MULTIPROCESSOR APPROXIMATE MESSAGE PASSING WITH COLUMN-WISE PARTITIONING

Yanting Ma,⋆ Yue M. Lu,† and Dror Baron⋆

⋆Department of Electrical and Computer Engineering; NC State University; Raleigh, NC 27695
†Paulson School of Engineering and Applied Sciences; Harvard University; Cambridge, MA 02138

ABSTRACT

Solving a large-scale regularized linear inverse problem using multiple processors is important in various real-world applications due to the limitations of individual processors and constraints on data sharing policies. This paper focuses on the setting where the matrix is partitioned column-wise. We extend the algorithmic framework and the theoretical analysis of approximate message passing (AMP), an iterative algorithm for solving linear inverse problems, whose asymptotic dynamics are characterized by state evolution (SE). In particular, we show that column-wise multiprocessor AMP (C-MP-AMP) obeys an SE under the same assumptions when the SE for AMP holds. The SE results imply that (*i*) the SE of C-MP-AMP converges to a state that is no worse than that of AMP and (*ii*) the asymptotic dynamics of C-MP-AMP and AMP can be identical. Moreover, for a setting that is not covered by SE, numerical results show that damping can improve the convergence performance of C-MP-AMP.

***Index Terms*—** approximate message passing, column-wise partitioning, linear inverse problem, multiprocessor computing.

1. INTRODUCTION

Many scientific and engineering problems can be modeled as solving a regularized linear inverse problem of the form

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w}, \tag{1}$$

where the goal is to estimate the unknown $\mathbf{x} \in \mathbb{R}^N$ given the matrix $\mathbf{A} \in \mathbb{R}^{n\times N}$ and statistical information about the signal $\mathbf{x}$ and the noise $\mathbf{w} \in \mathbb{R}^n$.

In some scenarios, it might be desirable to partition the matrix $\mathbf{A}$ either column-wise or row-wise and store the sub-matrices at different processors. The partitioning style depends on data availability, computational considerations, and privacy concerns. For example, in high-dimensional settings where $N \gg n$, or in situations where the columns of $\mathbf{A}$, which represent features in feature selection problems [1], cannot be shared among processors for privacy preservation, column-wise partitioning might be preferable. In this paper, we consider multiprocessor computing for the (non-overlapping) column-wise partitioned linear inverse problem:

$$\mathbf{y} = \sum_{p=1}^{P} \mathbf{A}_p \mathbf{x}_p + \mathbf{w}, \tag{2}$$

where P is the number of processors, $\mathbf{A}_p \in \mathbb{R}^{n\times N_p}$ is the sub-matrix that is stored in Processor p, and $\sum_{p=1}^{P} N_p = N$.

Many studies on solving the column-wise partitioned linear inverse problem (2) have been in the context of distributed feature selection. Zhou *et al.* [2] modeled feature selection as a parallel group testing problem. Wang *et al.* [3] proposed to de-correlate the data matrix before partitioning, and each processor then works independently using the de-correlated matrix without communication with other processors. Peng *et al.* [4] studied problem (2) in the context of optimization, where they proposed a greedy coordinate-block descent algorithm and a parallel implementation of the fast iterative shrinkage-thresholding algorithm (FISTA) [5].

Our work is based on the approximate message passing (AMP) framework [6]. AMP is an efficient iterative algorithm for solving linear inverse problems (1). In the large scale random setting, its average asymptotic dynamics are characterized by a state evolution (SE) formalism [7], which allows one to accurately predict the average estimation error at every iteration. Recently, a finite-sample analysis of AMP [8, 9] showed that when the prior distribution of the input signal $\mathbf{x}$ has independent and identically distributed (i.i.d.) sub-Gaussian entries,[1] the average performance of AMP concentrates to the SE prediction at an exponential rate in N. This concentration result has explained the good empirical performance of AMP when the signal dimension is above a few thousands.

Our goal is to extend the AMP algorithmic framework and the SE analysis in [9] to the column-wise partitioned linear inverse problem (2). We show that column-wise multiprocessor AMP (C-MP-AMP) obeys a new SE under the same model assumptions where the SE for AMP holds. With the new SE, we can predict the average estimation error in each processor at every iteration. Moreover, the comparison between the SE of AMP and that of C-MP-AMP implies that (*i*) the estimation error of C-MP-AMP is no worse than that of AMP and (*ii*) with a specific communication schedule between the processors and the fusion center that coordinates the processors, the asymptotic dynamics of C-MP-AMP are identical to that of AMP. This result implies a speedup linear in the number of processors.

It is worth mentioning that row-wise multiprocessor AMP [10–12] obeys the same SE as AMP, because it distributes the computation of matrix-vector multiplication among multiple processors and aggregates the results before any other operations. Some existing work on row-wise multiprocessor AMP [12–14] introduces lossy compression to the communication between processors and the fusion center, whereas we assume perfect communication and focus on the theoretical analysis of C-MP-AMP.

The remainder of the paper is organized as follows. Section 2 first reviews AMP and its SE, and then introduces our C-MP-AMP

Yanting Ma and Dror Baron were supported by the National Science Foundation (NSF) under grant CCF-1217749. Yue M. Lu was supported in part by the NSF under grant CCF-1319140 and in part by the U.S. Army Research Office under grant W911NF-16-1-0265.

[1] A random variable X is sub-Gaussian if there exist positive constants c and κ such that $P(|X - \mathbb{E}X| > \epsilon) \leq ce^{-\kappa\epsilon^2}, \forall \epsilon > 0$.

and the new SE, as well as the implications of the new SE, Section 3 provides the proof sketch of SE for C-MP-AMP, Section 4 demonstrates the properties of C-MP-AMP via numerical results, and Section 5 concludes the paper.

2. COLUMN-WISE MULTIPROCESSOR AMP

2.1. Review of AMP

Approximate message passing (AMP) [6] is a fast iterative algorithm for solving linear inverse problems (1). Starting with an all-zero vector $\mathbf{x}^0$ as its initial estimate, at the tth iteration, AMP proceeds according to

$$\mathbf{z}^t = \mathbf{y} - \mathbf{A}\mathbf{x}^t + \frac{\mathbf{z}^{t-1}}{n}\sum_{i=1}^{N}\eta'_{t-1}([\mathbf{x}^{t-1} + \mathbf{A}^*\mathbf{z}^{t-1}]_i), \quad (3)$$

$$\mathbf{x}^{t+1} = \eta_t(\mathbf{x}^t + \mathbf{A}^*\mathbf{z}^t), \quad (4)$$

where vectors with negative iteration indices are all-zero vectors, $\mathbf{A}^*$ denotes the transpose of a matrix $\mathbf{A}$, $\eta_t : \mathbb{R} \to \mathbb{R}$ is a Lipschitz continuous function with weak derivative η'_t, for any $\mathbf{u} \in \mathbb{R}^N$, $[\mathbf{u}]_i$ denotes its ith entry, and the vector $(\eta_t(u_1), \eta_t(u_2), ..., \eta_t(u_N))$ is denoted by $\eta_t(\mathbf{u})$. Examples of $\eta_t(\cdot)$ are soft-thresholding denoisers [6] and Bayes-optimal (posterior mean) denoisers [15].

Under the assumptions listed in [9, Section 1.1] for proving SE for AMP, denoting by p_X the distribution of the i.i.d. entries of the signal $\mathbf{x}$, the sequence of the estimates $\{\mathbf{x}^t\}$ generated by AMP (3,4) has the following property [7–9]:

$$\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}\phi(x_i^{t+1}, x_i) \overset{\text{a.s.}}{=} \mathbb{E}\left[\phi(\eta_t(X + \tau^t Z), X)\right], \quad (5)$$

where $\phi : \mathbb{R}^2 \to \mathbb{R}$ is a pseudo-Lipschitz function of order 2 (PL(2)),[2] $X \sim p_X$, Z is a standard normal random variable that is independent of X, and τ^t is defined via the following recursion ($(\sigma^0)^2 = \delta^{-1}\mathbb{E}[X^2]$, $\delta = n/N$):

$$(\tau^t)^2 = \sigma_W^2 + (\sigma^t)^2,$$
$$(\sigma^{t+1})^2 = \delta^{-1}\mathbb{E}\left[\left(\eta_t(X + \tau^t Z) - X\right)^2\right]. \quad (6)$$

If we choose $\phi(x, y) = (x - y)^2$, then (5) characterizes the mean square error (MSE) achieved by AMP at each iteration.

2.2. Column-wise multiprocessor AMP

In our proposed column-wise multiprocessor AMP (C-MP-AMP) algorithm, the fusion center collects vectors that represent the estimations of the portion of the measurement vector $\mathbf{y}$ contributed by the data from individual processors according to a pre-defined communication schedule. The sum of these vectors is computed in the fusion center and transmitted to all processors. Each processor performs standard AMP iterations with a new equivalent measurement vector, which is computed using the vector received from the fusion center. The pseudocode for C-MP-AMP is presented in Algorithm 1.

[2]Recall the definition of PL(2) in [7]: A function $f : \mathbb{R}^m \to \mathbb{R}$ is said to be PL(2) if there is $L > 0$ such that $|f(x) - f(y)| \le L(1 + \|x\| + \|y\|)\|x - y\|, \forall x, y \in \mathbb{R}^m$, where $\|\cdot\|$ denotes the Euclidean norm.

Algorithm 1 C-MP-AMP

Inputs to Processor p: $\mathbf{y}$, $\mathbf{A}_p$, $\{\widehat{t}_s\}_{s=0,...,\widehat{s}}$ (maximum number of inner iterations at each outer iteration).
Initialization: $\mathbf{x}_p^{0,\widehat{t}_0} = \mathbf{0}$, $\mathbf{z}_p^{0,\widehat{t}_0-1} = \mathbf{0}$, $\mathbf{r}_p^{0,\widehat{t}_0} = \mathbf{0}$, $\forall p$.
for $s = 1 : \widehat{s}$ **do** (loop over outer iterations)
At the fusion center: $\mathbf{g}^s = \sum_{u=1}^{P}\mathbf{r}_u^{s-1,\widehat{t}_{s-1}}$
At Processor p:
$\mathbf{x}_p^{s,0} = \mathbf{x}_p^{s-1,\widehat{t}_{s-1}}$, $\mathbf{r}_p^{s,0} = \mathbf{r}_p^{s-1,\widehat{t}_{s-1}}$
for $t = 0 : \widehat{t}_s - 1$ **do** (loop over inner iterations)
$\mathbf{z}_p^{s,t} = \mathbf{y} - \mathbf{g}_s - \left(\mathbf{r}_p^{s,t} - \mathbf{r}_p^{s,0}\right)$
$\mathbf{x}_p^{s,t+1} = \eta_{s,t}(\mathbf{x}_p^{s,t} + \mathbf{A}_p^*\mathbf{z}_p^{s,t})$
$\mathbf{r}_p^{s,t+1} = \mathbf{A}_p\mathbf{x}^{s,t+1} - \frac{\mathbf{z}_p^{s,t}}{n}\sum_{i=1}^{N_p}\eta'_{s,t}([\mathbf{x}_p^{s,t} + \mathbf{A}_p^*\mathbf{z}_p^{s,t}]_i)$.
Output from Processor p: $\mathbf{x}_p^{\widehat{s},\widehat{t}_{\widehat{s}}}$.

2.3. State evolution

Similar to AMP, the dynamics of the C-MP-AMP algorithm can be characterized by an SE formula. Let $(\sigma_p^{0,\widehat{t}})^2 = \delta_p^{-1}\mathbb{E}[X^2]$, where $\delta_p = n/N_p$, $\forall p = 1, ..., P$. For outer iterations $1 \le s \le \widehat{s}$ and inner iterations $0 \le t \le \widehat{t}_s$, we define the sequences $\{(\sigma_p^{s,t})^2\}$ and $\{(\tau_p^{s,t})^2\}$ as

$$(\sigma_p^{s,0})^2 = (\sigma_p^{s-1,\widehat{t}})^2, \quad (7)$$

$$(\tau_p^{s,t})^2 = \sigma_W^2 + \sum_{u=1}^{P}(\sigma_u^{s,0})^2 + \left((\sigma_p^{s,t})^2 - (\sigma_p^{s,0})^2\right), \quad (8)$$

$$(\sigma_p^{s,t+1})^2 = \delta_p^{-1}\mathbb{E}\left[\left(\eta_{s,t}(X + \tau_p^{s,t}Z) - X\right)^2\right], \quad (9)$$

where Z is standard normal and independent of X. With these definitions, we have the following theorem for C-MP-AMP.

Theorem 1. *Under the assumptions listed in [9, Section 1.1], for $p = 1, ..., P$, let $n/N_p \to \delta_p \in (0, \infty)$ be a constant. Define $N = \sum_{p=1}^{P} N_p$. Then for any PL(2) function $\phi : \mathbb{R}^2 \to \mathbb{R}$, we have*

$$\lim_{N\to\infty}\frac{1}{N_p}\sum_{i=1}^{N_p}\phi(x_{p,i}^{s,t+1}, x_{p,i}) \overset{\text{a.s.}}{=} \mathbb{E}\left[\phi(\eta_{s,t}(X + \tau_p^{s,t}Z), X)\right], \forall p,$$

where $\mathbf{x}_p^{s,t+1}$ is generated by the C-MP-AMP algorithm, $\tau_p^{s,t}$ is defined in (7–9), $X \sim p_X$, and Z is a standard normal random variable that is independent of X.

Remark 1: C-MP-AMP converges to a fixed point that is no worse than that of AMP. This statement can be demonstrated as follows. When C-MP-AMP converges, the quantities in (7–9) do not keep changing, hence we can drop all the iteration indices for fixed point analysis. Notice that the last term on the right hand side (RHS) of (8) vanishes, which leaves the RHS independent of p. That is, $(\tau_p^{s,t})^2$ are equal for all p, hence we can further drop the processor index for $(\tau_p^{s,t})^2$. Denote $(\tau_p^{s,t})^2$ by τ^2 for all s, t, p, and plug (9) into (8), then

$$\tau^2 = \sigma_W^2 + \sum_{p=1}^{P}\delta_p^{-1}\mathbb{E}\left[(\eta(X + \tau Z) - X)^2\right]$$
$$\overset{(a)}{=} \sigma_W^2 + \delta^{-1}\mathbb{E}\left[(\eta(X + \tau Z) - X)^2\right],$$

which is identical to the fixed point equation obtained from (6). In the above, step (a) holds because $\sum_{p=1}^{P}\delta_p^{-1} = \sum_{p=1}^{P}\frac{N_p}{n} = \frac{N}{n}$.

Because AMP always converges to the worst fixed point of (6) [16], the average asymptotic performance of C-MP-AMP is identical to AMP when there is only one solution to the fixed point equation, and at least as good as AMP in case of multiple fixed points.

Remark 2: The asymptotic dynamics of C-MP-AMP can be identical to AMP with a specific communication schedule. This can be achieved by letting $\widehat{t}_s = 1, \forall s$. In this case, the quantity $(\tau_p^{s,t})$ is involved only for $t = 0$. Because the last term in (8) is 0 when $t = 0$, the computation of $(\tau_p^{s,0})^2$ is independent of p. Therefore, $\tau_p^{s,0}$ are again equal for all p. Dropping the processor index for $(\tau_p^{s,t})^2$, the recursion in (7–9) can be simplified as

$$(\tau^{s,0})^2 = \sigma_W^2 + \sum_{p=1}^{P} \delta_p^{-1} \mathbb{E}\left[\left(\eta_{s,0}(X + \tau^{s,0}Z) - X\right)^2\right]$$
$$= \sigma_W^2 + \delta^{-1}\mathbb{E}\left[\left(\eta_{s-1,0}(X + \tau^{s-1,0}Z) - X\right)^2\right]$$

where the iteration evolves over s, which is identical to (6) evolving over t.

3. PROOF SKETCH OF THEOREM 1

We now provide the sketch of our proof for Theorem 1. Without loss of generality, we assume the sequence $\{\widehat{t}_s\}_s$ in Algorithm 1 that determines the communication schedule to be a constant $\widehat{t}$. Using similar notations as in the SE proof for AMP [7, 9], we write the recursion in Algorithm 1 in a different form:

$$\mathbf{b}_p^k = \mathbf{A}_p\mathbf{q}_p^k - \lambda_p^k\mathbf{m}_p^{k-1}, \qquad \mathbf{q}_p^k = f_k(\mathbf{h}_p^k, \mathbf{x}_p),$$
$$\mathbf{h}_p^{k+1} = \mathbf{A}_p^*\mathbf{m}_p^k - \mathbf{q}_p^k, \qquad \mathbf{m}_p^k = \mathbf{b}_p^k - \mathbf{w} + \sum_{u=1,u\neq p}^{P} \mathbf{b}_u^{\theta(k)}, \quad (10)$$

where
$f_k(\mathbf{h}_p^k, \mathbf{x}_p) = \eta_{k-1}(\mathbf{x}_p - \mathbf{h}_p^k) - \mathbf{x}_p, \quad \lambda_p^k = \frac{1}{n}\sum_{i=1}^{N_p} f_k'(h_{p,i}^k, x_{p,i}).$

The derivative $f_k'(\cdot)$ of $f_k(\cdot)$ is taken with respect to the first argument. The equivalence of the above recursion and Algorithm 1 can be seen by letting $k = s\widehat{t} + t$, $\theta(k) = \lfloor k/\widehat{t}\rfloor\widehat{t}$, $\mathbf{m}_p^k = -\mathbf{z}_p^{s,t}$, $\mathbf{q}_p^k = \mathbf{x}_p^{s,t} - \mathbf{x}_p$, $\mathbf{b}_p^k = \mathbf{r}_p^{s,t} - \mathbf{A}_p\mathbf{x}_p$, and $\mathbf{h}_p^{k+1} = \mathbf{x}_p - \left(\mathbf{x}_p^{s,t} + \mathbf{A}_p^*\mathbf{z}_p^{s,t}\right)$. We notice that the difference between (10) and the recursion for AMP [9] is the update for $\mathbf{m}_p^k$, where (10) has an extra term $\sum_{u=1,u\neq p}^{P}\mathbf{b}_u^{\theta(k)}$, which represents the interference from other processors.

To prove Theorem 1, we need to show that for any PL(2) function $\phi : \mathbb{R}^2 \to \mathbb{R}$, we have

$$\lim_{N_p\to\infty} \frac{1}{N_p}\sum_{i=1}^{N_p} \phi\left(\eta_k(x_{p,i} - h_{p,i}^{k+1}), x_{p,i}\right)$$
$$\overset{\text{a.s.}}{=} \mathbb{E}\left[\phi(\eta_k(X + \tau_p^k Z), X)\right]. \quad (11)$$

That is, the equivalent noise vector $\mathbf{h}_p^{k+1}$ is approximately i.i.d. Gaussian with mean zero and variance $(\tau_p^k)^2$ at every iteration. Moreover, as a necessary step in proving Theorem 1, the proof will show that $\|\mathbf{m}_p^k\|_2^2/n$ also concentrates around $(\tau_p^k)^2$. Notice that $\mathbf{m}_p^k = -\mathbf{z}_p^{s,t}$, hence $(\tau_p^{s,t})^2$ can be approximated by $\|\mathbf{z}_p^{s,t}\|_2^2/n$, which can be used as side-information for $\eta_{s,t}$ when implementing C-MP-AMP. Recall that $\mathbf{m}_p^k$ contains interference $\sum_{u=1,u\neq p}^{P}\mathbf{b}_u^{\theta(k)}$ from other processors, hence the main challenge in our proof is to characterize the concentration of $|(\mathbf{b}_p^r)^*\mathbf{b}_q^k|/n, \forall p \neq q, 0 \leq r \leq k$.

Following the SE proof for AMP [7, 9], we need to characterize the distribution of $\mathbf{h}_p^{k+1}$ conditioned on the sigma algebra generated by the quantities that have already been computed when updating $\mathbf{h}_p^{k+1}$, which is the sigma algebra generated by $\mathbf{b}_p^0, ..., \mathbf{b}_p^k$, $\mathbf{m}_p^0, ..., \mathbf{m}_p^k$, $\mathbf{q}_p^0, ..., \mathbf{q}_p^k$, $\mathbf{h}_p^1, ..., \mathbf{h}_p^k$, $\forall p$, as well as $\mathbf{x}$ and $\mathbf{w}$. We first compute the conditional distribution of $\mathbf{A}_p$, and then use the updating equation to compute the conditional distribution of $\mathbf{h}_p^{k+1}$. Similar to the situation in [7, 9], conditioning on the sigma algebra is equivalent to conditioning on the linear observations

$$\mathbf{Y}_p^{k+1} = \mathbf{A}_p\mathbf{Q}_p^{k+1}, \quad \mathbf{X}_p^k = \mathbf{A}_p^*\mathbf{M}_p^k.$$

In the above, only $\mathbf{A}_p$, $p = 1, ..., P$, are treated as random and

$$\mathbf{Y}_p^{k+1} = [\mathbf{b}_p^0|\mathbf{b}_p^1 + \lambda_p^1\mathbf{m}_p^0|...|\mathbf{b}_p^k + \lambda_p^k\mathbf{m}_p^{k-1}], \ \mathbf{Q}_p^{k+1} = [\mathbf{q}_p^0|...|\mathbf{q}_p^k],$$
$$\mathbf{X}_p^k = [\mathbf{h}_p^1 + \mathbf{q}_p^0|...|\mathbf{h}_p^k + \mathbf{q}_p^{k-1}], \quad \mathbf{M}_p^k = [\mathbf{m}_p^0|...|\mathbf{m}_p^{k-1}].$$

Let $\mathsf{P}^{\parallel}_{\mathbf{Q}_p^{k+1}} = (\mathbf{Q}_p^{k+1})\left((\mathbf{Q}_p^{k+1})^*(\mathbf{Q}_p^{k+1})\right)^{-1}(\mathbf{Q}_p^{k+1})^*$, $\mathsf{P}^{\perp}_{\mathbf{Q}_p^{k+1}} = \mathsf{I} - \mathsf{P}^{\parallel}_{\mathbf{Q}_p^{k+1}}$, $\mathsf{P}^{\parallel}_{\mathbf{M}_p^k} = (\mathbf{M}_p^k)\left((\mathbf{M}_p^k)^*(\mathbf{M}_p^k)\right)^{-1}(\mathbf{M}_p^k)^*$, and $\mathsf{P}^{\perp}_{\mathbf{M}_p^k} = \mathsf{I} - \mathsf{P}^{\parallel}_{\mathbf{M}_p^k}$. The conditional distribution of $\mathbf{A}_p$ is

$$\mathbf{A}_p|\mathcal{G}^{k+1,k} \overset{\text{d}}{=} \mathbf{E}^{k+1,k} + \mathsf{P}^{\perp}_{\mathbf{M}_p^k}\widetilde{\mathbf{A}}_p\mathsf{P}^{\perp}_{\mathbf{Q}_P^{k+1}}, \text{ where}$$

$$\mathbf{E}_p^{k+1,k} = \mathbf{Y}_p^{t_1}((\mathbf{Q}_p^{t_1})^*\mathbf{Q}_p^{t_1})^{-1}(\mathbf{Q}_p^{t_1})^* + \mathbf{M}_p^t((\mathbf{M}_p^t)^*\mathbf{M}_p^t)^{-1}(\mathbf{X}_p^t)^*$$
$$- \mathbf{M}_p^t((\mathbf{M}_p^t)^*\mathbf{M}_p^t)^{-1}(\mathbf{M}_p^t)^*\mathbf{Y}_p^{t_1}((\mathbf{Q}_p^{t_1})^*\mathbf{Q}_p^{t_1})^{-1}(\mathbf{Q}_p^{t_1})^*.$$

In the above, $\widetilde{\mathbf{A}}_p$ (an independent copy of $\mathbf{A}_p$) is independent of $\mathcal{G}^{k+1,k}$ and $\widetilde{\mathbf{A}}_p$ is independent of $\widetilde{\mathbf{A}}_q$ for $p \neq q$. The independence can be demonstrated by considering $\mathbf{A}_p = \mathsf{P}^{\parallel}_{\mathbf{Q}_p^{k+1}}\mathbf{A}_p + \mathsf{P}^{\perp}_{\mathbf{Q}_p^{k+1}}\mathbf{A}_p$. Notice that the conditioning involves $\mathsf{P}^{\parallel}_{\mathbf{Q}_p^{k+1}}\mathbf{A}_p$, which is uncorrelated with (hence independent due to being Gaussian) $\mathsf{P}^{\perp}_{\mathbf{Q}_p^{k+1}}\mathbf{A}_p$. Similarly, $\mathsf{P}^{\parallel}_{\mathbf{M}_p^k}\mathbf{A}_p$ is independent of $\mathsf{P}^{\perp}_{\mathbf{M}_p^k}\mathbf{A}_p$. Together with the fact that $\mathbf{A}_p$ is independent of $\mathbf{A}_q$, we can show that $\widetilde{\mathbf{A}}_p$ is independent of the conditioning, and $\widetilde{\mathbf{A}}_p$ is independent of $\widetilde{\mathbf{A}}_q$.

To prove (11), we follow the strategy of Rush and Venkataramanan [9]. That is, we prove the concentration results listed in [9, Lemma 5] by induction for the recursion defined in (10). As mentioned earlier, the difference between (10) and the recursion for AMP [9] is the update for $\mathbf{m}_p^k$, where (10) has an extra term coming from other processors. Hence, in addition to [9, Lemma 5], we need to include the following concentration inequalities that involve interactions among processors. We need to show that for all $0 < \epsilon < 1$, there is $K_k, \kappa_k > 0$ such that

$$P\left(\frac{|(\Delta_p^{k,k})^*\Delta_q^{k,k}|}{n} \geq \epsilon\right) \leq K_k e^{-\kappa_k n\epsilon^2}, \forall p \neq q, \quad (12)$$
$$P\left(\left|\frac{(\mathbf{b}_p^r)^*\mathbf{b}_q^k}{n}\right| \geq \epsilon\right) \leq K_k e^{-\kappa_k n\epsilon^2}, \forall p \neq q, \ 0 \leq r \leq k, \quad (13)$$

where the expression for $\Delta_p^{k,k}$ resembles those of (34) and (36) in [9, Lemma 4]. The proof of (12) can be completed by applying the Cauchy-Schwarz inequality $|(\Delta_p^{k,k})^*\Delta_q^{k,k}| \leq \|\Delta_p^{k,k}\|\|\Delta_q^{k,k}\|$ and part (a) of [9, Lemma 5]. The proof of (13) uses induction in k and relies on the fact that $\widetilde{\mathbf{A}}_p$ is independent of $\widetilde{\mathbf{A}}_q$ for $p \neq q$.

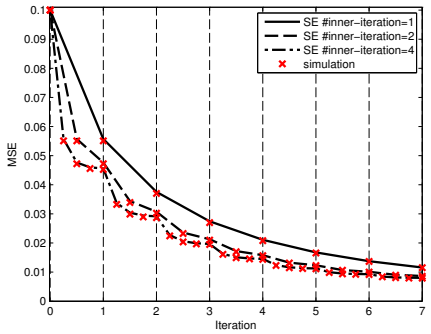

Fig. 1. Verification of SE for C-MP-AMP with various communication schedules. (P=3, N=30000, n=9000, SNR=15dB.)

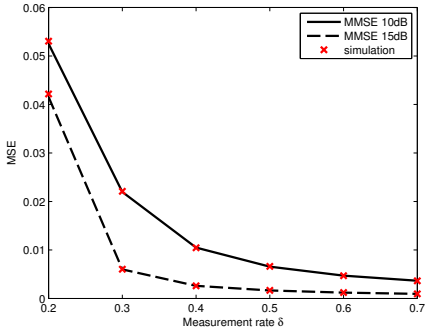

Fig. 2. Verification that C-MP-AMP achieves the MMSE at various measurement rates $\delta = n/N$ and SNR levels. (P=3, N=30000.)

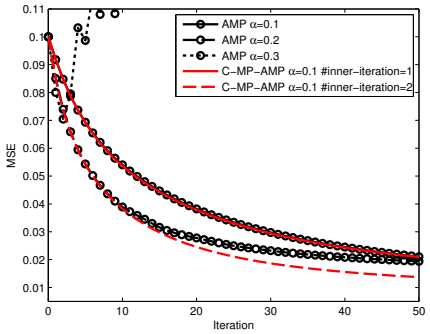

Fig. 3. Numerical example of C-MP-AMP with damping [17] for non-Gaussian matrices. (P=3, N=30855, n=9257, SNR=15dB.)

4. NUMERICAL RESULTS

In this section, we provide numerical results for C-MP-AMP for both the Gaussian matrix setting and non-Gaussian matrix setting. In the Gaussian matrix setting, where SE is justified rigorously in Section 3, we numerically verify SE and the properties implied by SE. In the non-Gaussian matrix setting, where SE is not justified for AMP or C-MP-AMP, we show numerical evidence that C-MP-AMP converges when damping [17], which is commonly used in AMP for non-Gaussian matrices to improve the convergence performance, is applied. In all simulations, entries of the unknown vector $\mathbf{x}$ are independent realizations of a Bernoulli-Gaussian random variable X, which has density function $f_X(x) = 0.9\delta(x) + 0.1\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2}$, where $\delta(\cdot)$ is the Dirac delta function. The measurement noise vector $\mathbf{w}$ has i.i.d. Gaussian $\mathcal{N}(0, \sigma_W^2)$ entries, where σ_W^2 depends on SNR as SNR $:= 10\log_{10}\left((N\mathbb{E}[X^2])(n\sigma_W^2)\right)$. The estimation function $\eta_{s,t}$ is defined as $\eta_{s,t}(u) = \mathbb{E}[X|X + \tau_p^{s,t}Z = u]$, where Z is a standard normal random variable that is independent of X and $\tau_p^{s,t}$ is estimated by $\|\mathbf{z}_p^{s,t}\|/\sqrt{n}$, which is implied by SE. All numerical results are averaged over 50 trials.

4.1. Gaussian matrix setting

We first show that the MSE of C-MP-AMP is accurately predicted by SE when the matrix $\mathbf{A}$ has i.i.d. Gaussian entries with $A_{i,j} \sim \mathcal{N}(0, 1/n)$. It can be seen from Figure 1 that the MSE achieved by C-MP-AMP from simulations (red crosses) matches the MSE predicted by SE (black curves) at every outer iteration s and inner iteration t for various choices of numbers of inner iterations (the number of red crosses within a grid).

As we have discussed in Remark 1, the average estimation error of C-MP-AMP is no worse than that of AMP, which implies that C-MP-AMP can achieve the minimum mean square error (MMSE) of large random linear systems [18] whenever AMP achieves it.[3] This point is verified in Figure 2.

4.2. Non-Gaussian matrix setting

The non-Gaussian matrices used in our simulation model the 3rd order Taylor expansion of a function $g : \mathbb{R}^J \to \mathbb{R}$. The first J columns contain i.i.d. Gaussian entries, and the rest of the columns are obtained by taking element-wise products of each pair of the J columns (2nd order terms) and each three of the J columns (3rd order terms). Hence, the unknown vector $\mathbf{x}$ contains the coefficients in the Taylor expansion. The matrix is normalized to have column norm equal to 1. In our simulations, both AMP (3,4) and C-MP-AMP (Algorithm 1) have diverged with this type of matrix.

Damping [17] is a simple and effective technique that improves the convergence performance of AMP by taking a convex combination of the estimates from the last iteration and the current iteration to slow down the evolution of the algorithm. In other words, the update for $\mathbf{x}^{t+1}$ in (4) becomes

$$\mathbf{x}^{t+1} = \alpha\eta_t(\mathbf{x}^t + \mathbf{A}^*\mathbf{z}^t) + (1-\alpha)\mathbf{x}^t,$$

where $\alpha \in (0, 1]$ is a parameter that controls the damping strength. It is possible to make α adaptive [19], but it is not the focus of this paper; we keep α constant for both AMP and C-MP-AMP in our simulation. Damping for C-MP-AMP is done at every processor. That is, the update for $\mathbf{x}_p^{s,t+1}$ changes to $\mathbf{x}_p^{s,t+1} = \alpha\eta_{s,t}(\mathbf{x}_p^{s,t} + \mathbf{A}_p^*\mathbf{z}_p^{s,t}) + (1-\alpha)\mathbf{x}_p^{s,t}$, $\forall p$, in Algorithm 1. Figure 3 shows that with the same damping parameter α, C-MP-AMP with one inner iteration per outer iteration has the same average dynamics as AMP, and that increasing the number of inner iterations can reduce the number of outer iterations, which reduces the communication frequency between the fusion center and the processors while achieving the same error.

5. CONCLUSION

In this paper, we extended the algorithmic framework and the state evolution (SE) analysis of approximate message passing (AMP) [6] to the column-wise partitioned linear inverse problem (2). We showed that the column-wise multiprocessor AMP (C-MP-AMP) algorithm obeys an SE under the same model assumptions when the SE for AMP holds. We concluded from the SE that there is no increase in estimation error in the settings when SE holds and that with a specific communication schedule between the fusion center and the processors, C-MP-AMP enjoys a speedup linear in the number of processors. Moreover, we provided numerical results for a setting that is not covered by SE. The results showed that damping [17] can improve the convergence performance of C-MP-AMP.

[3] AMP can achieve the MMSE in the limit of large linear systems when the model parameters (n/N, signal to noise ratio, sparsity of the unknown $\mathbf{x}$) are within a region [16].

6. REFERENCES

[1] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning*, Springer, Aug. 2001.

[2] Y. Zhou, U. Porwal, C. Zhang, H. Ngo, L. Nguyen, C. Ré, and V. Govindaraju, "Parallel feature selection inspired by group testing," in *Neural Inf. Process. Syst. (NIPS)*, Dec. 2014, pp. 3554–3562.

[3] X. Wang, D. Dunson, and C. Leng, "DECOrrelated feature space partitioning for distributed sparse regression," *Arxiv preprint arXiv:1602.02575*, Feb. 2016.

[4] Z. Peng, M. Yan, and W. Yin, "Parallel and distributed sparse optimization," in *Proc. IEEE 47th Asilomar Conf. Signals, Syst., and Comput.*, Nov. 2013, pp. 659–646.

[5] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, Mar. 2009.

[6] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing," *Proc. Nat. Academy Sci.*, vol. 106, no. 45, pp. 18914–18919, Nov. 2009.

[7] M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 764–785, Feb. 2011.

[8] C. Rush and R. Venkataramanan, "Finite-sample analysis of approximate message passing," *Proc. Int. Symp. Inf. Theory (ISIT)*, June 2016.

[9] C. Rush and R. Venkataramanan, "Finite-sample analysis of approximate message passing," *Arxiv preprint arXiv:1606.01800*, June 2016.

[10] P. Han, R. Niu, M. Ren, and Y. C. Eldar, "Distributed approximate message passing for sparse signal recovery," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Atlanta, GA, Dec. 2014, pp. 497–501.

[11] P. Han, R. Niu, and Y. C. Eldar, "Modified distributed iterative hard thresholding," in *IEEE Int. Conf. Acoustics, Speech, Signal Process. (ICASSP)*, Brisbane, Australia, Apr. 2015, pp. 3766–3770.

[12] P. Han, J. Zhu, R. Niu, and D. Baron, "Multi-processor approximate message passing using lossy compression," in *IEEE Int. Conf. Acoustics, Speech, Signal Process. (ICASSP)*, Shanghai, China, Mar. 2016.

[13] J. Zhu, A. Beirami, and D. Baron, "Performance trade-offs in multi-processor approximate message passing," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Barcelona, Spain, July 2016.

[14] J. Zhu, D. Baron, and A. Beirami, "Optimal trade-offs in multi-processor approximate message passing," *Arxiv preprint arXiv:1601.03790v2*, Nov. 2016.

[15] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. Motivation and construction," in *IEEE Inf. Theory Workshop*, Jan. 2010.

[16] F. Krzakala, M. Mézard, F. Sausset, Y. Sun, and L. Zdeborová, "Probabilistic reconstruction in compressed sensing: Algorithms, phase diagrams, and threshold achieving matrices," *J. Stat. Mech. - Theory E.*, vol. 2012, no. 08, pp. P08009, Aug. 2012.

[17] S. Rangan, P. Schniter, and A. Fletcher, "On the convergence of approximate message passing with arbitrary matrices," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, July 2014, pp. 236–240.

[18] D. Guo and S. Verdú, "Randomly spread CDMA: Asymptotics via statistical physics," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 1983–2010, June 2005.

[19] J. Vila, P. Schniter, S. Rangan, F. Krzakala, and L. Zdeborová, "Adaptive damping and mean removal for the generalized approximate message passing algorithm," in *IEEE Int. Conf. Acoustics, Speech, Signal Process. (ICASSP)*, Apr. 2015, pp. 2021–2025.