# Min-Sum-Min Message-Passing for Quadratic Optimization

Guoqiang Zhang and Richard Heusdens
*Department of Mediamatics*
*Delft University of Technology*
*Delft, the Netherlands*
*Email: {g.zhang-1,r.heusdens}@tudelft.nl*

***Abstract*—We study the minimization of a quadratic objective function in a distributed fashion. It is known that the min-sum algorithm can be applied to solve the minimization problem if the algorithm converges. We propose a min-sum-min message-passing algorithm which includes the min-sum algorithm as a special case. As the name suggests, the new algorithm involves two minimizations in each iteration as compared to the min-sum algorithm which has one minimization. The algorithm is derived based on a new closed-loop quadratic optimization problem which has the same optimal solution as the original one. Experiments demonstrate that our algorithm improves the convergence speed of the min-sum algorithm by properly selecting a parameter in the algorithm. Furthermore, we find empirically that in some situations where the min-sum algorithm fails, our algorithm still converges to the right solution. Experiments show that if our algorithm converges, our algorithm outperform a reference method with fast convergence speed.**

***Keywords*-Distributed optimization, Gaussian belief propagation, message-passing algorithms**

## I. Introduction

In this paper we consider minimizing a quadratic optimization problem, namely

$$\min_{x\in\mathbb{R}^n} f(x) = \frac{1}{2}x^TJx - h^Tx, \tag{1}$$

where $J \in \mathbb{R}^{n\times n}$ is a positive definite matrix and $h \in \mathbb{R}^n$. It is known that the optimal solution $x^*$ satisfies a linear equation

$$Jx^* = h.$$

We suppose that the matrix $J$ is sparse and the dimensionality $n$ is large. In this situation, the direct computation (without using the sparse structure of $J$) of the optimal solution may be expensive and unscalable. One natural question is how to exploit the sparse geometry to efficiently obtain the optimal solution. To achieve this goal, the quadratic function $f(x)$ can be associated with an undirected graph $G = (V, E)$. That is, the graph has a node for each variable $x_i$ and an edge between $i$ and $j$ for each nonzero $J_{ij}$ term. The algorithms that exploit the sparse geometry exchange information between nodes in the graph until reaching consensus.

Existing algorithms are either applicable to a specific class of $J$ or are computationally expensive (which we will explain in detail in next section). Our work will focus on designing an efficient distributed message-passing algorithm for a general positive definite matrix $J$.

The reminder of the paper is organized as follows. In Section II, we provide a literature review. Section III briefly describes the GaBP algorithm, or equivalently, the min-sum algorithm for quadratic optimization. In Section IV, we present our new min-sum-min message-passing algorithm. Section V provides the experimental results. Finally, we draw conclusions in Section VI.

## II. Related Work

The quadratic optimization problem is closely related to the Gaussian belief propagation (GaBP) for inference in graphic models. This is due to the fact that $f(x)$ can be associated with a Gaussian distribution $p(x)$ via

$$p(x) \propto \exp(-(1/2)x^TJx + h^Tx).$$

The mean value of $p(x)$ is the same as the optimal solution of the quadratic optimization problem. The GaBP algorithm is a min-sum message-passing algorithm for estimating the mean of the Gaussian random vector. Due to its simplicity, the GaBP algorithm has found many applications in practice, such as signal processing [1][2], consensus propagation in sensor networks [3], multiuser detection [4] and Turbo decoding with Gaussian densities [5]. It is known that if the GaBP algorithm converges, it converges to the mean value of $p(x)$ (see [6],[7]). Unfortunately, the GaBP algorithm does not always converge, which limits its application. Two general sufficient conditions for the convergence of the GaBP algorithm are established: diagonal dominance of $J$ [8] and walk-summability of $J$ [9][6]. For completeness, we give their definitions in the following.

**Definition** [8],[10] A matrix $J \in \mathbb{R}^{n\times n}$, with all ones on its diagonal, is walk-summable if the spectral radius of the matrix $\bar{J} - I$, where $\bar{J} = [|J_{ij}|]_{i,j=1}^n$ , is less than one.

**Definition** [10] A matrix $J \in \mathbb{R}^{n\times n}$ is diagonally dominant if $|J_{ii}| > \sum_{j\neq i} |J_{ij}|$ for all $i$.

Recently research attention has moved to overcome the convergence-failure of the GaBP algorithm for a general matrix $J$. In [10], Ruozzi and Tatikonda proposed a variant

 ISBN: 978-1-61208-161-8 

of the GaBP algorithm by changing the edge structure of the graph. In their algorithm, two parameters have been introduced to ensuring the correct convergence. However, it is not clear how to choose the two parameters. In [11], Johnson et al. proposed a double-loop algorithm with the GaBP algorithm as a subroutine (corresponds to the inner loop). Each time the GaBP algorithm is called a better estimate of the mean vector is obtained. The double-loop algorithm guarantees the convergence at the cost of high computational complexity. The basic idea of the double-loop algorithm is to precondition the matrix $J$ such that the new matrix is diagonal dominant, allowing the use of the GaBP algorithm.

In this paper we generalize the min-sum algorithm by proposing a new min-sum-min algorithm. We first construct a new closed-loop quadratic optimization problem which has the same optimal solution as that of the original problem. Instead of solving the original problem, we solve the new problem by developing the min-sum-min algorithm. The basic idea behind the algorithm is to transform the closed-loop optimization problem into $n$ scalar closed-loop optimization problems, one for each node. Note that our algorithm has two minimizations for each iteration as compared to the min-sum algorithm which has one minimization. The additional minimization in our algorithm serves to break the loop at each node.

We test our algorithm for two scenarios. When the min-sum algorithm converges, we find that our algorithm can be more efficient than the min-algorithm by properly choosing a parameter in the algorithm. When the min-sum algorithm fails, we find that our algorithm still converges in some situations. Experiments show that our algorithm significantly improves the convergence speed of the double-loop algorithm [11].

## III. MIN-SUM MESSAGE-PASSING

In this section we briefly review the min-sum message-passing algorithm for quadratic optimization (which is actually the GaBP algorithm). This algorithm is the basis for developing our new min-sum-min message-passing algorithm.

Before considering the quadratic optimization problem (1), we first study a more general objective function $f(x)$, which takes the form

$$f(x) = \sum_{j\in V} f_j(x_j) + \sum_{(i,j)\in E} f_{ij}(x_i, x_j). \tag{2}$$

In the literature, $f_j$ and $f_{ij}$ are often called self-potentials and edge potentials, respectively. $f_{ij}$ captures the correlation between nodes $i$ and $j$. Due to the pairwise correlations, the $j$th component $x_j^*$ of $x^*$ that minimizes $f(x)$ requires a global knowledge of $f(x)$. The min-sum algorithm describes the form of the messages exchanged between the nodes. Specifically, the message sent from node $i$ to node $j$ at iteration $t+1$ takes the form

$$\begin{aligned} m_{i\to j}^{(t+1)}(x_j) \;=\; \kappa + \min_{x_i}\Bigg( & f_i(x_i) \\ & + f_{ij}(x_i, x_j) + \sum_{u\in N(i)\setminus j} m_{u\to i}^{(t)}(x_i)\Bigg), \end{aligned} \tag{3}$$

where $N(i)$ denotes the set of neighboring nodes of $i$, i.e., $N(i) = \{j|(i,j)\in E\}$. The parameter $\kappa$ in (3) represents an arbitrary offset term that may be different from message to message. (3) implies that there are two messages associated with each edge $(i,j) \in E$, one for each direction on the edge. To facilitate the performance analysis, we introduce a directed graph $\vec{G} = (V, \vec{E})$ for $G$. For every edge $(i,j) \in E$, there are two elements $(i \to j)$, $(j \to i)$ in $\vec{E}$.

At each time $t$, each vertex $j$ forms a local belief function $f_j^{(t)}(x_j)$ by combining messages received from all neighbors

$$f_j^{(t)}(x_j) = f_j(x_j) + \sum_{u\in N(j)} m_{u\to j}^{(t)}(x_j). \tag{4}$$

An estimate of the $j$th component $x_j^*$ is then given by

$$\hat{x}_j^{(t)} = \arg\min_{x_j} f_j^{(t)}(x_j). \tag{5}$$

The min-sum algorithm is successful if $\hat{x}_j^{(\infty)}$ is equal to $x_j^*$ for all $j \in V$.

When the function $f(x)$ in (2) is specified to the quadratic function as in (1), the min-sum algorithm becomes the GaBP algorithm. In this situation, the self-potentials and edge potentials are given by

$$\begin{aligned} f_j(x_j) &= (1/2)J_{jj}x_j^2 - h_j x_j \\ f_{ij}(x_i, x_j) &= J_{ij}x_i x_j. \end{aligned}$$

Without loss of generality, we may assume that $J$ is normalized to have unit diagonal, i.e., $J_{jj} = 1$. Since the functions $f_j$ and $f_{ij}$ are in quadratic form, the belief function $f_j^{(t)}$ also takes a quadratic form [7]:

$$\begin{aligned} f_j^{(t)}(x_j) \;&=\; \frac{1}{2}\left(1 - \sum_{i\in N(j)} J_{ij}^2\gamma_{ij}^{(t)}\right)x_j^2 \\ &\quad - \left(h_j - \sum_{i\in N(j)} z_{ij}^{(t)}\right)x_j, \end{aligned} \tag{6}$$

where $\gamma_{ij}^{(t)}$ and $z_{ij}^{(t)}$ are updated as

$$\gamma_{ij}^{(t+1)} = \frac{1}{1 - \sum_{u\in N(i)\setminus j} J_{ui}^2\gamma_{ui}^{(t)}}, \tag{7}$$

$$z_{ij}^{(t+1)} = \frac{J_{ij}}{1 - \sum_{u\in N(i)\setminus j} J_{ui}^2\gamma_{ui}^{(t)}}\left(h_i - \sum_{u\in N(i)\setminus j} z_{ui}^{(t)}\right). \tag{8}$$

 ISBN: 978-1-61208-161-8 

The update of $\gamma_{ij}^{(t+1)}$ and $z_{ij}^{(t+1)}$ are valid if $\sum_{u\in N(i)\backslash j} J_{ui}^2\gamma_{ui}^{(t)} < 1$ for all $i$, $j$ and $t$. These inequalities are always satisfied under the walk-summability condition or the diagonal dominant condition [8],[10]. Given the form of the belief function $f_j^{(t)}(x_j)$ in (6), the estimate of $x_j^*$ is obtained by applying (5)

$$\hat{x}_j^{(t)} = \frac{1}{1-\sum_{i\in N} J_{ij}^2\gamma_{ij}^{(t)}}\left(h_j - \sum_{i\in N(j)} z_{ij}^{(t)}\right). \quad (9)$$

The message-updating equations (6)-(9) are described above for comparison with our algorithm in Section IV. We will explain how our min-sum-min algorithm is derived based on the min-sum messages (3)-(5).

## IV. Min-Sum-Min Message-Passing

In this section we first construct a new closed-loop quadratic minimization problem. The new problem has the same optimal solution as that of the original problem. We then propose a so-called min-sum-min message-passing algorithm for the new problem. Finally, we provide explicit message-updating expressions for solving the new problem.

### A. A New Cost Function

Based on (1), we define a new quadratic minimization problem:

$$x^* = \arg\min_x \tilde{f}(x, x^*), \quad (10)$$

where

$$\tilde{f}(x,x^*) = \frac{1}{2}x^T(sI+(1-s)J)x - [(1-s)h+sx^*]^T x, \quad (11)$$

where $s$ is a scalar parameter and $I$ is the identity matrix. Different from (1), the optimal solution $x^*$ appears on both sides of (10). Thus, (10) is in fact a closed-loop optimization problem. It is obvious that the min-sum algorithm cannot be directly applied here since $x^*$ is not known. We explain in the following how $\tilde{f}(x,x^*)$ is constructed as in (11).

Before providing the motivation for $\tilde{f}(x,x^*)$, we first show that the optimal solution of (10) is the same as that of the original minimization problem. We let $\tilde{J}_s = sI + (1-s)J$. Same as $J$, the new matrix $\tilde{J}_s$ also has unit diagonal. In order that the new optimization problem is well defined, we choose $s$ such that $\tilde{J}_s$ is positive definite. It should be noted that $s$ can be negative depending on $J$. To solve (10), we first fix $x^*$ in $\tilde{f}(x,x^*)$. We then set the first derivative of $\tilde{f}(x,x^*)$ w.r.t. $x$ to be 0. By doing so, we have

$$\begin{aligned}[sI+(1-s)J]\,x^* &= (1-s)h + sx^*,\\ Jx^* &= h.\end{aligned}$$

Thus instead of solving (1), we can solve the new optimization problem.

Note that the introduction of $sx^*$ in constructing $\tilde{f}(x,x^*)$ is the key point in designing a new message-passing algorithm. Due to the simple form of $sx^*$, the self-potentials and edge potentials of $\tilde{f}(x,x^*)$ also take a simple form:

$$\begin{aligned}\tilde{f}_j(x_j,x_j^*) &= (1/2)x_j^2 - [(1-s)h_j + sx_j^*]x_j, \quad (12)\\ \tilde{f}_{ij}(x_i,x_j) &= (1-s)J_{ij}x_ix_j. \quad (13)\end{aligned}$$

We point out that the self-potential $\tilde{f}_j(x_j,x_j^*)$ has only $x_j^*$ involved instead of the whole vector $x^*$. This property of $\tilde{f}_j(x_j,x_j^*)$ makes it possible for node $j$ to deal with $x_j^*$ locally. In fact, one can introduce $\Gamma x^*$, where $\Gamma$ is a diagonal matrix, in constructing a new closed-loop function (the matrix $\tilde{J}_s$ should be changed accordingly). For the same reason, one can also design a massage-passing algorithm. In this paper we focus on $sx^*$ for simplicity.

We point out that the diagonal-loading on $J$ to obtain $\tilde{J}_s$ is inspired by the work in [11]. The main difference between our work and [11] is that we propose a new message-passing algorithm based on (10). On the other hand, the authors in [11] took the min-sum algorithm as a subroutine to solve (1) directly.

### B. Algorithm design

In this subsection we present the min-sum-min algorithm for solving the closed-loop optimization problem (10). We show that one of the two minimizations of the algorithm unlocks the loopy effect of $x^*$ in (10).

In order to tackle the unknown parameters $x_j^*$ in self-potentials $\tilde{f}_j(x_j,x_j^*)$, we first revisit the min-sum algorithm as described by (3)-(5). Note that after each iteration of message-passing, an estimate $\hat{x}_j^{(t)}$ of $x_j^*$ can be obtained from the local belief function $f_j^{(t)}(x_j)$. In other words, a new estimate of $x_j^*$ is always accessible to node $j$ after each iteration. Inspired by this property of the min-sum algorithm, we propose to compute an estimate of $x_j^*$ in (12) at each iteration in designing our new algorithm. We then take the estimate of $x_j^*$ for message-updating in the next iteration. In principle, if the estimate of $x_j^*$ becomes more and more accurate as the information diffuses through message-passing, the algorithm converges to the right solution.

Based on the above analysis, we propose new message-updating expressions as

$$\hat{x}_i^{(t)} = \arg\min_{x_i}\left(\tilde{f}_i(x_i,\hat{x}_i^{(t)}) + \sum_{u\in N(i)} \tilde{m}_{u\to i}^{(t)}(x_i)\right), (14)$$

$$\check{x}_i^{(t)} = g_i\left(\hat{x}_i^{(t)}, \hat{x}_u^{(t)}, u\in N(i)\right), \quad (15)$$

$$\tilde{m}_{i\to j}^{(t+1)}(x_j) = \kappa + \min_{x_i}\left(\tilde{f}_i(x_i,\check{x}_i^{(t)}) + \tilde{f}_{ij}(x_i,x_j) + \sum_{u\in N(i)\backslash j}\tilde{m}_{u\to i}^{(t)}(x_i)\right). \quad (16)$$

Note that there are two minimization operations in (14)-(16) for each iteration, as compared to (3) which has only one minimization. The name *min-sum-min* for our new algorithm then arises naturally. The first minimization in (14) comes from (5) and (12). This minimization plays an important role in breaking the loop in (10). The second minimization in (16) comes from the min-sum algorithm. The function $g_i$ in (15) is utilized to refine the estimate using the outputs of the first minimization. (14)-(15) together provide an estimate of $x_i^*$ for node $i$ at each iteration.

Note that (14) is again a closed-loop minimization with respect to $\hat{x}_i^{(t)}$. Thus we successfully transform the global closed-loop optimization problem into $n$ local closed-loop optimization problems, one for each node. As all the messages are in quadratic form, it is not difficult to compute $\hat{x}_i^{(t)}$ after each iteration. Once $\{g_i|i \in V\}$ is specified, we effectively provide a min-sum-min algorithm to solve (10).

We can also interpret (14)-(16) from another viewpoint. Note that (14)-(15) combines information from neighboring nodes in computing an estimate of the optimal solution. Thus (14)–(15) can be viewed as an information-fusion step. On the other hand, the second minimization (16) carries information from node $i$ to a neighboring node $j$. Correspondingly, (16) can be viewed as an information-diffusion step. The two steps are implemented in order until reaching consensus at each individual node. That is, the estimate $\check{x}_i^{(t)}$ of the optimal component $x_i^*$ is stable over time for all $i$.

**Remark 1**: The min-sum-min algorithm is a natural extension of the min-sum algorithm. To see this, we let $s$ approach to $0$, it is immediate that $\tilde{m}_{ij}(x_j) \to m_{ij}(x_j)$. Since our algorithm has a free parameter $s$ to choose, we can improve the performance of the algorithm by properly adjusting the parameter.

**Remark 2**: Note that the min-sum-min algorithm is not limited to the quadratic minimization problem. In fact, as long as an original optimization problem can be reformulated into a proper closed-loop optimization problem, the min-sum-min algorithm can be applied in correspondence.

### C. Explicit message-updating expressions

In this subsection we provide explicit message-updating expressions for solving the closed-loop optimization problem. We study the three updating expressions (14)-(16) one by one for the quadratic form of the potentials (12)-(13).

We first consider the minimization (14). We suppose that the message $\tilde{m}_{u\to i}^{(t)}(x_i)$ at iteration $t$ takes the form

$$\tilde{m}_{u\to i}^{(t)}(x_i) = -\frac{1}{2}(1-s)^2 J_{ui}^2 \tilde{\gamma}_{ui}^{(t)} x_i^2 + \tilde{z}_{ui}^{(t)} x_i, \qquad (17)$$

where $\tilde{\gamma}_{ui}^{(t)}$ and $\tilde{z}_{ui}^{(t)}$ are the associated parameters characterizing the quadratic form. By plugging (17) into (14), we obtain

$$\begin{aligned} \hat{x}_i^{(t)} \;=\; & \arg\min_{x_i} \Bigg( \frac{1}{2}\Big[1 - \sum_{u\in N(i)} (1-s)^2 J_{ui}^2 \tilde{\gamma}_{ui}^{(t)}\Big] x_i^2 \\ & - \Big[(1-s)h_i + s\hat{x}_i^{(t)} - \sum_{u\in N(i)} \tilde{z}_{ui}^{(t)}\Big] x_i \Bigg). \qquad (18) \end{aligned}$$

The optimal solution $\hat{x}_i^{(t)}$ can be easily computed from (18), expressed as

$$\hat{x}_i^{(t)} = \frac{(1-s)h_i - \sum_{u\in N(i)} \tilde{z}_{ui}^{(t)}}{1 - s - \sum_{u\in N(i)} (1-s)^2 J_{ui}^2 \tilde{\gamma}_{ui}^{(t)}}. \qquad (19)$$

Given the expression for $\hat{x}_i^{(t)}$, we then specify the function set $\{g_i|i \in V\}$ in (15). To achieve this goal, we construct an equality

$$x^* = \frac{1}{2}\Big(h + x^* - (J - I)x^*\Big), \qquad (20)$$

where $x^*$ is the optimal solution to the minimization problem. Based on (20), we then let $g_i$ be

$$\check{x}_i^{(t)} = \frac{1}{2}\Big(h_i + \hat{x}_i^{(t)} - \sum_{u\in N(i)} J_{iu}\hat{x}_u^{(t)}\Big) \quad \forall i \in V. \qquad (21)$$

The new estimate $\check{x}_i^{(t)}$ is obtained by combining information from neighboring nodes and the node itself. The update expression (21) is just one instance of $g_i$. In principle, there are many ways to construct the function $g_i$ by building new equalities in terms of $x^*$.

Upon obtaining the expression for $\check{x}_i^{(t)}$, we study the second minimization (16). Again by plugging (17) into (16), we obtain

$$\begin{aligned} & \tilde{m}_{i\to j}^{(t)}(x_j) \\ & = \kappa + \min_{x_i} \Bigg( \frac{1}{2}\Big[1 - \sum_{u\in N(i)\backslash j} (1-s)^2 J_{ui}^2 \tilde{\gamma}_{ui}^{(t+1)}\Big] x_i^2 - \Big[s\check{x}_i^{(t)} \\ & \quad + (1-s)h_i - (1-s)J_{ij}x_j - \sum_{u\in N(i)\backslash j} \tilde{z}_{ui}^{(t)}\Big] x_i \Bigg), \end{aligned}$$

where $\check{x}_i^{(t)}$ is given by (21). We then simplify $\tilde{m}_{i\to j}^{(t)}(x_j)$ by solving the minimization. The resulting expression takes the from:

$$\tilde{m}_{i\to j}^{(t+1)}(x_j) = -\frac{1}{2}(1-s)^2 J_{ij}^2 \tilde{\gamma}_{ij}^{(t+1)} x_j^2 + \tilde{z}_{ij}^{(t+1)} x_j + \kappa',$$

where

$$\tilde{\gamma}_{ij}^{(t+1)} = \frac{1}{1 - \sum_{u\in N(i)\backslash j} (1-s)^2 J_{ui}^2 \tilde{\gamma}_{ui}^{(t)}}, \qquad (22)$$

 ISBN: 978-1-61208-161-8 

and

$$\tilde{z}_{ij}^{(t+1)} = \frac{(1-s)J_{ij}\Big((1-s)h_i + s\check{x}_i^{(t)} - \sum_{u\in N(i)\backslash j}\tilde{z}_{ui}^{(t)}\Big)}{1-\sum_{u\in N(i)\backslash j}(1-s)^2J_{ui}^2\tilde{\gamma}_{ui}^{(t)}}, \quad (23)$$

and $\kappa'$ is a new constant. $\tilde{m}_{i\to j}^{(t+1)}(x_j)$ again takes a quadratic form, which is consistent with (17).

One observes that $\tilde{\gamma}_{ij}^{(t+1)}$ and $\gamma_{ij}^{(t+1)}$ essentially take the same message-passing form. The only difference between them is that $\tilde{\gamma}_{ij}^{(t+1)}$ is derived from $\tilde{J}_s$ while $\gamma_{ij}^{(t+1)}$ is derived from $J$. Thus as long as $s$ is chosen such that $\tilde{J}_s$ is diagonal dominant or walk-summable, $\tilde{\gamma}_{ij}^{(t+1)}$ always converges.

The parameter $\tilde{z}_{ij}^{(t+1)}$ has an additional term $s\check{x}_i^{(t)}$ compared to $z_{ij}^{(t+1)}$. This additional term $s\hat{x}_i^{(t)}$ is an estimate of $sx_i^*$ in the self-potential (12). If $z_{ij}^{(t)}$ converges, the min-sum-min algorithm then converges to the right solution.

| | Stage | Operation |
|---|---|---|
| 1 | *Initialize* | Choose a value $s$;<br>Set $\tilde{\gamma}_{ij}=0$ and $\tilde{z}_{ij}=0$, $\forall(i\to j)\in\vec{E}$ |
| 2 | *Iterate* | For all $(i\to j)\in\vec{E}$<br>Update $\hat{x}_i$ using (19)<br>Update $\check{x}_i$ using (21)<br>Update $\tilde{\gamma}_{ij}$ using (22)<br>Update $\tilde{z}_{ij}$ using (23)<br>End |
| 3 | *Check* | If $\{\check{x}_i\}, \{\tilde{\gamma_{ij}}\}$ and $\{\tilde{z}_{ij}\}$ become stable, go to 4; else, return to 2. |
| 4 | *Output* | Return $\check{x}_i$, $\forall i$. |

Table I
MIN-SUM-MIN MESSAGE-PASSING FOR COMPUTING $x^* = \arg\min_x \frac{1}{2}x^TJx - h^Tx$.

Based on the above analysis, we briefly summarize the min-sum-min algorithm for the quadratic minimization (1) in Table I. In the algorithm, we choose the parameter $s$ such that $\tilde{J}_s$ is diagonal dominant. This guarantees that $\tilde{\gamma}_{ij}$ converge for all $(i\to j)\in\vec{E}$.

## V. EXPERIMENTAL RESULTS

In the experiment, we test the convergence speed of the min-sum-min algorithm. We study two scenarios: the one where the min-sum algorithm converges and the other one where the min-sum algorithm fails.

We considered two graphs for constructing $J$ as shown in Fig. 1. Graph (a) is a 4-cycle with a chord. Graph (b) is a 5-cycle. For each graph, the matrix $J$ is constructed with its diagonal elements being 1 and its off-diagonal elements being the edge weights as described in the graph. The $h$ vector in (1) for the two graphs are $h = [\ 1\ \ 2\ \ 1\ \ 2\ ]^T$ and $h = [\ 1\ \ 2\ \ 1\ \ 2\ \ 1\ ]^T$, respectively.

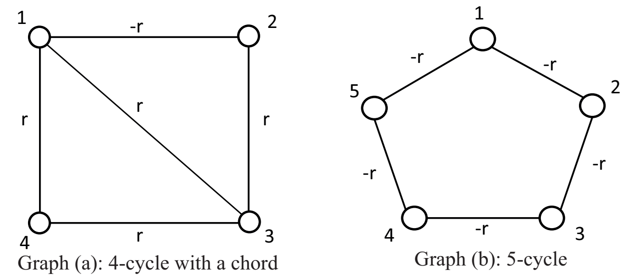


Figure 1. The two graphs for constructing $J$. The edge weights are as denoted by $-r$ or $r$ in the two graphs.

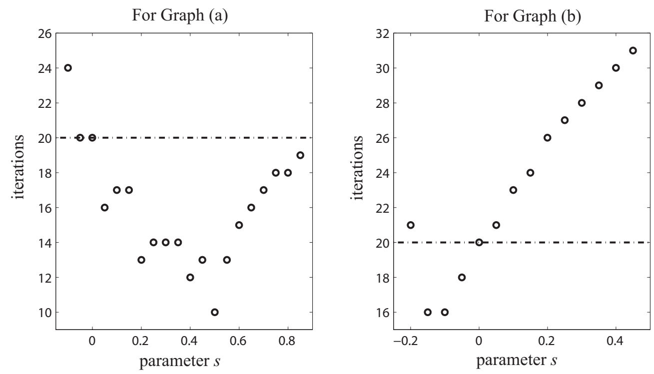


Figure 2. Effect of diagonal loading on the convergence speed. We use the symbol "$\circ$" to denote the number of iterations required for different values of $s$ in the min-sum-min algorithm. For comparison, we use the dash-dot line to denote the number of iterations required for the min-sum algorithm.

### A. Comparison between the min-sum-min and the min-sum algorithms

In the first experiment, we investigate the scenario where the min-sum algorithm converges. We take the min-sum algorithm as a reference for performance comparison.

We set $r = 0.34$ and $r = 0.4$ in Graph (a) and (b), respectively. Correspondingly, we obtain two realizations for $J$. The spectral radius of $\bar{J} - I$, where $\bar{J} = [|J_{ij}|]_{i,j=1}^n$, are 0.8709 (for (a)) and 0.8 (for (b)). Thus the $J$ matrices satisfy the walk-summable condition.

In the implementation, we chose the criterion for terminating the algorithm to be $\frac{1}{n}\sum_{i=1}^n|\check{x}_i^{(t)} - x_i^*| \le 10^{-5}$. We selected the parameter $s$ between -0.2 and 1 for our algorithm.

The experiment results are as shown in Fig. 2. Surprisingly, we observe that for a range of $s$ values, the min-sum-min algorithm outperforms the min-sum algorithm in both cases. The results suggest that there exist more efficient algorithms than the min-sum algorithm. The min-sum-min algorithm is one example in improving the convergence speed. We also tested other values $r$ in Fig 1. The results are similar to those in Fig. 2.


ISBN: 978-1-61208-161-8

### *B. Comparison between the min-sum-min and the double-loop algorithms*

In the second experiment, we investigate the scenario where the min-sum algorithm fails. We take the double-loop algorithm [11] as a reference for performance comparison.

In this situation, we set $r = 0.45$ and $r = -0.52$ in Graph (a) and (b), respectively. Correspondingly, the spectral radius of $\bar{J} - I$, are 1.1527 (for (a)) and 1.04 (for (b)). The $J$ matrices are not walk-summable anymore.

Again we chose the criterion for terminating the algorithm to be $\frac{1}{n}\sum_{i=1}^{n}|\tilde{x}_i^{(t)} - x_i^*| \leq 10^{-5}$. In implementing the double-loop algorithm, we had to setup one more criterion for the inner-loop iteration. We terminated the inner-loop each time when $\frac{1}{n}\sum_{i=1}^{n}|\hat{x}_i^{(t)} - \hat{x}_i^{(t-1)}| \leq 10^{-5}$.

Fig. 3 and Fig. 4 display the experiment results for Graph (a) and (b), respectively. It is seen from the figures that if our algorithm converges, it converges much faster than the double-loop algorithm. The performance gain in terms of the number of iterations range from hundreds to thousands in the experiment.

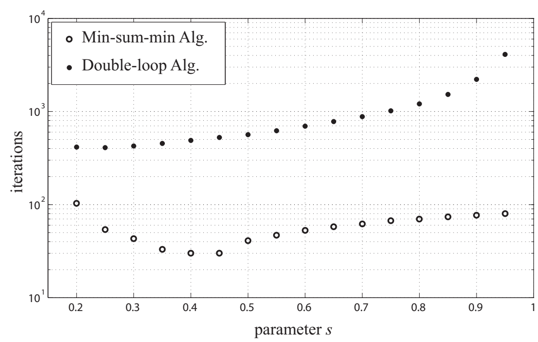


Figure 3. Comparison between the min-sum-min algorithm and the double-loop algorithm for Graph (a).

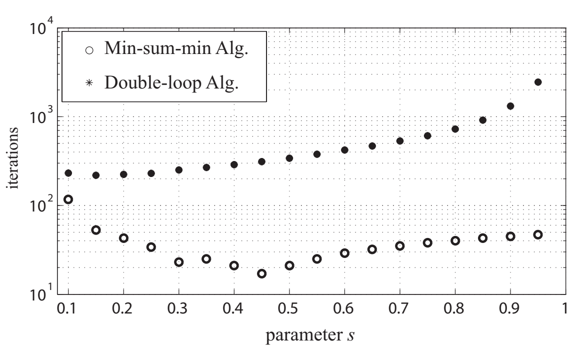


Figure 4. Comparison between the min-sum-min algorithm and the double-loop algorithm for Graph (b).

## VI. Conclusion

We have proposed a new min-sum-min message-passing algorithm which includes the min-sum algorithm as a special case. The new algorithm has been derived based on a closed-loop optimization problem which has the same optimal solution as the original problem. Compared to the min-sum algorithm, the min-sum-min algorithm has a free parameter $s$ to choose. This property renders two advantages of the min-sum-min algorithm over the min-sum algorithm. First, the min-sum-min algorithm provides faster convergence speed when the parameter $s$ is chosen properly. Second, in some situations where the min-sum algorithm fails, the min-sum-min algorithm still converges.

One open issue is how to choose the parameter $s$ to make our algorithm most efficient. This issue is quite relevant to engineering in practice.

## References

[1] D. Bickson, O. Shental, and D. Dolev, "Distributed Kalman Filter via Gaussian Belief Propagation," in *the 46th Allerton Conf. on Communications, Control and Computing*, 2008.

[2] H. A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, "The Factor Graph Approach to Model-Based Signal Processing," in *Proceedings of the IEEE*, vol. 95, 2007, pp. 1295–1322.

[3] C. C. Moallemi and B. V. Roy, "Consensus Propagation," *IEEE Trans. Inf. Theory*, vol. 52, no. 11, pp. 4753–4766, 2006.

[4] A. Montanari, B. Prabhakar, and D. Tse, "Belief Propagation Based Multi-User Detection," in *Proc. 43rd Allerton Conf. on Communications, Control and Computing*, 2005.

[5] P. Rusmevichientong and B. B. Roy, "An analysis of Belief Propagation on the Turbo Decoding Graph with Gaussian Densities," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 745–765, 2001.

[6] D. M. Malioutov, J. K. Johnson, and A. S. Willsky, "Walk-Sums and Belief Propagation in Gaussian Graphical Models," *J. Mach. Learn. Res.*, vol. 7, pp. 2031–2064, 2006.

[7] C. C. Moallemi and B. V. Roy, "Convergence of Min-Sum Message Passing for Quadratic Optimization," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2413–2423, 2009.

[8] Y. Weiss and W. T. Freeman, "Correctness of Belief Propagation in Gaussian Graphical Models of Arbitrary Topology," *Neural Computation*, vol. 13, pp. 2173–2200, 2001.

[9] J. K. Johnson, D. M. Malioutov, and A. S. Willsky, "Walk-sum Interpretation and Analysis of Gaussian Belief Propagation," in *Advances in Neural Information Processing Systems*, vol. 18, Cambridge, MA: MIT Press, 2006.

[10] N. Ruozzi and S. Tatikonda, "Unconstrained Minimization of Quadratic Functions via Min-Sum," in *Proceedings of thee Conference on Information Sciences and systems (CISS)*, March 2010.

[11] J. K. Johnson, D. Bickson, and D. Dolev, "Fixing Convergence of Gaussian Belief Propagation," in *the International Symposium on Information Theory*, 2009.