Undergraduate Research Opportunity Programme in Science

ADMM-type methods for linear programming problems

Gao Yuan

Supervisor: Prof. Toh Kim Chuan

Department of Mathematics
National University of Singapore

AY2015/2016 Semester I

Abstract

We reviewed and implemented several DMM-type methods and conducted numerical experiments on large assignment problem instances. The methods are (1) classical ADMM with reformulation proposed by He and Yuan in [1], (2) Bregman ADMM proposed by Wang and Banerjee in [2] and semi-proximal ALM. In order to achieve efficient variable updates, different methods require different formulations of the input LP problem, with Bregman ADMM exploiting the special structure of the assignment problem to achieve fast updates while maintain feasibility throughout. Meanwhile, the other two algorithms work for general standard form LP problems. For implementation, sparse matrix representation, compressed matrix operations for inner loops, and C-accelerated numerical linear algebra routines are exploited to boost performance. Gurobi, a commercial optimization software, is used as reference for numerical experiments. Results of the experiments suggest that for all instances, Bregman ADMM and semi-proximal ALM converge much faster than classical ADMM with reformulation. For instances with small dimensions, Bregman ADMM marginally outperforms semi-proximal ALM; when the dimension is large (dimension of the cost matrix of the assignment problem ranging from 100 to 500), semi-proximal ALM significantly outperforms Bregman ADMM in terms of iteration time and count. Semi-proximal ALM also gives more accurate answer (as compared to the optimal objective given by Gurobi) when the terminal residual threshold for both algorithms is set to be equal. Meanwhile, classical ADMM with reformulation converges only for instances with small dimensions (≤ 15) and does not converge after a maximum number of iterations in terms of residual and objective value.

Chapter 1

Introduction

We will present a general introduction on the standard form LP problem, 2-block convex minimization problem with linear constraints and theories on sufficient conditions for optimality. Optimality conditions for the standard form LP problem and the 2-block convex minimization problem will also be derived. Chapter 2 is devoted to applying the algorithms on the standard form LP problem and deriving efficient updating formulas suitable for implementation. In addition, one subsection convergence analysis. In Chapter 3, we discuss briefly how the algorithms are implemented in Matlab. Numerical experiments are designed and conducted to test and compare the algorithms. Experiment results are presented and discussed. Chapter 4 summarizes the work done.

1.1 Standard form LP problem

Consider the standard form LP problem

$$\min\{\langle c, x\rangle \mid Ax = b, x \in \mathbb{R}^n_{\geq 0}\}, \tag{1.1}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^n$ and $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product. A wide range of real-world problems can be modeled as LP problems, while there has been extensive research on algorithms for efficiently solving LP problems. The most widely used algorithms for solving LP problems include the simplex method, its variants and the interior point methods. The alternating direction method of multipliers, which serves as a framework for efficiently solving a more general class of convex optimization problems, has been extensively researched and developed since 1970. In this paper, we investigate the applicability and efficiency of several ADMM-type methods on solving standard form LP problems.

1.2 2-block convex minimization problem and classical ADMM

In theory, the classical ADMM solves the general convex minimization problem with convex constraint sets and a linear constraint

$$\min\{f_1(\boldsymbol{x}_1) + f_2(\boldsymbol{x}_2) \mid \mathcal{A}_1\boldsymbol{x}_1 + \mathcal{A}_2\boldsymbol{x}_2 = \boldsymbol{b}, \boldsymbol{x}_1 \in \mathcal{X}_1, \boldsymbol{x}_2 \in \mathcal{X}_2\}, \tag{1.2}$$

where $\mathcal{X}_i \subseteq \mathbb{R}^{d_i}$ are nonempty closed convex sets, $\mathcal{A}_i \in \mathbb{R}^{m\times d_i}$, $\boldsymbol{b} \in \mathbb{R}^m$ and $f_i : \mathbb{R}^{d_i} \to \mathbb{R} \cup \{\infty\}$ are closed *proper convex* functions for $i = 1, 2$. We say f is *proper convex* if f is a convex function with range $\mathbb{R} \cup \{\infty\}$ such that $f(x) < \infty$ for at least one x. With a proper convex objective function and a convex feasible set (in this case, described by the linear constraint and $\mathcal{X}_1$, $\mathcal{X}_2$), (1.2) is known as a *convex problem*.

Since a convex function is always continuous and the feasible set described by the constraints is closed in $\mathbb{R}^{d_1} \times \mathbb{R}^{d_2}$, if the objective function is bounded on the feasible set, an optimal solution exists by continuity of the objective function and compactness of the feasible set. In fact this is a lot more than enough for the existence of an optimal solution, but is sufficient for our purpose for now.

Definition The *augmented Lagrangian function* of (1.2) is defined as

$$\mathcal{L}_\sigma(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{\lambda}) = f_1(\boldsymbol{x}_1) + f_2(\boldsymbol{x}_2) + \langle \boldsymbol{\lambda}, \mathcal{A}_1\boldsymbol{x}_2 + \mathcal{A}_2\boldsymbol{x}_2 - \boldsymbol{b}\rangle + \frac{\sigma}{2}||\mathcal{A}_1\boldsymbol{x}_1 + \mathcal{A}_2\boldsymbol{x}_2 - \boldsymbol{b}||^2, \tag{1.3}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^m$ is the Lagrangian multiplier and $\sigma > 0$ the penalty parameter.

Note that different authors use different sign conventions for the term $\langle \boldsymbol{\lambda}, \mathcal{A}_1\boldsymbol{x}_2 + \mathcal{A}_2\boldsymbol{x}_2 - \boldsymbol{b}\rangle$. We will stick to (1.3) for now.

Definition For any function $f : \Omega \to \mathbb{R}$, the set of minimizers of f on Ω is denoted as

$$\arg\min\{f(x) \mid x \in \Omega\} = \Big\{y \in \Omega \mid f(y) = \inf\{f(x) \mid x \in \Omega\}\Big\}.$$

One typical ADMM iteration consists of choosing an initial point $(\boldsymbol{x}_1^0, \boldsymbol{x}_2^0, \boldsymbol{\lambda}^0)$ and the updating steps[1]

$$\begin{aligned}
\boldsymbol{x}_1^{k+1} &\in \arg\min\{\mathcal{L}_\sigma(\boldsymbol{x}_1, \boldsymbol{x}_2^k, \boldsymbol{\lambda}^k) \mid \boldsymbol{x}_1 \in \mathcal{X}_1\}, && (1.4\text{a})\\
\boldsymbol{x}_2^{k+1} &\in \arg\min\{\mathcal{L}_\sigma(\boldsymbol{x}_1^{k+1}, \boldsymbol{x}_2, \boldsymbol{\lambda}^k) \mid \boldsymbol{x}_2 \in \mathcal{X}_2\}, && (1.4\text{b})\\
\boldsymbol{\lambda}^{k+1} &= \boldsymbol{\lambda}^k + \tau\sigma(\mathcal{A}_1\boldsymbol{x}_1^{k+1} + \mathcal{A}_2\boldsymbol{x}_2^{k+1} - \boldsymbol{b}), && (1.4\text{c})
\end{aligned}$$

where the step length $\tau \in (0, (1+\sqrt{5})/2)$;[5] $(\boldsymbol{x}_1^k, \boldsymbol{x}_2^k, \boldsymbol{\lambda}^k)$ are the current variables and $(\boldsymbol{x}_1^{k+1}, \boldsymbol{x}_2^{k+1}, \boldsymbol{\lambda}^{k+1})$ the updated ones after the k-th iteration. In theory, for the iteration steps (1.4a) - (1.4c) to converge to an optimal solution of (1.2), $(\boldsymbol{x}_1^0, \boldsymbol{x}_2^0)$ needs not be feasible and there is no restriction on $\boldsymbol{\lambda}^0$.

The convergence of the above scheme (1.4a) - (1.4c) with step length τ has been established in previous literature [5]. In subsection 2.2.1 we will provide the sufficient conditions for the convergence of generalized Bregman ADMM, which gives the convergence of the classical ADMM with different restrictions on the step length τ.

1.3 Duality, optimality and the KKT conditions

1.3.1 Preliminaries

We then define a general form of the KKT conditions and provide several theorems on characterization of optimal solutions via the KKT conditions, which are essential components in theoretical convergence analysis and implementation of the algorithms.

Definition Suppose U is a convex open subset of $\mathbb{R}^n$. A *subgradient* of a convex function $f : U \to \mathbb{R} \cup \{\infty\}$ at y is a vector $v \in \mathbb{R}^n$ such that

$$f(x) \geq f(y) + \langle v, x - y\rangle$$

for any $x \in U$. The *subdifferential* $\partial f(y)$ is the set of all subgradients of f at y.

Remark When f is differentiable at y, $\partial f(y) = \{\nabla f(y)\}$, the singleton set containing the gradient of f at y; if f is convex, $\partial f(y)$ is always nonempty.

Definition Given a general minimization problem

$$\min\{f(x) \mid x \in \mathbb{R}^n, h_i(x) \leq 0, i = 1, ..., m,\ l_j(x) = 0, j = 1, ..., r\}, \tag{1.5}$$

where h_i, $l_j : \mathbb{R}^n \to \mathbb{R}$, its *Lagrangian* is defined as

$$\mathcal{L}(x, u, v) = f(x) + \sum_{i=1}^{m} u_i h_i(x) + \sum_{j=1}^{r} v_i l_i(x).$$

The *Lagrangian dual function* is

$$g(u, v) = \min\{\mathcal{L}(x, u, v) \mid x \in \mathbb{R}^n\}.$$

The *dual problem* is

$$\max\{g(u, v) \mid u \in \mathbb{R}^m_{\geq 0},\ v \in \mathbb{R}^n\}. \tag{1.6}$$

Remark In general, the dual problem is always convex since g is always concave. In addition, weak duality always holds: $f^* \geq g^*$, where f^* and g^* are primal and dual optimal values respectively. Additional conditions are needed for strong duality ($f^* = g^*$) to hold.

Theorem 1.3.1 *(Slater's Condition) If the primal problem is convex and there exists x such that $h_i(x) < 0, i = 1, ..., j$ and $l_j(x) = 0, j = 1, ..., r$, then strong duality holds: $f^* = g^*$.*

Definition The Karush-Kuhn-Tucker conditions or KKT conditions for problem (1.5) are [6]

$$0 \in \partial f(x) + \sum_{i=1}^{m} u_i \partial h_i(x) + \sum_{j=1}^{r} v_j \partial l_j(x), \tag{1.7a}$$

$$u_i \cdot h_i(x) = 0, \tag{1.7b}$$

$$h_i(x) \leq 0,\ i = 1, ..., m,\quad l_j(x) = 0,\ j = 1, ..., r, \tag{1.7c}$$

$$u_i \geq 0,\ i = 1, ..., m. \tag{1.7d}$$

Remark Conditions (1.7a) - (1.7c) are frequently referred to as *stationarity*, *complementarity*, *primary feasibility* and *dual feasibility* respectively.

Theorem 1.3.2 *If x^* and (u^*, v^*) are the primal and dual optimal solutions respectively with $f(x^*) = g(u^*, v^*)$ (strong duality), then (x^*, y^*, z^*) must satisfy the KKT conditions* (1.7a) - (1.7d).

Remark In the above theorem we do not assume convexity of the problem.

With a little more effort, it can be shown that

Theorem 1.3.3 *If the problem is convex (f, h_i and l_i are convex) and (x^*, u^*, v^*) satisfies the KKT conditions* (1.7a) - (1.7d)*, then x^* is an optimal solution to the primal problem* (1.5) *and (u^*, v^*) is an optimal solution to the dual problem* (1.6) *with zero duality gap. In other words, $f(x^*) = g(u^*, v^*)$.*

Remark The tuple (x^*, u^*, v^*) is known as a *KKT point* of problem (1.5) with dual problem (1.6). For convex problems, KKT conditions are always sufficient for primal and dual optimality. However, it does *not* imply strong duality, and is in fact also a necessary condition for optimality when strong duality indeed holds.

In the following analysis, all problems we consider will be convex (convex objective functions with convex feasible set), for which KKT conditions are always sufficient for optimality.

1.3.2 Optimality conditions for the standard form LP problem

We then derive the KKT conditions for the standard form LP problem (1.1), which is obviously convex. Let $h_i(x) = -x_{(i)}$, the i-th component of x, $i = 1, ..., n$, and $l_i(x) = a_i^T x - b_i$ where a_i^T denotes the i-th row of the coefficient matrix A. By notational convention, let $y = v$ denote the dual variable and $z = u$ denote the slack variable, we obtain its Lagrangian

$$\mathcal{L}(x, y, z) = \langle c, x\rangle + \langle z, -x\rangle + \langle y, b - Ax\rangle = \langle c - z - A^T y, x\rangle + b^T y.$$

Hence

$$g(y, z) = \min\{\mathcal{L}(x, y, z) \mid x \in \mathbb{R}^n\} = b^T y$$

and the dual problem is $\{g(y, z) \mid y \in \mathbb{R}^m, z \in \mathbb{R}^n_{\geq 0}\}$, or more conventionally,

$$\min\{-b^T y \mid A^T y + z = c,\ y \in \mathbb{R}^m, z \in \mathbb{R}^n_{\geq 0}\}. \tag{1.8}$$

Note that (1.8) incorporates the stationarity condition $A^T y + z - c = 0$ in its constraints, which arises from

Proposition 1.3.4 *The KKT conditions for the standard form LP problem* (1.1) *is*

$$\begin{aligned} Ax - b &= 0, && (1.9a)\\ A^T y + z - c &= 0, && (1.9b)\\ x_{(i)} \cdot z_{(i)} &= 0,\ i = 1, ..., n && (1.9c)\\ x, z &\geq 0, && (1.9d) \end{aligned}$$

where $x_{(i)}$ and $z_{(i)}$ are the i-th components of x and z respectively, which are necessary and sufficient for optimality.

Remark For the standard form LP problem, strong duality holds when the primal problem (1.1) has a finite optimal objective $\langle c, x^*\rangle$. In this case, its dual problem (1.8) also has a finite optimal objective value $\langle b, y^*\rangle$ with $\langle c, x^*\rangle = \langle b, y^*\rangle$.

1.3.3 Optimality conditions for the 2-block convex minimization problem

We can derive a set of sufficient optimality conditions from the KKT system (1.7a) - (1.7d) for problem (1.2). Consider the problem without the convex constraint sets $\mathcal{X}_1$ and $\mathcal{X}_2$, whose Lagrangian is

$$\mathcal{L}\left(\begin{pmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{pmatrix}, \boldsymbol{\lambda}\right) = f_1(\boldsymbol{x}_1) + f_2(\boldsymbol{x}_2) + \langle \boldsymbol{\lambda}, \mathcal{A}_1\boldsymbol{x}_1 + \mathcal{A}_2\boldsymbol{x}_2 - \boldsymbol{b}\rangle,$$

where $(\boldsymbol{x}_1, \boldsymbol{x}_2)$ is viewed as one vector and there is no slack variables since the problem does not carry any inequality constraint. As such, its KKT conditions consist of only stationarity and primal feasibility conditions, namely,

$$0 \in \partial \begin{pmatrix} f_1(\boldsymbol{x}_1)\\ f_2(\boldsymbol{x}_2)\end{pmatrix} + \begin{pmatrix}\mathcal{A}_1^T\boldsymbol{\lambda}\\ \mathcal{A}_2^T\boldsymbol{\lambda}\end{pmatrix},$$
$$\mathcal{A}_1\boldsymbol{x}_1 + \mathcal{A}_2\boldsymbol{x}_2 - \boldsymbol{b} = 0.$$

The above conditions are sufficient for the optimality of the solutions to the problem without the constraints $\boldsymbol{x}_1 \in \mathcal{X}_1$ and $\boldsymbol{x}_2 \in \mathcal{X}_2$. Hence, for the original problem (1.2) with the constraints, a set of sufficient conditions for optimality might be

$$\begin{aligned} &-\mathcal{A}_1^T\boldsymbol{\lambda} \in \partial f_1(\boldsymbol{x}_1),\ \ -\mathcal{A}_2^T\boldsymbol{\lambda} \in \partial f_2(\boldsymbol{x}_2), && (1.10a)\\ &\mathcal{A}_1\boldsymbol{x}_1 + \mathcal{A}_2\boldsymbol{x}_2 - \boldsymbol{b} = 0, && (1.10b)\\ &\boldsymbol{x}_1 \in \mathcal{X}_1,\quad \boldsymbol{x}_2 \in \mathcal{X}_2. && (1.10c) \end{aligned}$$

The above conditions (1.10a) - (1.10a) will be used in the convergence analysis of the generalized Bregman ADMM and to derive residuals used for implementation.

1.4 The assignment problem

The special type of LP problems used as test cases are known as *assignment problems*, which are special cases of *transportation problems*, which is defined as

$$\min\{\langle C, X\rangle \mid X \in \mathbb{R}^{m\times n}_{\geq 0}, X\mathbf{1}_n = a, X^T\mathbf{1}_m = b\}, \tag{1.11}$$

where $a \in \mathbb{R}^m$, $b \in \mathbb{R}^n$, $C \in \mathbb{R}^{m\times n}$, $\mathbf{1}_k \in \mathbb{R}^k$ is a column vector of 1's for any $k \in \mathbb{N}$, $\langle\cdot,\cdot\rangle$ is the Euclidean inner product of two matrices of the same dimension and is defined as $\langle A, B\rangle = \mathrm{Tr}(A^T B)$. By definition we have $\langle A, B\rangle = \sum_{i=1}^m \sum_{j=1}^n A_{ij}B_{ij}$ for $A, B \in \mathbb{R}^{m\times n}$. Note that we must have $a \geq 0$, $b \geq 0$ and $\mathbf{1}_m^T a = \mathbf{1}_n^T b$ for (1.11) to be feasible. The *assignment problem* is the transportation problem with $m = n$ and $a = b = \mathbf{1}_m$. Note that (1.11) is not the most general definition of a transportation problem, but is sufficient for our purpose.

It is not difficult to rewrite (1.11) into (1.1). Consider "vectorizing" $C = (C_1, ..., C_n)$ and $X = (X_1, ..., X_n)$ as

$$c = \begin{pmatrix} C_1 \\ \vdots \\ C_n \end{pmatrix} \in \mathbb{R}^{mn}, \quad x = \begin{pmatrix} X_1 \\ \vdots \\ X_n \end{pmatrix} \in \mathbb{R}^{mn}_{\geq 0}. \tag{1.12}$$

The constraints $X\mathbf{1}_n = a$ and $X^T\mathbf{1}_m = b$ then translate to

$$\begin{pmatrix} I_m & \dots & I_m \end{pmatrix} x = a, \quad \begin{pmatrix} \mathbf{1}_m^T & & \\ & \ddots & \\ & & \mathbf{1}_m^T \end{pmatrix} x = b, \tag{1.13}$$

where both block matrices consist of I_m and $\mathbf{1}_m^T$ repeated n times respectively. To rewrite (1.11) into (1.1), we set

$$A = \begin{pmatrix} I_m & \cdots & I_m \\ \mathbf{1}_m^T & & \\ & \ddots & \\ & & \mathbf{1}_m^T \end{pmatrix} \in \{0,1\}^{(m+n)\times(mn)}, \quad \boldsymbol{b} = \begin{pmatrix} a \\ b \end{pmatrix} \in \mathbb{R}^{m+n}. \tag{1.14}$$

Note that A in (1.14) is rank deficient, namely, the first $(m + n - 1)$ rows are linearly independent and the last row is a linear combination of them. By Theorem 13.3 in [3] on a sufficient condition for total unimodularity, we see that A defined in (1.14) is totally unimodular. Theorem 13.1 in [3] then implies that all basic feasible solutions of the polyhedral $\{x : Ax = b, x \geq 0\}$ are integer vectors if A and b are defined as in (1.14) and b is a integer vector. In the case of an assignment problem where $m = n$, $a = b = \mathbf{1}_m$ and $X \geq 0$, we see that every basic feasible solution of $\{x : Ax = b, x \geq 0\}$ is a $\{0,1\}$-vector. Since the problem is bounded feasible, there exists an optimal basic feasible solution $x^* \in \{0,1\}^{mn}$. Since $\{0,1\}^{mn}$ contains finitely many vectors, for any C that is randomly generated from a uniform distribution on $\mathbb{R}^{m\times n}_{\geq 0}$, with probability 1 the optimal solution x^* is unique. Converting x^* back to X^*, with the constraints in (1.11) in mind, this implies that each row or column of X^* has exactly one nonzero entry which equals 1. In other words, there exists an optimal solution X^* that is a *permutation matrix*; for randomly generated C, the optimal solution is unique with probability 1.

The above unimodularity analysis is useful in checking the correctness of the solutions given by computer programs since we know *a priori* the existence and structure of an optimal solution, which is often *the* unique optimal solution.

Chapter 2

Algorithms

2.1 ADMM with reformulation

In [1], He and Yuan (2015) reformulate the LP problem with only inequality constraints into a form that is suitable for efficient ADMM iterations. In this section, we will extend the idea to the standard form LP problem.

We consider an equivalent form of the standard form LP problem (1.1)

$$\min \left\{ \sum_{i=1}^{m} \theta_i(x_i) \mid a_i^T x_i = b_i, x_i = \bar{x} \in \mathbb{R}^n_{\geq 0}, i = 1, 2, ..., m \right\}, \tag{2.1}$$

where $\theta_i : \mathbb{R}^n \to \mathbb{R}$, $\theta_i(x) = c^T x$, $i = 1, ..., m$. Note that the objective function in (2.1) is in fact scaled by m, the number of rows of A. We further define $X_i = \{x \in \mathbb{R}^n \mid a_i^T x = b_i\}$, $i = 1, ..., m$, $\mathcal{X}_1 = X_1 \times ... \times X_m$, $\mathcal{X}_2 = \bar{\mathcal{X}} = \mathbb{R}^n_{\geq 0}$, and

$$\mathcal{A}_1 = \begin{pmatrix} I & & \\ & \ddots & \\ & & I \end{pmatrix}, \ \mathcal{A}_2 = \begin{pmatrix} -I \\ -I \\ \vdots \\ -I \end{pmatrix}, \ \boldsymbol{b} = \boldsymbol{0}, \ \boldsymbol{x}_1 = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} \in \mathcal{X}_1, \ \boldsymbol{x}_2 = \bar{x} \in \mathcal{X}_2 = \bar{\mathcal{X}}, \ f_1 = \sum_{i=1}^{m} \theta_i, \ f_2 = 0.$$

The above specifications reformulate (1.1) into the framework (1.2). In order to derive efficient updating formulas for the standard LP problem from (1.4a) - (1.4c), we need the following simple facts in linear algebra.

Lemma 2.1.1 *For $q \in \mathbb{R}^n$, we have*

$$q + \frac{b - \langle a, q \rangle}{||a||^2} \cdot a \quad \in \quad \arg\min \left\{ ||x - q||^2 \mid x \in \mathbb{R}^n, a^T x = b \right\}, \tag{2.2}$$

$$\{q^+\} \quad = \quad \arg\min \left\{ ||x - q||^2 \mid x \in \mathbb{R}^n_{\geq 0} \right\}, \tag{2.3}$$

where $y^+ := \left(\max\{y_{(1)}, 0\}, ..., \max\{y_{(n)}, 0\}\right)^T$ *for* $y = \left(y_{(1)}, ..., y_{(n)}\right)^T \in \mathbb{R}^n$.

Proof A minimizer of $\{||x - q||^2 \mid x \in \mathbb{R}^n, a^T x = b\}$ is the projection of q onto the hyperplane $S = \{x \in \mathbb{R}^n \mid a^T x = b\}$. Let $x_0 \in S$, $a^T x_0 = b$, the projection can be expressed as

$$q - \left\langle q - x_0, \frac{a}{||a||} \right\rangle \frac{a}{||a||} = q + \frac{b - \langle a, q \rangle}{||a||^2} \cdot a,$$

which gives (2.2). For (2.3), notice that

$$||x-q||^2 = \sum_{i=1}^{n} \left(x_{(i)} - q_{(i)}\right)^2,$$

and each $\left(x_{(i)} - q_{(i)}\right)^2$ attains its minimum at $x_{(i)} = q_{(i)}$ if $q_{(i)} \geq 0$ or $x_{(i)} = 0$ if $q_{(i)} < 0$. $\square$

Let $\boldsymbol{\lambda} = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_m \end{pmatrix}$ with $\lambda_i \in \mathbb{R}^n$, $i = 1, ..., m$ be the Lagrangian multiplier. From (1.4a), we have

$$\begin{aligned} \boldsymbol{x}_1^{k+1} &\in \arg\min \left\{ \mathcal{L}_\sigma(\boldsymbol{x}_1, \bar{x}^k, \boldsymbol{\lambda}^k) \mid \boldsymbol{x}_1 \in \mathcal{X}_1 \right\} \\ &= \arg\min \left\{ \sum_{i=1}^{m} c^T x_i + \sum_{i=1}^{m} (\lambda_i^k)^T x_i + \frac{\sigma}{2} \sum_{i=1}^{m} ||x_i - \bar{x}^k||^2 \mid \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} \in \mathcal{X}_1 \right\} \end{aligned}$$

By (2.2), for $i = 1, ..., m$, we have

$$\begin{aligned} x_i^{k+1} &\in \arg\min \left\{ c^T x_i + (\lambda_i^k)^T x_i + \frac{\sigma}{2} ||x_i - \bar{x}^k||^2 \mid x_i \in X_i \right\} \\ &= \arg\min \left\{ ||x - (\bar{x}^k - \frac{\sigma}{2}(c + \lambda_i^k))||^2 \mid x \in \mathbb{R}^n, a_i^T x = b_i \right\} \\ &= \left\{ q_i^k + \frac{b_i - \langle a_i, q_i^k \rangle}{||a_i||^2} \cdot a_i \right\}, \end{aligned}$$

where $q_i^k = \bar{x}^k - \frac{1}{\sigma}(c + \lambda_i^k) \in \mathbb{R}^n$. Hence

$$\boldsymbol{x}_1^{k+1} = \begin{pmatrix} x_1^{k+1} \\ \vdots \\ x_m^{k+1} \end{pmatrix} \in \arg\min \left\{ \mathcal{L}_\sigma(\boldsymbol{x}_1, \bar{x}^k, \boldsymbol{\lambda}^k) \mid \boldsymbol{x}_1 \in \mathcal{X}_1 \right\}, \tag{2.4}$$

where

$$x_i^{k+1} = q_i^k + \frac{b_i - \langle a_i, q_i^k \rangle}{||a_i||^2} \cdot a_i. \tag{2.5}$$

From (1.4b), we have

$$\begin{aligned} \boldsymbol{x}_2^{k+1} = \bar{x}^{k+1} &\in \arg\min \left\{ \mathcal{L}_\sigma(\boldsymbol{x}_1^{k+1}, \bar{x}, \boldsymbol{\lambda}^k) \mid \bar{x} \in \bar{\mathcal{X}} \right\} \\ &= \arg\min \left\{ -\sum_{i=1}^{m} (\lambda_i^k)^T \bar{x} + \frac{\sigma}{2} \sum_{i=1}^{m} ||x_i^{k+1} - \bar{x}||^2 \mid \bar{x} \in \bar{\mathcal{X}} \right\} \\ &= \arg\min \left\{ ||\bar{x} - \frac{1}{m} \sum_{i=1}^{m} (x_i^{k+1} + \frac{1}{\sigma} \lambda_i^k)||^2 \mid \bar{x} \in \mathbb{R}^n_{\geq 0} \right\}. \end{aligned}$$

Therefore, by (2.3),

$$\boldsymbol{x}_2^{k+1} = \bar{x}^{k+1} = \left(\frac{1}{m} \sum_{i=1}^{m} (x_i^{k+1} + \frac{1}{\sigma} \lambda_i^k) \right)^+ \in \arg\min \left\{ \mathcal{L}_\sigma(\boldsymbol{x}_1^{k+1}, \boldsymbol{x}_2, \boldsymbol{\lambda}^k) \mid \boldsymbol{x}_2 \in \mathcal{X}_2 \right\}. \tag{2.6}$$

The updating formula for the Lagrange multiplier $\boldsymbol{\lambda}$ (1.4c) is

$$\lambda_i^{k+1} = \lambda_i^k + \tau\sigma(x_i^{k+1} - y^k), \; i = 1, ..., m. \tag{2.7}$$

2.2 Generalized Bregman ADMM

The (generalized) Bregman ADMM was first introduced by Wang and Banerjee in [2]. The idea is to replace the quadratic penalty term in ADMM by the more general Bregman divergence. An immediate generalization is to further allow variable-specific Bregman divergence terms in order to achieve more efficient update, which gives the genralized BADMM. In order words, ADMM and its variants (with step length $\tau = 1$), including the Bregman ADMM, are special cases of the generalized BADMM.

Definition Suppose $\Omega \in \mathbb{R}^n$ is a closed convex subset set of a Euclidean space and $\phi : \Omega \to \mathbb{R}$ is continuously differentiable and strictly convex on $\Omega \backslash \partial\Omega$, the relative interior of Ω. Define $B_\phi : \Omega \times (\Omega \backslash \partial\Omega) \to \mathbb{R}_{\geq 0}$, the Bregman divergence induced by ϕ, as

$$B_\phi(x, y) = \phi(x) - \phi(y) - \langle \nabla\phi(y), x - y \rangle, \tag{2.8}$$

where $\nabla\phi(y)$ denotes the gradient of ϕ at y.

For example, the square of the Euclidean 2-norm $B_\phi(x, y) = ||x - y||^2$ can be obtained by taking $\phi : z \mapsto ||z||^2$, and the Kullback-Leibler (KL) divergence

$$B_\phi(x, y) = \sum_{i=1}^{n} x_{(i)} \log \frac{x_{(i)}}{y_{(i)}} - \sum_{i=1}^{m} x_{(i)} + \sum_{i=1}^{m} y_{(i)} \tag{2.9}$$

can be obtained by taking $\phi : z \mapsto \sum_{i=1}^n z_{(i)} \log z_{(i)} - \sum_{i=1}^n z_{(i)}$. Note that in both cases ϕ is continuously differentiable and strictly convex on $\mathbb{R}^n_{>0}$ (since their Hessians are well-defined and positive definite). An important property of the Bregman divergence is $B_\phi(x, y) \geq 0$ where the equality holds if and only if $x = y$. This is because ϕ is strictly convex.

Under the framework (1.2), the generalized BADMM consists of the 3 updating steps

$$\begin{aligned}
\boldsymbol{x}_1^{k+1} &\in \arg\min\{f_1(\boldsymbol{x}_1) + \langle \boldsymbol{\lambda}^k, \mathcal{A}_1\boldsymbol{x}_1 + \mathcal{A}_2\boldsymbol{x}_2^k - \boldsymbol{b} \rangle \\
&+ \sigma B_\phi(\boldsymbol{b} - \mathcal{A}_1\boldsymbol{x}_1, \mathcal{A}_2\boldsymbol{x}_2^k) + \sigma_1 B_{\phi_1}(\boldsymbol{x}_1, \boldsymbol{x}_2^k) \mid \boldsymbol{x}_1 \in \mathcal{X}_1\}, && (2.10a) \\
\boldsymbol{x}_2^{k+1} &\in \arg\min\{f_2(\boldsymbol{x}_2) + \langle \boldsymbol{\lambda}^k, \mathcal{A}_1\boldsymbol{x}_1^{k+1} + \mathcal{A}_2\boldsymbol{x}_2 - \boldsymbol{b} \rangle \\
&+ \sigma B_\phi(\boldsymbol{b} - \mathcal{A}_1\boldsymbol{x}_1^{k+1}, \mathcal{A}_2\boldsymbol{x}_2) + \sigma_2 B_{\phi_2}(\boldsymbol{x}_1^{k+1}, \boldsymbol{x}_2) \mid \boldsymbol{x}_2 \in \mathcal{X}_2\}, && (2.10b) \\
\boldsymbol{\lambda}^{k+1} &= \boldsymbol{\lambda}^k + \tau\sigma(\mathcal{A}_1\boldsymbol{x}_1^{k+1} + \mathcal{A}_2\boldsymbol{x}_2^{k+1} - \boldsymbol{b}), && (2.10c)
\end{aligned}$$

where ϕ, ϕ_1 and ϕ_2 are continuously differentiable and strictly convex functions on some convex domains depending on the problem, $\sigma > 0$, $\tau > 0$, $\sigma_1 \geq 0$, $\sigma_2 \geq 0$.

The ordinary Bregman ADMM is obtained by setting $\sigma_1 = \sigma_2 = 0$ in (2.10a) - (2.10c) and the classical ADMM (1.4a) - (1.4c) can be obtained by further setting $\phi : z \mapsto \frac{1}{2}||z||$, which gives $B_\phi(x, y) = \frac{1}{2}||x - y||^2$.

2.2.1 Convergence analysis

The following discussion originates from [2] and establishes the global convergence of the generalized BADMM, which yields the convergence of Bregman ADMM and the classical ADMM.

Referring to problem (1.2) and the iterative steps (2.10a) - (2.10c), in additions to the properties of f_1, f_2, $\mathcal{X}_1$ and $\mathcal{X}_2$ specified by (1.2), for a chosen ϕ, additional technical conditions on σ, σ_1, σ_2 and τ are needed for (2.10a) - (2.10c) to generate a sequence that converges to an optimal solution.

Let $(\boldsymbol{x}_1^*, \boldsymbol{x}_2^*, \boldsymbol{\lambda}^*)$ satisfies the optimality conditions (1.10a) - (1.10c). Consider the subproblems (2.10a) and (2.10b) (finding $\boldsymbol{x}_1^{k+1}$ and $\boldsymbol{x}_2^{k+1}$ as minimizers of two modified augmented Lagrangian functions with Bregman terms). For (2.10a), the optimality conditions are

$$\begin{aligned}
\boldsymbol{x}_1^{k+1} &\in \mathcal{X}_1, \\
0 &\in \partial f_1(\boldsymbol{x}_1^{k+1}) + \mathcal{A}_1^T \boldsymbol{\lambda} + \sigma \nabla_{\boldsymbol{x}_1} B_\phi(\boldsymbol{b} - \mathcal{A}_1 \boldsymbol{x}_1^{k+1}, \mathcal{A}_2 \boldsymbol{x}_2^k) + \sigma_1 \nabla_{\boldsymbol{x}_1} B_{\phi_1}(\boldsymbol{x}_1^{k+1}, \boldsymbol{x}_2^k),
\end{aligned}$$

which can be written as

$$\boldsymbol{x}_1^{k+1} \in \mathcal{X}_1, \;\; -\mathcal{A}_1^T \boldsymbol{\lambda}^k - \sigma \mathcal{A}_1^T \left(\nabla\phi(\boldsymbol{b} - \mathcal{A}_1 \boldsymbol{x}_1^{k+1}) - \nabla\phi(\mathcal{A}_2 \boldsymbol{x}_2^k) \right) - \sigma_1 \left(\nabla\phi_1(\boldsymbol{x}_1^{k+1}) - \nabla\phi_1(\boldsymbol{x}_1^k) \right) \in \partial f_1(\boldsymbol{x}_1^{k+1}). \tag{2.11}$$

Note that we have used $\nabla_x B_\phi(x, y) = \nabla\phi(x) - \nabla\phi(y)$. Similarly, for (2.10b), the optimality conditions are

$$\boldsymbol{x}_2^{k+1} \in \mathcal{X}_2, \;\; -\mathcal{A}_2^T \boldsymbol{\lambda}^k - \sigma \mathcal{A}_2^T \left(\nabla\phi(\mathcal{A}_2 \boldsymbol{x}_2^{k+1}) - \nabla\phi(\boldsymbol{b} - \mathcal{A}_1 \boldsymbol{x}_1^{k+1}) \right) - \sigma_2 \left(\nabla\phi_2(\boldsymbol{x}_2^{k+1}) - \nabla\phi_2(\boldsymbol{x}_2^k) \right) \in \partial f_2(\boldsymbol{x}_2^{k+1}). \tag{2.12}$$

Lemma 2.2.1 *If $(\boldsymbol{x}_1^k, \boldsymbol{x}_2^k, \boldsymbol{\lambda}^k)$ and $(\boldsymbol{x}_1^{k+1}, \boldsymbol{x}_2^{k+1}, \boldsymbol{\lambda}^{k+1})$ satisfy*

$$B_{\phi_1}(\boldsymbol{x}_1^{k+1}, \boldsymbol{x}_1^k) = 0 \;\; B_{\phi_2}(\boldsymbol{x}_2^{k+1}, \boldsymbol{x}_2^k) = 0, \tag{2.13}$$

$$\mathcal{A}_2(\boldsymbol{x}_2^{k+1} - \boldsymbol{x}_2^k) = 0, \;\; \mathcal{A}_1 \boldsymbol{x}_1^{k+1} + \mathcal{A}_2 \boldsymbol{x}_2^{k+1} - \boldsymbol{b} = 0, \tag{2.14}$$

$$\boldsymbol{x}_1 \in \mathcal{X}_1, \;\; \boldsymbol{x}_2 \in \mathcal{X}_2, \tag{2.15}$$

then $(\boldsymbol{x}_1^k, \boldsymbol{x}_2^k, \boldsymbol{\lambda}^k) = (\boldsymbol{x}_1^{k+1}, \boldsymbol{x}_2^{k+1}, \boldsymbol{\lambda}^{k+1})$ is a KKT point.

Proof First, (2.18c) and the second equation in (2.14) imply that $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k$. We will show that $(\boldsymbol{x}_1^k, \boldsymbol{x}_2^k, \boldsymbol{\lambda}^k)$ satisfies (1.10a) - (1.10c). In fact, the two equations in(2.14) imply that $\mathcal{A}_1 \boldsymbol{x}_1^{k+1} + \mathcal{A}_2 \boldsymbol{x}_2^k = \boldsymbol{b}$, while (2.13) implies $\boldsymbol{x}_1^{k+1} = \boldsymbol{x}_1^k$. Hence by (2.12), we have

$$-\mathcal{A}_1 \boldsymbol{\lambda}^k \in \partial f_1(\boldsymbol{x}_1^k).$$

Similarly, (2.13) implies $\boldsymbol{x}_2^{k+1} = \boldsymbol{x}_2^k$, and by (2.12) we have

$$-\mathcal{A}_2 \boldsymbol{\lambda}^k \in \partial f_1(\boldsymbol{x}_2^k).$$

□

Note that in the case of Bregman ADMM with $\sigma_1 = \sigma_2 = 0$, (2.14) and (2.15) serve as sufficient optimality conditions for optimality for the general problem (1.2).

Definition (See [2]) The residual of the optimality condition (2.13) and (2.14) is defined as

$$R(k) = B_\phi(\boldsymbol{b} - \mathcal{A}_1 \boldsymbol{x}_1^{k+1}, \mathcal{A}_2 \boldsymbol{x}_2^k) + \frac{\sigma_1}{\sigma} B_{\phi_1}(\boldsymbol{x}_1^{k+1}, \boldsymbol{x}_1^k) + \frac{\sigma_2}{\sigma} B_{\phi_2}(\boldsymbol{x}_2^{k+1}, \boldsymbol{x}_2^k) + \gamma ||\mathcal{A}_1 \boldsymbol{x}_1^{k+1} + \mathcal{A}_2 \boldsymbol{x}_2^{k+1} - \boldsymbol{b}||, \tag{2.16}$$

where $\gamma > 0$.

If $R(k) = 0$ and $\boldsymbol{x}_1^k \in \mathcal{X}_1$, $\boldsymbol{x}_2^k \in \mathcal{X}_2$, (2.13) - (2.15) are satisfied and an optimal solution is obtained. Hence, in order to establish convergence of the generalized BADMM, it is sufficient to show $R(k)$ converges to 0 under certain conditions on the parameters. The following theorem in [2] summarizes the above discussion.

Theorem 2.2.2 *Suppose (i) an optimal solution of* (1.2) *exists and (ii) the Bregman divergence* B_ϕ *is defined on* $\phi : \Omega \to \mathbb{R}$ *which is* α*-strongly convex with respect to a* p*-norm. In other words, there exists* $\alpha > 0$ *such that for any* $x \in \Omega$, $y \in \Omega\backslash\Omega$,

$$B_\phi(x, y) \geq \frac{\alpha}{2}||x - y||_p^2. \tag{2.17}$$

Let the sequence $(\boldsymbol{x}_1^k, \boldsymbol{x}_2^k, \boldsymbol{\lambda}^k)$ *be generated by the generalized BADMM iterative steps* (2.10a) - (2.10c)*. Let* $(\boldsymbol{x}_1^*, \boldsymbol{x}_2^*, \boldsymbol{\lambda}^*)$ *be a KKT point. If* $\delta = \min\{1, m^{\frac{2}{p}-1}\}$, $\tau \leq (\alpha\delta - 2\gamma)$ *and* $0 < \gamma < \frac{\alpha\delta}{2}$ *for some* $\gamma > 0$*, then* $R(k) \to 0$ *and* $(\boldsymbol{x}_1^k, \boldsymbol{x}_2^k, \boldsymbol{\lambda}^k) \to (\boldsymbol{x}_1^*, \boldsymbol{x}_2^*, \boldsymbol{\lambda}^*)$ *as* $k \to \infty$.

Remark When $p \in (0, 2]$, we have $\delta = 1$, and $\tau \leq \alpha - 2\gamma$. In fact, for $\phi(x) = \frac{1}{2}||x||^2$ and $B_\phi(x, y) = \frac{1}{2}||x - y||^2$, (2.17) holds with $\alpha = 1$ and $p = 2$; for B_ϕ being the KL divergence, we have (2.17) holds with $\alpha = p = 1$ in the interior of the unit simplex.

2.2.2 Bregman ADMM for the assignment problem

We then derive the updating formulas for the transportation problem (1.11) from (2.10a) - (2.10c). Instead of converting (1.11) into (1.1), we will convert its original matrix form into (1.2). In fact, this can be done by setting

$$\begin{aligned}
&\boldsymbol{x}_1 = X \in \mathcal{X} = \{X \in \mathbb{R}_{\geq 0}^{m \times n} \mid X\mathbf{1}_n = a\}, \\
&\boldsymbol{x}_2 = Z \in \mathcal{Z} = \{Z \in \mathbb{R}_{\geq 0}^{m \times n} \mid X^T\mathbf{1}_m = b\}, \\
&\boldsymbol{\lambda} = Y, \\
&\mathcal{A}_1 = I, \ \mathcal{A}_2 = -I, \boldsymbol{b} = 0, \\
&f_1(X) = \langle C, X\rangle, \ f_2(X) = 0.
\end{aligned}$$

Again, to have a feasible problem, it is necessary that $a \geq 0$, $b \geq 0$ and $\mathbf{1}_m^T a = \mathbf{1}_n^T b$. Let $\sigma > 0$, $\sigma_1 = \sigma_2 = 0$ and $B_\phi = B_{KL} : (A, B) \mapsto \sum_{i=1}^m \sum_{j=1}^n (A_{ij} \log \frac{A_{ij}}{B_{ij}} - A_{ij} + B_{ij})$, the KL divergence (2.9). By (2.10a) - (2.10c), one iteration consists of

$$\begin{aligned}
X^{k+1} &\in \arg\min\{\langle C, X\rangle + \langle Y^{k+1}, X\rangle + \sigma B_{KL}(X, Z^k) \mid X \in \mathcal{X}\}, &(2.18\text{a})\\
Z^{k+1} &\in \arg\min\{\langle Y^k, -Z\rangle + \sigma B_{KL}(Z, X^{k+1}) \mid Z \in \mathcal{Z}\}, &(2.18\text{b})\\
Y^{k+1} &= Y^k + \frac{1}{2}\sigma(X^{k+1} - Z^{k+1}). &(2.18\text{c})
\end{aligned}$$

Note that we set the step length $\tau = 1/2$ in (2.18c), where we take $\gamma = 1/4$ and $\alpha = p = 1$ in Theorem 2.2.2. Setting $\alpha = p = 1$ makes (2.17) hold in the interiors of $\mathcal{X}$ and $\mathcal{Z}$.

Proposition 2.2.3 *The updating formulas* (2.18a) *and* (2.18b) *both have closed from solutions*

$$\begin{aligned}
X_{ij}^{k+1} &= \frac{Z_{ij}^k \exp(-\frac{C_{ij}+Y_{ij}^k}{\sigma})}{\sum_{j=1}^n Z_{ij}^k \exp(-\frac{C_{ij}+Y_{ij}^k}{\sigma})} a_i, &(2.19\text{a})\\
Z_{ij}^{k+1} &= \frac{X_{ij}^{k+1} \exp(\frac{Y_{ij}^k}{\sigma})}{\sum_{i=1}^m X_{ij}^{k+1} \exp(\frac{Y_{ij}^k}{\sigma})} b_j. &(2.19\text{b})
\end{aligned}$$

Proof We show that X^{k+1} given by (2.19a) satisfies (2.18a). Denote

$$h(X) = \langle C, X\rangle + \langle Y^{k+1}, X\rangle + \sigma B_{KL}(X, Z^k).$$

Notice that

$$h(X) = \sum_{i=1}^{m} h_i(X_{i1}, \cdots, X_{in}),$$

where

$$h_i(X_{i1}, \cdots, X_{in}) = \sum_{j=1}^{n} \left((C_{ij} + Y_{ij}^{k+1}) X_{ij} + \sigma (X_{ij} \log \frac{X_{ij}}{Z_{ij}^k} - X_{ij} + Z_{ij}^k) \right).$$

Let

$$g_i(X_{i1}, \cdots, X_{in}) = \sum_{j=1}^{n} X_{ij} - a_i.$$

The problem is reduced to minimizing $h_i(X_{i1}, \cdots, X_{in})$ subject to the constraint $g_i(X_{i1}, \cdots, X_{in}) = 0$ for each i. Consider the Lagrangian

$$\mathcal{L}_i(X_{i1}, \cdots, X_{in}, \lambda) = h_i(X_{i1}, \cdots, X_{in}) + \lambda g_i(X_{i1}, \cdots, X_{in}).$$

Since h_i is strictly convex (positive definite Hessian on $\mathbb{R}^n_{>0}$), a possible set of optimality conditions for the i-th subproblem is

$$g_i(X_{i1}, \cdots, X_{in}) = 0,$$

$$\frac{\partial}{\partial X_{ij}} \mathcal{L}_i(X_{i1}, \cdots, X_{in}, \lambda) = 0, \quad j = 1, ..., n$$

and

$$\frac{\partial}{\partial \lambda} \mathcal{L}_h(X_{i1}, \cdots, X_{in}, \lambda) = 0.$$

In other words,

$$\begin{aligned} \left(C_{ij} + Y_{ij}^{k+1}\right) + \sigma \left(\log X_{ij}^{k+1} - \log Z_{ij}^k\right) + \lambda &= 0, \;\; j = 1, ..., n \quad (2.20) \\ X_{i1}^{k+1} + \cdots + X_{in}^{k+1} &= a_i. \quad (2.21) \end{aligned}$$

From (2.20) we have

$$X_{ij} = \exp(-\lambda^*) \cdot Z_{ij}^{k+1} \exp\left(-\frac{C_{ij} + Y_{ij}^k}{\sigma}\right) \quad (2.22)$$

Substitute (2.22) into (2.21) we obtain

$$\exp(-\lambda) = \frac{a_i}{\sum_{j=1}^n Z_{ij}^k \exp\left(-\frac{C_{ij} + Y_{ij}^k}{\sigma}\right)}.$$

Hence (2.19a) is the solution to (2.20) and solves (2.18a). A similar argument proves (2.19b). □

One desirable property of the above iterative scheme (2.18a) - (2.18c) is that the feasibility of X^k and Z^k is maintained throughout, as long as we choose $X^0 \in \mathcal{X}$ and $Z^0 \in \mathcal{Z}$ such that $X^0, Z^0 > 0$, which is possible when $a, b > 0$.

2.2.3 ADMM for the assignment problem in matrix form

Note that the classical ADMM is a special case of the generalized BADMM with $\sigma_1 = \sigma_2 = 0$ and B_ϕ being the squared Euclidean norm. As such, the assignment problem (1.11) can be written in the form for classical ADMM: $\min\{\langle C, X\rangle \mid X - Z = 0, X \in \mathcal{X}, Z \in \mathcal{Z}\}$. The updating formulas are therefore, by (1.4a) - (1.4c) or (2.18a) - (2.18c),

$$\begin{aligned}
X^{k+1} &\in \arg\min\{\langle C, X\rangle + \langle Y^k, X\rangle + \frac{\sigma}{2}||X - Z^k||^2 \mid X \in \mathcal{X}\}, && (2.23a)\\
Z^{k+1} &\in \arg\min\{\langle Y^k, -Z\rangle + \frac{\sigma}{2}||X^{k+1} - Z||^2 \mid Z \in \mathcal{Z}\}, && (2.23b)\\
Y^{k+1} &= Y^k + \tau\sigma(X^{k+1} - Z^{k+1}), && (2.23c)
\end{aligned}$$

where $\tau \in (0, (1+\sqrt{5})/2)$ is the step length. Using (2.2), (2.23a) translates to for every i,

$$\begin{aligned}
(X_{i1}^{k+1}, \cdots, X_{in}^{k+1}) &\in \arg\min\left\{\sum_{j=1}^{n}\left((C_{ij} + Y_{ij}^k)X_{ij} + \frac{\sigma}{2}(X_{ij} - Z_{ij}^k)^2\right) \mid X_{i1} + \cdots + X_{in} = a_i, X_{ij} \geq 0\right\}\\
&= \arg\min\left\{\sum_{j=1}^{n}\left(X_{ij} - (Z_{ij}^k - \frac{1}{\sigma}\left(C_{ij} + Y_{ij}^k\right))\right)^2 \mid X_{i1} + \cdots + X_{in} = a_i, X_{ij} \geq 0\right\}.
\end{aligned} \tag{2.24}$$

Solving (2.24) is equivalent to finding the "projection" of a vector in $\mathbb{R}^n$ onto a simplex. An efficient method in $O(n)$ time (where n is the dimension of the Euclidean space) is proposed in [4]. It has been shown in [2] that this iterative scheme (2.23a) - (2.23c) converges slower than Gurobi and the BADMM when solving assignment problems and does not converge after a maximum number of iterations when the dimension of input instances is large.

2.3 Semi-proximal augmented Lagrangian method

We will content ourselves by considering the special case of the general semi-proximal ALM for standard form LP problems. Consider the dual of the standard form LP (1.8) and its augmented Lagrangian function

$$\mathcal{L}_\sigma(y, z, x) = -b^T + \langle x, A^T y + z - c\rangle + \frac{\sigma}{2}||A^T y + z - c||^2. \tag{2.25}$$

The semi-proximal augmented Lagrangian method consists of the following 4 updating steps

$$\begin{aligned}
\bar{y}^{k+1} &\in \arg\min\{\mathcal{L}_\sigma(y, z^k, x^k) \mid y \in \mathbb{R}^m\}, && (2.26a)\\
z^{k+1} &\in \arg\min\{\mathcal{L}_\sigma(\bar{y}^{k+1}, z, x^k) \mid z \in \mathbb{R}^n_{\geq 0}\}, && (2.26b)\\
y^{k+1} &\in \arg\min\{\mathcal{L}_\sigma(y, z^{k+1}, x^k) \mid y \in \mathbb{R}^m\}, && (2.26c)\\
x^{k+1} &= x^k + \tau\sigma(A^T y^{k+1} + z^{k+1} - c), && (2.26d)
\end{aligned}$$

where in order to establish convergence we need to set $\tau \in (0, 2)$.

We show that (2.26a) - (2.26c) all have closed form solutions. For (2.26a), $\mathcal{L}_\sigma(y, z^k, x^k)$ is a quadratic function of $y \in \mathbb{R}^m$. Since there is no constraint on y, a possible optimality condition is

$$\nabla_y \mathcal{L}_\sigma(\bar{y}^{k+1}, z^k, x^k) = 0,$$

which gives

$$-b + Ax^k + \sigma A(A^T \bar{y}^{k+1} + z^k - c) = 0,$$

or

$$AA^T \bar{y}^{k+1} = \frac{1}{\sigma}(b - Ax^k) - A(z^k - c), \tag{2.27}$$

which has a unique solution when $\mathrm{rank}(A) = m$. For (2.26b), notice that

$$\begin{aligned} & \arg\min\{\mathcal{L}_\sigma(\bar{y}^{k+1}, z, x^k) \mid z \in \mathbb{R}^n_{\geq 0}\} \\ = \quad & \arg\min\left\{(x^k)^T z + \frac{\sigma}{2}||A^T\bar{y}^{k+1} + z - c||^2 \mid z \in \mathbb{R}^n_{\geq 0}\right\} \\ = \quad & \arg\min\left\{||z - (c - A^T\bar{y}^{k+1} - \frac{1}{\sigma}x^k)|| \mid z \in \mathbb{R}^n_{\geq 0}\right\}. \end{aligned}$$

By (2.3), we have

$$z^{k+1} = \left(c - A^T\bar{y}^{k+1} - \frac{1}{\sigma}x^k\right)^+ \in \arg\min\{\mathcal{L}_\sigma(\bar{y}^{k+1}, z, x^k) \mid z \in \mathbb{R}^n_{\geq 0}\}. \tag{2.28}$$

Similar to the derivation of (2.27),

$$AA^T y^{k+1} = \frac{1}{\sigma}(b - Ax^k) - A(z^{k+1} - c). \tag{2.29}$$

Chapter 3

Numerical experiments and results

We tested the above algorithms with their respective problem formulations in Matlab. Gurobi, a commercial optimization software, is used to verify the optimality of answers and as a performance benchmark.

Unlike the simplex method which, in theory, gives an exact optimal basic feasible solution to a bounded feasible LP problem, the ADMM-type methods only guarantee, in theory, a sequence of points that converge to an optimal solution. As such, for implementation of the algorithms and numerical experiments, we need to set terminal conditions. Note that all algorithms of interest consist of initialization and iteratively applying a set of updating formulas. Hence, we may set the terminal conditions as either (a) the total number of iteration exceeds a predetermined threshold or (b) the residual of the optimality conditions is sufficiently small.

3.1 Residuals

In this section, we derive formulas for residuals mentioned above from both theoretical and computational perspectives.

Definition In the context of the standard form LP problem (1.1) with dual problem (1.8), consider its KKT conditions (1.9a) - (1.9d), for $\left\{(x^k, y^{k+1}, z^k)\right\}_{k\geq 0}$, define

$$R_{LP}(k) = \max\left\{\frac{||Ax^k - b||}{1+||b||}, \frac{||A^T y^k + z^k - c||}{1+||c||}\right\}. \tag{3.1}$$

The denominators are included to enhance numerical stability in computation: to normalize the magnitudes of the norms in the numerators and make the program less sensitive to scaling of input data. Since the standard form LP problem is convex and strong duality holds if the problem is feasible and bounded, x^* solves the primal problem and (y^*, z^*) solves the dual problem (with the same optimal objective value) if and only if (x^*, y^*, z^*) is a KKT point. By the continuity of $R_{LP}(k)$ in (x^k, y^k, z^k), we have

Proposition 3.1.1 *for any sequence (x^k, y^k, z^k) that converges to a KKT point, we have $R_{LP}(k) \to 0$. Conversely, if (x^k, y^k, z^k) converges to (x^*, y^*, z^*), $R_{LP}(k) \to 0$ and for each k, we have $x^k \geq 0$, $z^{k+1} \geq 0$ and $(z^k)_{(i)} \cdot (x^k)_{(i)} \to 0$ for any $i = 1, ..., n$ (the dual feasibility and complementarity conditions), then (x^*, y^*, z^*) is a KKT point.*

For semi-proximal augmented Lagrangian method, $R_{LP}(k)$ can be easily computed since the tuple (x^k, y^k, z^k) is maintained during each iteration. For the classical ADMM with reformulation, since the dual and slack variables are not maintained, we need an alternative but similar definition of residual, which involves constructing the slack variable z from primal, dual feasibility and complementarity.

Definition In the context of Section 2.1, let $\eta^k \in \mathbb{R}^n$ with

$$(\eta^k)_{(i)} = \frac{b_i - a_i^T q_i}{a_i^T a_i},$$

where $q_i^k = \bar{x}^k - \frac{1}{\sigma}(c + \lambda_i^k) \in \mathbb{R}^n$. Define the residual as

$$R_{LP}^{ADMM}(k) = \max\left\{\frac{||A\bar{x}^k - b||}{1 + ||b||}, \frac{||A^T y^k + z^{k+1} - c||}{1 + ||c||}\right\}, \tag{3.2}$$

where $\bar{x}^k$ is maintained in every iteration, $y^k = \sigma\eta^k/m$ and $z^{k+1} = \sigma(\bar{x}^{k+1} - \bar{x}^k) + c - A^T y$ are the modified dual and slack variables respectively.

Note that from the above definition we have $A^T y^k + z^{k+1} - c = \sigma(\bar{x}^{k+1} - \bar{x}^k)$, which is consistent with (2.14) and (2.15). In other words, $R_{LP}^{ADMM}(k) = 0$ implies that a solution to the optimality conditions (2.14) - (2.15) is obtained.

Proposition 3.1.2 *If a sequence $(\boldsymbol{x}_1^k, \bar{x}^k, \boldsymbol{\lambda}^k)$ generated by ADMM with reformulation* (2.4) - (2.7) *converges to a point $(\boldsymbol{x}_1^*, \bar{x}^*, \boldsymbol{\lambda}^*)$ with $R_{LP}^{ADMM}(k) \to 0$, then we have $(\boldsymbol{x}_1^*, \bar{x}^*, \boldsymbol{\lambda}^*)$ is a KKT point of the problem.*

For the Bregman ADMM, we will use the residual defined in (2.16), which is a generalization of the residual of the optimality condition for the 2-block convex minimization problem with linear constraints. In the case of KL divergence and the assignment problem, we have

$$R_{BADMM}(k) = B_{KL}(X^{k+1}, Z^k) + \frac{1}{4}||X^k - Z^k||^2.$$

By (2.13) - (2.15), the definition of the residual (2.16), it is clear that

Proposition 3.1.3 *If a sequence (X^k, Y^k, Z^k) generated by Bregman ADMM* (2.18a) - (2.18c) *converges to (X^*, Y^*, Z^*), with $X^* \in \mathcal{X}$, $Z^* \in \mathcal{Z}$ and $R_{BADMM}(k) \to 0$, then (X^k, Y^k, Z^k) converges to a KKT point of the problem.*

3.2 Implementation and numerical experiments

We discuss briefly how the algorithms are implemented from a computational perspective. For all algorithms, the termination condition is set to be either (a) the total number of iterations exceeds 10000 or (b) the residual is less than 10^{-5}. Note that for experiment purpose, we let each program calculates the residual during every iteration, which is time-consuming and may not be necessary in practice.

For ADMM with reformulation, we set $\tau = 1.618$. The vector $\boldsymbol{x}_1^k$, $\boldsymbol{\lambda}^k$ are stored as matrices of dimension $n \times m$, with each column representing $x_i, \lambda_i \in \mathbb{R}^n$ respectively. In this way, updating all x_i's can be done in efficient matrix operations without any inner loop.

For Bregman ADMM, with careful rearrangement, the updates can be done efficiently in compressed matrix operations. There is no inner loop inside the main loop.

For semi-proximal ALM, we set $\tau = 1.9$. The Cholesky factorization of AA^T is computed before the main loop for efficiently solving the linear systems during each iteration. In addition, several external mex files for efficient numerical linear algebra routines such as matrix-vector multiplication are called.

3.2.1 Running statistics

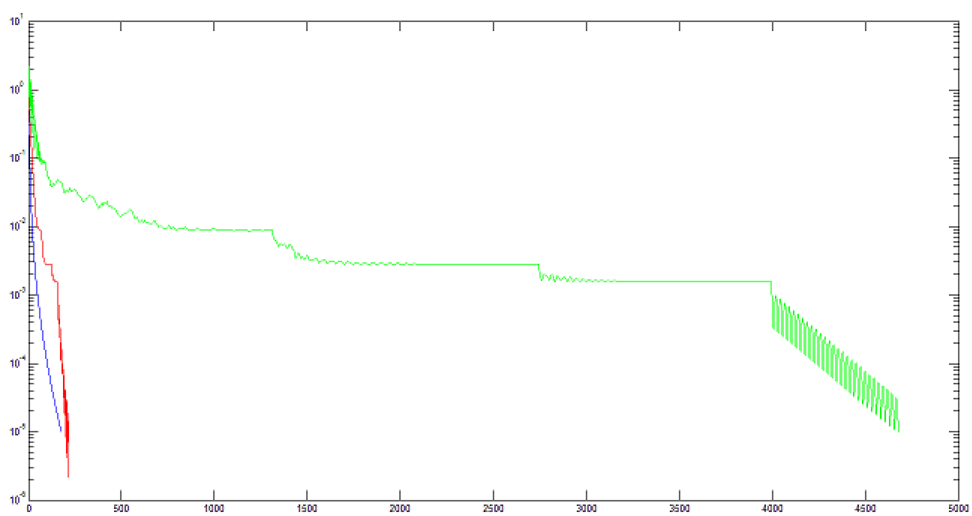
Below is the running statistics for the 3 algorithms as input dimension ranges from $m = n = 10$ to $m = n = 500$. We omit the execution of ADMM with reformulation when $m = n \geq 100$ since the patterns suggest that it does not converge after the maximum number of iterations. For each input dimension the main program generates the input data and calls the 3 algorithms (as Matlab functions). See Appendix A for the programs.

Problem dimension ($m = n$)	Gurobi optimal objective	BADMM opt_obj | it_count | time	SPALM opt_obj | it_count | time	ADMM LP reformulation opt_obj | it_count | time
10	0.1337202778	0.133793 | 176 | 0.05	0.133676 | 216 | 0.03	0.133721 | 4676 | 0.78
20	0.1608133806	0.161327 | 520 | 0.13	0.160761 | 444 | 0.08	0.162288 | 10000 | 4.22
30	0.1856271350	0.186206 | 590 | 0.17	0.185599 | 473 | 0.08	0.187329 | 10000 | 9.51
40	0.1705273270	0.171129 | 506 | 0.16	0.170447 | 912 | 0.20	0.171518 | 10000 | 23.85
50	0.1576763389	0.158461 | 609 | 0.28	0.157586 | 805 | 0.27	0.159822 | 10000 | 76.39
75	0.1647900104	0.166104 | 855 | 0.61	0.164747 | 823 | 0.51	0.168612 | 10000 | 276.50
100	0.1605997205	0.161848 | 969 | 0.94	0.160652 | 1168 | 1.03	0.166659 | 10000 | 630.13
150	0.1464856269	0.148102 | 1265 | 2.25	0.146557 | 930 | 1.56	-
200	0.1603708148	0.162349 | 1518 | 4.49	0.160448 | 979 | 2.67	-
300	0.1658212714	0.168437 | 1950 | 15.97	0.165877 | 1085 | 6.68	-
500	0.1648862238	0.168237 | 2622 | 69.61	0.164768 | 1100 | 20.27	-

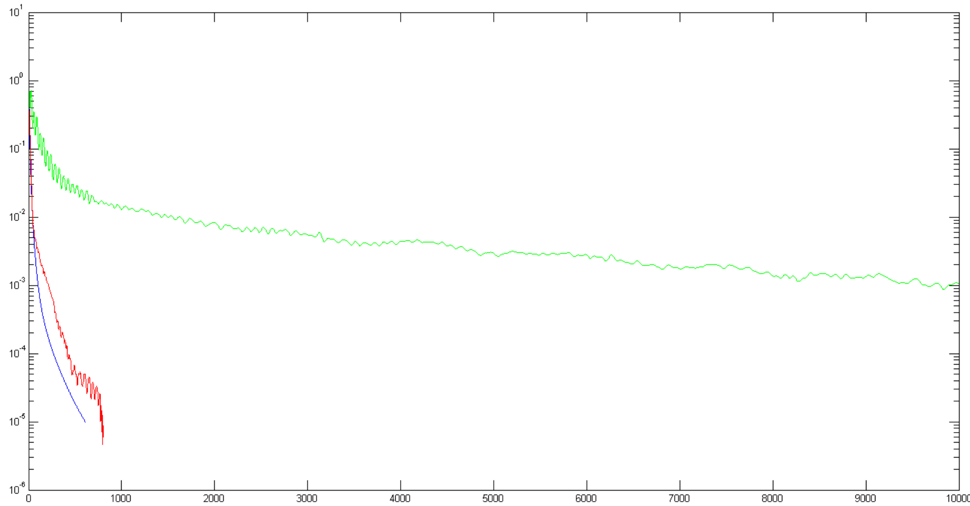
3.2.2 Residual plots

During each execution, the residuals are tracked and plotted in a log scale. Notice that for instances with small dimensions, Bregman ADMM converges slightly faster than semi-proximal ALM with respect to iteration count. When the dimension is large, semi-proximal ALM converges significantly faster. ADMM with reformulation does not display favorable residual convergence pattern even after the maximum number of iterations.

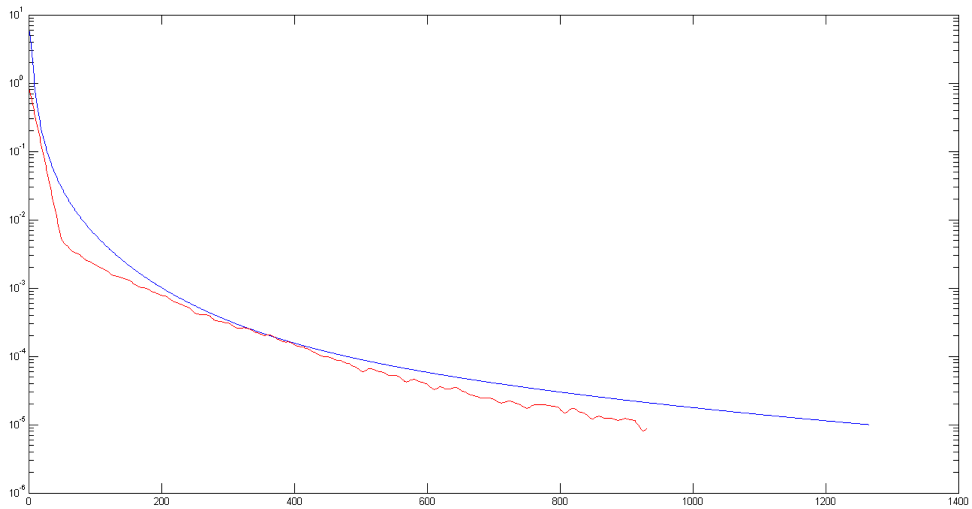
`m=n=10: (Blue: Bregman ADMM; Red: SPALM; Green: ADMM with reformulation)`



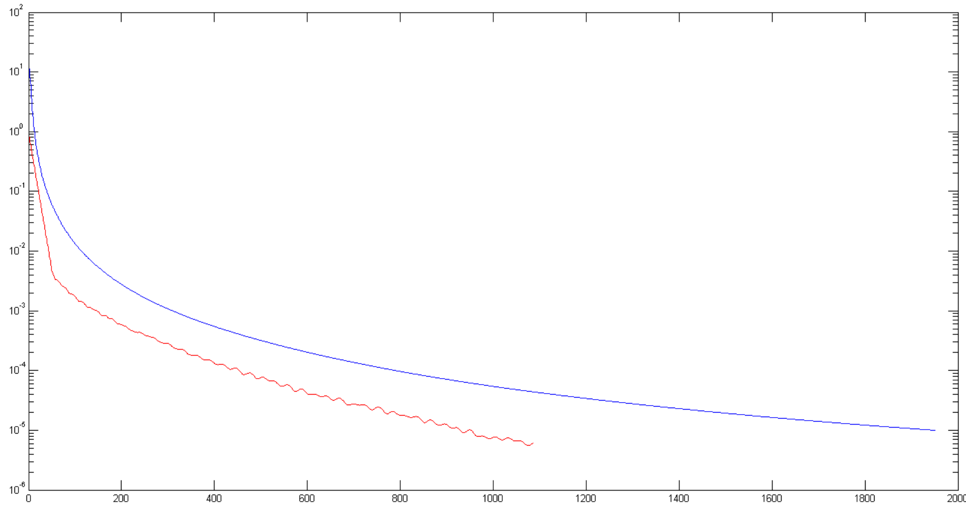
m=n=50



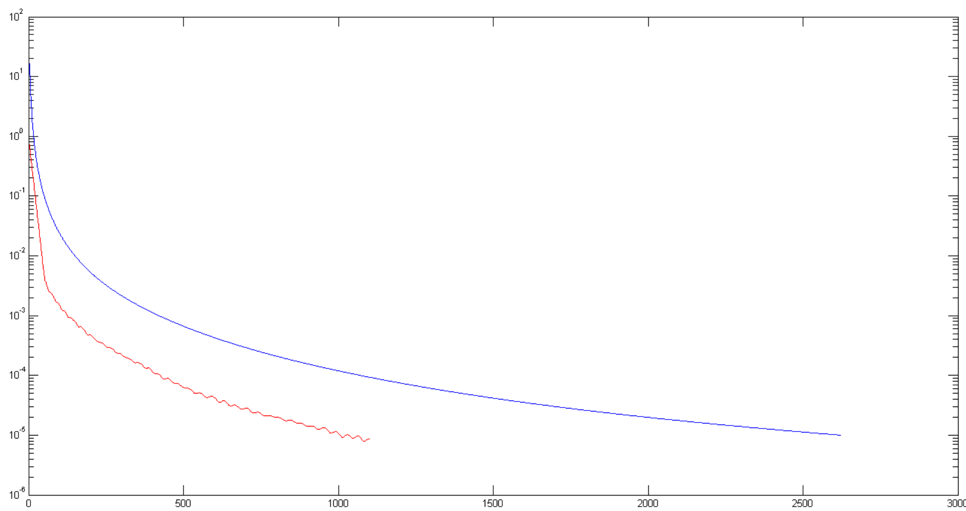
m=n=150 (ADMM with reformulation is not executed)



```
m=n=300 (ADMM with reformulation is not executed)
```




```
m=n=500 (ADMM with reformulation is not executed)
```



3.2.3 Discussion

Experiment results suggest that the classic ADMM with this reformulation is not practically applicable even on instances with small to medium dimensions. Besides the observed sub-linear rate of convergence, the residual plot shows an undesirable oscillating pattern near the end of the main loop ($m = n = 10$, after the 4000-th iteration). In addition, from the running statistics table in 3.2.1 it can be observed that the time for each iteration increases at a rate faster than $O(n^2)$. This is primarily due to the reformulation scheme, which constructs $\boldsymbol{x}_1 \in \mathbb{R}^{nm^2}$, where the input cost matrix $C \in \mathbb{R}^{m \times n}$.

When the dimension is large, semi-proximal ALM outperforms Bregman ADMM both in terms of total number of iterations and total running time. Besides an observed faster rate of convergence (after the same number of iterations, the residual of semi-proximal ALM is much smaller), semi-proximal ALM also exploits efficient linear algebra routines during each iteration, although the program for Bregman ADMM is also highly optimized via using compressed matrix operations to replace inner loops. In theory, both algorithms require $O(n \times m)$ floating point operations for each iteration.

As mentioned above, in practice, it is not necessary to calculate the objective value and residual during every iteration. When the dimension of the input instance is large, calculating the residual and objective value becomes relatively more time-consuming, since they always have the same order of time complexity as the main iterative schemes. An alternative is to calculate the residual and objective value after every fixed number of iterations.

One drawback of the above experiment is that the residuals of the algorithms are not consistently defined. The residual defined for Bregman ADMM measures the violation to the optimality condition (2.13) - (2.15), while the other two measure the violation to the optimality condition for the standard form LP problem, namely the KKT system (1.9a) - (1.9d). As such, setting the same terminal residual threshold may lead to biased results. However, for high-dimensional input instances, the experiment results suggest that semi-proximal ALM gives more accurate optimal objective as compared to Bregman ADMM in less number of iterations and less total running time. As such, using different residuals in this scenario does not affect the conclusion of the comparison.

Chapter 4

Conclusion

We have shown that ADMM-type methods can be used for solving LP problems, with semi-proximal ALM and Bregman ADMM significantly outperforms the classical ADMM with reformulation for instances with dimensions ranging from $m = n = 10$ to $m = n = 100$. For instances with large dimensions ($m = n = 100$ to $m = n = 500$), semi-proximal ALM converges faster than the Bregman ADMM and terminates faster given the same terminal residual threshold of 10^{-5}. Under this terminal residual threshold, semi-proximal ALM also gives more accurate objective values as compared to Bregman ADMM, relative to the corresponding optimal objectives given by Gurobi.

One thing to note is that the above Bregman ADMM iterative scheme (2.18a) - (2.18c) only works for the assignment problem, while semi-proximal ALM works on any standard form LP problem and does not depend on the structure of the assignment problem to achieve the demonstrated efficiency.

There have been numerous open problems on both theoretical aspects and applications in the field of alternating direction method of multipliers, including the study of rate of convergence, enabling time-varying step lengths and multi-block extensions of (1.2). Specific to the above algorithms, possible future work directions include formulating more general types of optimization problems into the linearly constrained convex optimization framework (1.2) with generic iterative scheme (1.4a) - (1.4c), deriving efficient updating formulas that lead to efficient implementation through C-acceleration or parallel computing techniques. In addition, as pointed out in [2], for a specific optimization problem, the Bregman divergence chosen may lead to large p and hence a step length τ that is too small to be practical. In fact, experiments in [2] shows that setting a larger τ which does not satisfy $\tau \leq \alpha\delta - 2\gamma$ in Theorem 2.2.2 still works well. Hence it is reasonable to conjecture that the conditions can be relaxed via alternative proving techniques.

References

[1] B. He & X. Yuan. *Alternating Direction Method of Multipliers for Linear Programming.* Optimization Online, SIAM. 2015.

[2] H. Wang & A. Banerjee. *Bregman Alternating Direction Method of Multipliers.* arXiv:1306.3203.

[3] C.H. Papadimitriou & K. Steiglitz. *13.2 Total Unimodularity.* Combinatorial Optimization: Algorithms and Complexity. New York, Dover Publications. 1998.

[4] J. Duchi, S. Shalev-Shwartz, Y. Singer & T. Chandra. *Efficient projections onto the l1-ball for learning in high dimensions.* ICML, pp. 272−279. 2008.

[5] M. Fortin & R. Glowinski. *On decomposition-coordination methods using an augmented Lagrangian.* Augmented Lagrangian Methods: Applications to the Solution of Boundary Problems. North-Holland, Amsterdam. pp. 97-146. 1983.

[6] M.S. Bazaraa & C.M. Shetty. *Appendix B.3.* Nonlinear programming: theory and algorithms. Singapore, John Wiley & Sons. 1990.

Appendix A: Matlab programs for the algorithms

Please refer to the files attached for the Matlab programs.

`admm_reformulated.m`

This is the program for the classic ADMM with reformulation proposed by He and Yuan in [1] in Section 2.1.

`badmm_assignment_problem_solver.m`

This is the program for the Bregman ADMM for assignment problem in Section 2.2.

`semi_proximal_alm_lp_solver.m`

This is the program for the semi-proximal ALM in Section 2.3.

`main_script_comparing_the_algorithms.m`

This is the script that generates random input instances, calculates the optimal objective value using Gurobi, calls the 3 programs and plots the residuals in log-scale.