proximal point methods for linear programming. *Journal of Optimization Theory and Applications*, 65(3):531–554, Jun 1990.

Lee, J. D., Simchowitz, M., Jordan, M. I., and Recht, B. Gradient descent only converges to minimizers. In *Proc. of Annual Conference on Learning Theory (COLT)*, pp. 1246–1257, 2016a.

Lee, J. D., Simchowitz, M., Jordan, M. I., and Recht, B. Gradient descent converges to minimizers. 2016b. Preprint, available at arXiv:1602.04915v1.

Lee, J. D., Panageas, I., Piliouras, G., Simchowitz, M., Jordan, M. I., and Recht, B. First-order methods almost always avoid saddle points. 2017. Preprint.

Li, G. and Pong, T.-K. Splitting methods for nonconvex composite optimization. 2014a. arXiv preprint arXiv:1407.0753.

Li, G. and Pong, T.-K. Splitting methods for nonconvex composite optimization. 2014b. arXiv preprint arXiv:1407.0753.

Li, M., Andersen, D. G., and Smola, A. Distributed delayed proximal gradient methods. In *NIPS Workshop on Optimization for Machine Learning*, 2013.

Lian, Xiangru, Zhang, Ce, Zhang, Huan, Hsieh, Cho-Jui, Zhang, Wei, and Liu, Ji. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *The Proceeding of NIPS*, 2017.

Liao, W.-C., Hong, M., Farmanbar, H., and Luo, Z.-Qu. Semi-asynchronous routing for large-scale hierarchical networks. In *The Proceedings of IEEE ICASSP*, 2015.

Lorenzo, P. D. and Scutari, G. Next: In-network nonconvex optimization. 2016. Preprint.

Mateos, G., Bazerque, J. A., and Giannakis, G. B. Distributed sparse linear regression. *IEEE Transactions on Signal Processing*, 58(10):5262–5276, 2010.

Max L.N. Goncalves, Jefferson G. Melo and Monteiro, Renato D.C. Convergence rate bounds for a proximal admm with over-relaxation stepsize parameter for solving nonconvex linearly constrained problems. 2017. Preprint, available at: arXiv:1702.01850.

Murty, Katta G. and Kabadi, Santosh N. Some NP-complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39(2):117–129, Jun 1987. ISSN 1436-4646. doi: 10.1007/BF02592948. URL http://dx.doi.org/10.1007/BF02592948.

Nedic, A. and Olshevsky, A. Distributed optimization over time-varying directed graphs. *IEEE Transactions on Automatic Control*, 60(3):601–615, 2015.

Nedic, A. and Ozdaglar, A. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009. ISSN 0018-9286. doi: 10.1109/TAC.2008.2009515.

Nedić, A. and Ozdaglar, A. Subgradient methods for saddle-point problems. *Journal of Optimization Theory and Applications*, 142(1):205–228, Jul 2009. ISSN 1573-2878. doi: 10.1007/s10957-009-9522-7. URL http://dx.doi.org/10.1007/s10957-009-9522-7.

Nesterov, Yurii and Polyak, Boris T. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

Reddi, Sashank J, Zaheer, Manzil, Sra, Suvrit, Póczos, Barnabás, Bach, Francis, Salakhutdinov, Ruslan, and Smola, Alexander J. A generic approach for escaping saddle points. *arXiv:1709.01434 [cs.LG]*, 2017.

Rockafellar, R. T. Augmented lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of operations research*, 1(2):97–116, 1976.

Schizas, I., Ribeiro, A., and Giannakis, G. Consensus in ad hoc wsns with noisy links - part i: Distributed estimation of deterministic signals. *IEEE Transactions on Signal Processing*, 56(1):350 – 364, 2008.

Schizas, I., Mateos, G., and Giannakis, G. Distributed LMS for consensus-based in-network adaptive processing,. *IEEE Transactions on Signal Processing*, 57(6): 2365 – 2382, 2009.

Shalev-Shwartz, S. and Zhang, T. Proximal stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Rsearch*, 14:567–599, 2013.

Shi, W., Ling, Q., Wu, G., and Yin, W. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2014.

Shub, M. *Global Stability of Dynamical Systems*. Springer Science & Business Media, 1987.

Stewart, G. W. and Sun, J.-G. *Matrix Perturbation Theory*. Academic Press, 1990.

Uzawa, H. Iterative methods in concave programming. In *Studies in Linear and Nonlinear Programming*, pp. 154165. Stanford University Press, 1958.

Wang, Y. and W. Yin, J. Zeng. Global convergence of admm in nonconvex nonsmooth optimization. 2015. arXiv Preprint, arXiv:1511.06324.

Zhang, Y. and Lin, X. Disco: Distributed optimization for self-concordant empirical loss. In Blei, David and Bach, Francis (eds.), *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pp. 362–370. JMLR Workshop and Conference Proceedings, 2015. URL http://jmlr.org/proceedings/papers/v37/zhangb15.pdf.