Supplementary note to the paper titled "A Convergent 3-Block Semi-Proximal Alternating Direction Method of Multipliers for Conic Programming with 4-Type Constraints"

Defeng Sun, Kim-Chuan Toh, Liuqin Yang

October 21, 2014

The purpose of this note is to illustrate the performance of various variants of ADMM which can be employed to solve an SDP problem of the form:

$$\text{(P)} \quad \min\left\{\langle c,\, x\rangle + \delta_{\mathcal{K}}(x) + \delta_{\mathcal{K}_{\mathcal{P}}}(x) \mid \mathcal{A}_E x - b_E = 0\right\},$$

where $\mathcal{K} = \mathcal{S}^n_+$ and $\mathcal{P} = \{X \in \mathcal{S}^n \mid X \geq 0\}$. Its associated dual SDP is given by

$$\text{(D)} \quad \min\left\{\delta_{\mathcal{K}^*}(s) + \delta_{\mathcal{K}^*_p}(z) - \langle b_E,\, y_E\rangle \mid s + z + \mathcal{A}^*_E y_E = c\right\}.$$

Observe that (D) is naturally partition into three blocks of variables (s, z, y).

By using the framework of the 2-block semi-proximal ADMM proposed in [9] (presented as Algorithm sPADMM2 below) for solving the following convex problem

$$\min\left\{f(y) + g(z) \mid \mathcal{F}^* y + \mathcal{G}^* z = c\right\},$$

one can derive various variants of ADMM for solving either (P) or (D).

ADMM2-Prim It is derived by applying the classical ADMM directly (with $\tau = 1.618$ and $\mathcal{S} = 0$, $\mathcal{T} = 0$) to the follwing equivalent primal problem:

$$\min\left\{\langle c,\, x\rangle + (\delta_{\mathcal{K}}(u) + \delta_{\mathcal{K}_{\mathcal{P}}}(v)) \;\middle|\; \begin{pmatrix} \mathcal{A}_E \\ -I \\ -I \end{pmatrix} x + \begin{pmatrix} 0 & 0 \\ I & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} b_E \\ 0 \\ 0 \end{pmatrix}\right\}.$$

This is a convex problem with two blocks of variables with x as the first block and and (u, v) as the second block.

sPADMM2-Prim It is derived by applying sPADMM2 (with $\tau = 1.618$ and an appropriately chosen $\mathcal{S} \succeq 0$, $\mathcal{T} = 0$) to the follwing equivalent primal problem:

$$\min\left\{(\langle c,\, x\rangle + \delta_{\mathcal{K}}(x)) + \delta_{\mathcal{K}_{\mathcal{P}}}(v) \;\middle|\; \begin{pmatrix} \mathcal{A}_E \\ -I \end{pmatrix} x + \begin{pmatrix} 0 \\ I \end{pmatrix} v = \begin{pmatrix} b_E \\ 0 \end{pmatrix}\right\}.$$

ADMM2. It is derived by applying sPADMM2 (with $\tau = 1.95$ and an appropriately chosen $\mathcal{S} \succeq 0$. The convergence of the algorithm with this larger step-length is guaranteed as the objective on the (y_E, z)-part is linear) to the following equivalent dual problem:

$$\min\left\{(\delta_{\mathcal{K}^*}(s) + \delta_{\mathcal{K}^*_p}(u)) - \langle b_E,\, y_E\rangle \;\middle|\; \begin{pmatrix} s \\ u \end{pmatrix} + \begin{pmatrix} z + \mathcal{A}^*_E y_E \\ -z \end{pmatrix} = \begin{pmatrix} c \\ 0 \end{pmatrix}\right\}.$$

ADMM3c Our convergent semi-proximal ADMM algorithm (with $\tau = 1.618$) solving (D). It is derived from sPADMM2 with $\mathcal{S} = 0$ and an appropriately chosen $\mathcal{T} \succeq 0$.

ADMM3g A convergent ADMM algorithm (with Gaussian back substitution) proposed in [B. He, M. Tao, and X. Yuan, SIAM Journal on Optimization, 22 (2012), pp. 313–340] that is applied directly to (D).

Algorithm sPADMM2: **A generic 2-block semi-proximal ADMM.**

Choose appropriate positive semidefinite linear operators $\mathcal{S}$ and $\mathcal{T}$. Let $\sigma > 0$ and $\tau \in (0, \infty)$ be given parameters. Choose $y^0 \in \mathrm{dom}(f)$, $z^0 \in \mathrm{dom}(g)$, and $x^0 \in \mathcal{X}$. Perform the kth iteration as follows:

Step 1. Compute $y^{k+1} = \arg\min L_\sigma(y, z^k; x^k) + \dfrac{\sigma}{2}\|y - y^k\|^2_{\mathcal{S}}$.

Step 2. Compute $z^{k+1} = \arg\min L_\sigma(y^{k+1}, z; x^k) + \dfrac{\sigma}{2}\|z - z^k\|^2_{\mathcal{T}}$.

Step 3. Compute $x^{k+1} = x^k + \tau\sigma(\mathcal{F}^* y^{k+1} + \mathcal{G}^* z^{k+1} - c)$.

In the literature, there are three types of 2-block ADMM:

(a) The classic (generic, common, standard, ...) ADMM takes $\mathcal{S} = 0$ and $\mathcal{T} = 0$ in Algorithm sPADMM2.

(b) The classic (generic, common, standard, ...) proximal ADMM takes positive definite proximal terms $\mathcal{S} \succ 0, \mathcal{T} \succ 0$.

(c) The semi-proximal ADMM, i.e., Algorithm sPADMM2, does not need either $\mathcal{S}$ or $\mathcal{T}$ to be positive definite, and they need only to be positive semidefinite.

While both (a) and (b) have a long history, (c) is a relatively recent addition appearing in Appendix B of [9] in 2013. Though version (c) is the most versatile algorithm, it was hardly known to the ADMM community before the publication of this paper. As mentioned in Section 2 of the paper, the most important ADMM used in this paper is version (c), by taking $\mathcal{S}$, $\mathcal{T}$ to be only positive semidefinite but not positive definite and $\tau > 1$ (in particular, $\tau = 1.618$).

Figure 1 shows the performance profiles of ADMM3c, ADMM2, ADMM3g, ADMM-Prim and sPADMM2-Prim for a total of about 170 problems tested in Section 5.1.2 of the paper. From the performance profiles, one can safely conclude that among all the ADMM-type algorithms our newly proposed ADMM3c is the most suitable, if not the best possible, for solving SDPs.

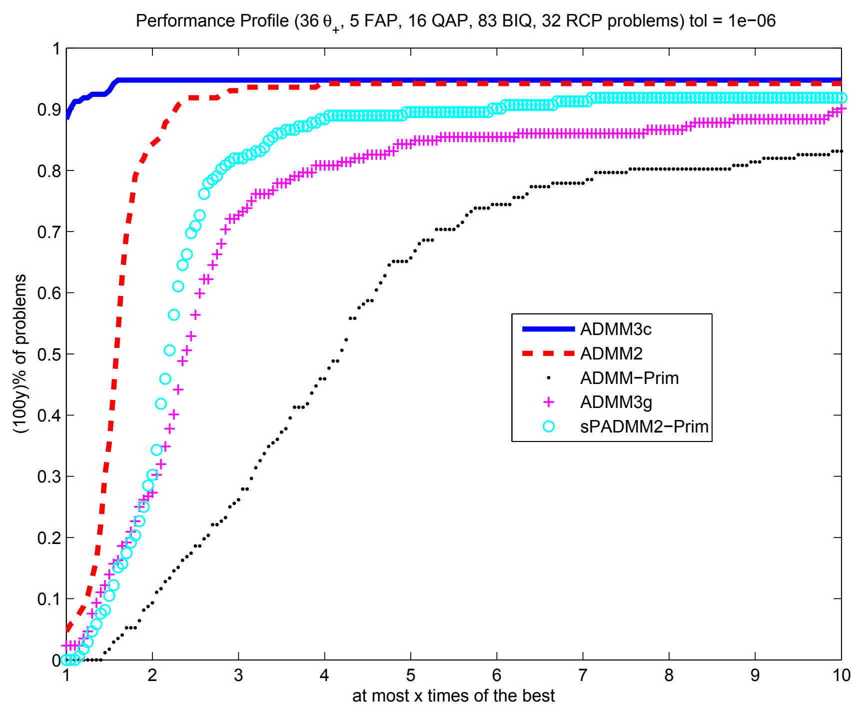

Figure 1: Performance profiles (in terms of computing time) of ADMM3c, ADMM2, ADMM3g, ADMM-Prim and sPADMM2-Prim on $[1, 10]$