

GENERALIZED LINEAR COORDINATE-DESCENT MESSAGE-PASSING FOR CONVEX 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

In this paper we propose a generalized linear coordinate-descent (GLiCD) algorithm for a class of unconstrained convex optimization problems. The considered objective function can be decomposed into edge-functions and node-functions of a graphical model. The messages of the GLiCD algorithm are in a form of linear functions, as compared to the min-sum algorithm of which the form of messages depends on the objective function. Thus, the implementation of the GLiCD algorithm is much simpler than that of the min-sum algorithm. A theorem is stated according to which the algorithm converges to the optimal solution if the objective function satisfies a diagonal-dominant condition. As an application, the GLiCD algorithm is exploited in solving the averaging problem in sensor networks, where the performance is compared to that of the min-sum algorithm.

Index Terms— Convex optimization, message passing, coordinate descent

1. INTRODUCTION

Consider an optimization problem where the objective function can be decomposed according to a undirected graph $G = (V, E)$ so that

$$\min_{x \in \mathcal{X}^{|V|}} f(x) = \min_{x \in \mathcal{X}^{|V|}} \sum_{(i,j) \in E} f_{ij}(x_i, x_j) + \sum_{i \in V} f_i(x_i). \quad (1)$$

Every variable x_i in the problem is associated with a node $i \in V$ and takes values in \mathcal{X} . The variables contribute to the objective function by interacting with their neighboring ones. Such problem formulation has found many applications in practice, such as digital communications [1], image processing [2], multi-user detection [3], and consensus propagation in sensor networks [4]. In many applications of interest, the graph is sparse. The research challenge is how to exploit the sparse geometry to efficiently obtain the optimal solution.

In this work, we consider the case where $\mathcal{X} = \mathbb{R}$, and the optimization problem is continuous. In the literature, not much progress has been obtained on designing efficient message-passing algorithms for solving the general continuous problem (1). Instead, researchers have focused on some special forms of (1). One instance is the problem with local quadratic functions $\{f_{ij}(\cdot, \cdot)\}$ and $\{f_i(\cdot)\}$. For the quadratic optimization problem, the min-sum algorithm is known to compute the optimal solution when it converges [5, 6]. A sufficient condition of the min-sum algorithm for convergence has been identified in [7, 8]. Due to the quadratic form of the local functions, the messages of the min-sum algorithm are also in a quadratic form. In [9], the linear coordinate-descent (LiCD) algorithm was proposed for the quadratic optimization, where the messages are in a linear

form. Further, the LiCD algorithm was shown to have the same sufficient condition for convergence as that of the min-sum algorithm.

Recent work by Moallemi and Roy [10] applied the min-sum algorithm to an unconstrained convex optimization problem. That is, the local functions are convex. The authors established a sufficient condition for the algorithm convergence. However, the form of the messages depends on the objective function. In other words, if the function is described by a set of parameters, the description of the messages may also require the same number of parameters as that of the set. This property may impose high communication cost which is undesirable in, e.g., sensor-network related problems. It is, therefore, of great interest to study whether there exists an efficient message-passing algorithm of which the implementation-complexity is much less influenced by the objective function.

In this paper we generalize the LiCD algorithm for solving unconstrained convex optimization problems. At each iteration, the generalized LiCD (GLiCD) algorithm updates messages based on feedback from previous iteration. On the other hand, the LiCD algorithm involves no feedback. The GLiCD algorithm maintains the property that the messages are in a linear form. Thus, the algorithm always has a simple implementation irrespective of the complexity of the objective function.

We establish a sufficient condition for the convergence of the GLiCD algorithm. In particular, when the objective function satisfies a diagonal-dominant condition, the GLiCD algorithm converges to the optimal solution. In addition, we apply the GLiCD algorithm to the averaging problem in sensor networks. We follow the line of work in [4], where the min-sum algorithm is utilized in solving the same problem. In the experiment, we study the efficiency of the GLiCD algorithm with the min-sum algorithm as a reference. Experimental results demonstrate that the algorithm efficiency improves as the associated graph G becomes dense.

2. PROBLEM FORMULATION

In [10], the pairwise separable convex program was introduced. The program specifies a broad class of unconstrained convex problems for the application of the min-sum algorithm. In this work we also consider the pairwise separable convex program, of which the definition is presented below.

Definition 2.1 [10] (Pairwise Separable Convex program): A pairwise separable convex program is an optimization problem of the form

$$\min_{x \in \mathbb{R}^{|V|}} f(x) \triangleq \sum_{i \in V} f_i(x_i) + \sum_{(i,j) \in E} f_{ij}(x_i, x_j) \quad (2)$$

where the factors $f_i(\cdot)$ are strictly convex, coercive¹, and twice con-

¹A function $h : \mathbb{R}^n \rightarrow \mathbb{R}$ is coercive if, for every sequence $\{x_k\} \subset \mathbb{R}^n$

tinuously differentiable, the factors $\{f_{ij}(\cdot, \cdot)\}$ are convex and twice continually differentiable, and

$$M \triangleq \min_{i \in V} \inf_{x \in \mathbb{R}^{|V|}} \frac{\partial^2}{\partial x_i^2} f(x) > 0.$$

The objective function $f(x)$ specified by the program is strictly convex and coercive. Thus, there exists a unique optimal solution $x^* \in \mathbb{R}^{|V|}$ that minimizes the objection function. The local functions $\{f_i(x_i)\}$ and $\{f_{ij}(x_i, x_j)\}$ in (2) are called self-potentials and edge-potentials, respectively. We use $N(i)$ to denote the set of all neighbors of node i . For each edge $(i, j) \in E$, we use $[j, i]$ and $[i, j]$ to denote its two directed edges. Correspondingly, we denote the set of all directed edges of the graph as \vec{E} .

In the following, we present the GLiCD algorithm for solving the pairwise separable convex program. We point out that the quadratic problem considered in [9] is not a special case of the pairwise separable convex program. The edge-potentials in (2) are required to be convex while in [9] there is no explicit conditions on the edge-potentials.

3. GENERALIZED LINEAR COORDINATE-DESCENT ALGORITHM

In this section we present the GLiCD algorithm in detail. As a generalization of the LiCD algorithm [9], the GLiCD algorithm incorporates feedback from last iteration in computing new messages. The amount of feedback at each node is controlled by a set of parameters, one for each neighbor.

3.1. Message updating expressions

The GLiCD algorithm attempts to minimize the objective function in an iterative, message-passing fashion. At time t , each node i keeps track of a message and an estimate of x_i^* from each neighbor $u \in N(i)$. We denote the message and the estimate from node u to i as $m_{ui}^{(t)}(x_i)$ and $\hat{x}_i^{u(t)}$, respectively. In principle, the form of the messages can be quite general. While the messages with a general form may carry much information, the corresponding algorithm may also be computationally expensive. For the GLiCD algorithm, we suppose that the message $m_{ui}^{(t)}(x_i), \forall [u, i] \in \vec{E}$, takes a linear form:

$$m_{ui}^{(t)}(x_i) = z_{ui}^{(t)} x_i, \quad t = 0, 1, \dots \quad (3)$$

The simplicity of the linear message-form is beneficial to the implementation of the algorithm.

With the messages (3), the pairwise separable function $f(x)$ in (2) can be reparameterized as

$$f(x) = \sum_{i \in V} f_i^{(t)}(x_i) + \sum_{(i,j) \in E} f_{ij}^{(t)}(x_i, x_j),$$

where the new self-potentials and edge-potentials are given by

$$f_i^{(t)}(x_i) = f_i(x_i) + \sum_{u \in N(i)} z_{ui}^{(t)} x_i, \quad (4)$$

$$f_{ij}^{(t)}(x_i, x_j) = f_{ij}(x_i, x_j) - z_{ji}^{(t)} x_i - z_{ij}^{(t)} x_j. \quad (5)$$

Thus, the objective function remains the same form irrespective of the messages. To briefly summarize, each node i at time t has the parameters $\{z_{ui}^{(t)}, \hat{x}_i^{u(t)}, u \in N(i)\}$. The estimates $\{\hat{x}_i^{u(t)}, u \in N(i)\}$ provide some information about the optimal solution x_i^* . Thus, the estimates can be used as feedback in computing new messages and estimates in next iteration.

We now consider utilizing the estimates $\{\hat{x}_i^{u(t)}\}$ as feedback in the derivation of $\{z_{ui}^{(t+1)}\}$ and $\{\hat{x}_i^{u(t+1)}\}$. The basic idea is to apply the estimates to construct a penalty function for each edge in with $\|x_k\| \rightarrow \infty, h(x_k) \rightarrow \infty$.

the graph. Then we use the constructed penalty functions and the potentials (4)-(5) in computing the new messages and estimates. In particular, we define the penalty function $g_{ij}^{(t)}(x_i, x_j)$ for $(i, j) \in E$ to be a quadratic function:

$$g_{ij}^{(t)}(x_i, x_j) = \sum_{u \in N(i) \setminus j} \frac{\eta_i^{u(t)}}{2} (x_i - \hat{x}_i^{u(t)})^2 + \sum_{v \in N(j) \setminus i} \frac{\eta_j^{v(t)}}{2} (x_j - \hat{x}_j^{v(t)})^2, \quad (6)$$

where the parameters $\eta_i^{u(t)}$ and $\eta_j^{v(t)}$ are the positive weighting factors for their corresponding quadratic terms. Note that the function $g_{ij}^{(t)}(\cdot, \cdot)$ does not involve $\hat{x}_i^{j(t)}$ and $\hat{x}_j^{i(t)}$. If the node i only has edge (i, j) , then the function $g_{ij}(x_i, x_j)$ reduces to $g_{ij}^{(t)}(x_j)$.

Upon introducing the penalty functions, we are ready to derive the updating expressions for $\{z_{uv}^{(t+1)}(x_v)\}$ and $\{\hat{x}_v^{u(t+1)}\}$. Without loss of generality, we focus on computing $\{z_{ij}^{(t+1)}, z_{ji}^{(t+1)}\}$ and $\{\hat{x}_i^{j(t+1)}, \hat{x}_j^{i(t+1)}\}$ that are associated with the edge $(i, j) \in E$. In doing so, we define a new function $L_{ij}^{(t)}(x_i, x_j)$ for $(i, j) \in E$ as

$$L_{ij}^{(t)}(x_i, x_j) \triangleq f_i^{(t)}(x_i) + f_j^{(t)}(x_j) + f_{ij}^{(t)}(x_i, x_j) + g_{ij}^{(t)}(x_i, x_j).$$

From Definition 2.1, we see that $L_{ij}^{(t)}(\cdot, \cdot)$ is strictly convex, coercive and twice continually differentiable. We minimize the function $L_{ij}^{(t)}(\cdot, \cdot)$ to compute the new estimates $\hat{x}_i^{j(t+1)}$ and $\hat{x}_j^{i(t+1)}$:

$$\left(\hat{x}_i^{j(t+1)}, \hat{x}_j^{i(t+1)} \right) = \arg \min_{x_i, x_j} L_{ij}^{(t)}(x_i, x_j).$$

Due to the decomposed form of $f(x)$, minimization of each L_{ij} is likely to minimize $f(x)$ while at the same time keeping the estimates from neighboring nodes close to each other. The partial derivatives of $L_{ij}^{(t)}(\cdot, \cdot)$ w.r.t. x_i and x_j satisfy

$$0 = \frac{d}{dx_i} f_i(\hat{x}_i^{j(t+1)}) + \sum_{u \in N(i) \setminus j} \eta_i^{u(t)} (\hat{x}_i^{j(t+1)} - \hat{x}_i^{u(t)}) + \sum_{u \in N(i) \setminus j} z_{ui}^{(t)} + \frac{\partial}{\partial x_i} f_{ij}(\hat{x}_i^{j(t+1)}, \hat{x}_j^{i(t+1)}), \quad (7)$$

$$0 = \frac{d}{dx_j} f_j(\hat{x}_j^{i(t+1)}) + \sum_{v \in N(j) \setminus i} \eta_j^{v(t)} (\hat{x}_j^{i(t+1)} - \hat{x}_j^{v(t)}) + \sum_{v \in N(j) \setminus i} z_{vj}^{(t)} + \frac{\partial}{\partial x_j} f_{ij}(\hat{x}_i^{j(t+1)}, \hat{x}_j^{i(t+1)}). \quad (8)$$

The above two equations are essentially the implicit updating expressions for $\hat{x}_i^{j(t+1)}$ and $\hat{x}_j^{i(t+1)}$, respectively.

Based on (7)-(8), we now derive the expressions for $z_{ij}^{(t+1)}$ and $z_{ji}^{(t+1)}$. Note that the first three terms on the right hand side of (7) only involves the variable x_i . A similar property can be observed from (8). It is the function $f_{ij}(x_i, x_j)$ that brings x_i and x_j together, as the last term in (7) or in (8). To separate x_i and x_j in the computation of $(\hat{x}_i^{j(t+1)}, \hat{x}_j^{i(t+1)})$, we define

$$z_{ij}^{(t+1)} = \frac{\partial}{\partial x_i} f_{ij}(\hat{x}_i^{j(t+1)}, \hat{x}_j^{i(t+1)}). \quad (9)$$

$$z_{ji}^{(t+1)} = \frac{\partial}{\partial x_j} f_{ij}(\hat{x}_i^{j(t+1)}, \hat{x}_j^{i(t+1)}). \quad (10)$$

The message $z_{ji}^{(t+1)}$ brings all the information about $\hat{x}_i^{j(t+1)}$ that is contained at node j to node i . The messages associated with other

edges in E take similar expressions as (9)-(10).

To complete the algorithm, we consider initializing the parameters $\{\hat{x}_v^{u(0)}\}$ and $\{z_{uv}^{(0)}(x_v)\}$. In particular, we let $\hat{x}_v^{u(0)} \in \mathbb{R}$, $\forall [u, v] \in \vec{E}$. We then compute the parameters $\{z_{uv}^{(0)}(x_v)\}$ according to (9)-(10). With the initial estimates and messages, the algorithm evolves according to (7)-(10). We use $\hat{x}^{(t)}$ to denote the estimation vector of $\{\hat{x}_u^{v(t)}\}$. The vector $\hat{x}^{(t)}$ is of dimensionality $|\vec{E}|$. If the GLiCD algorithm converges as $t \rightarrow \infty$, we have

$$\hat{x}^{(\infty)} = x_{edge}^*$$

where x_{edge}^* is the corresponding optimal solution constructed from x^* . In this situation, any element in $\{\hat{x}_i^{u(t)}, u \in N(i)\}$ is a good estimate of the optimal solution x_i^* .

We point out when there is no penalty function involved in $L_{ij}^{(t)}(\cdot, \cdot)$ (i.e., no feedback), the GLiCD algorithm degenerates to the LiCD algorithm [9]. Thus, the GLiCD algorithm is more flexible than the LiCD algorithm. The penalty function is imposed to bring the solution $\hat{x}_i^{j(t+1)}$ close to $\{\hat{x}_i^{u(t)}, u \in N(i) \setminus j\}$ for each $[j, i] \in \vec{E}$. Further, large weighting factors $\{\eta_i^{u(t)}, u \in N(i) \setminus j\}$ for the feedback $\{\hat{x}_i^{u(t)}, u \in N(i) \setminus j\}$ enforce high influence on the solution $\hat{x}_i^{j(t+1)}$. Our objective to introduce the penalty function is to make the solutions in the set $\{\hat{x}_i^{u(n)}, u \in N(i)\}$ become more and more close to each other for any $i \in V$ as the algorithm evolves. On the other hand, large weighting factors may negatively affect the convergence speed of the GLiCD algorithm.

3.2. On Convergence of the GLiCD Algorithm

In this subsection, we study under what conditions the GLiCD algorithm converges to the optimal solution x^* , i.e., $\lim_{t \rightarrow \infty} \hat{x}^{(t)} = x_{edge}^*$.

We first introduce the pairwise diagonal dominance condition:

Definition 3.1 (Pairwise Diagonal Dominance): A pairwise separable continuous function $f : \mathbb{R}^{|\vec{E}|} \rightarrow \mathbb{R}$ is pairwise diagonally dominant if

$$K_i \triangleq \inf_{x_i \in \mathbb{R}} \frac{\partial^2}{\partial x_i^2} f_i(x_i) > 0 \quad (11)$$

and

$$\left| \frac{\partial^2}{\partial x_i \partial x_j} f_{ij}(x_i, x_j) \right| \leq \frac{\partial^2}{\partial x_i^2} f_{ij}(x_i, x_j) \quad (12)$$

for all $i \in V$, $(i, j) \in E$ and $x_i, x_j \in \mathbb{R}$.

With the Definition 2.1 and 3.1, we are ready to present the convergence result of GLiCD algorithm in a theorem below.

Theorem 3.1 Consider applying the GLiCD algorithm for a pairwise separable convex program with an objective function that is pairwise diagonal dominant. For any $\hat{x}^{(0)} \in \mathbb{R}^{|\vec{E}|}$ and $t \geq 1$, if the weighting factors $\{\eta_v^{u(t)}\}$ satisfy

$$\eta_v^{u(k)} \geq \frac{\partial^2}{\partial x_v^2} f_{uv}(\hat{x}_u^{v(k)}, \hat{x}_v^{u(k)}) \quad \forall [u, v] \in \vec{E} \text{ and } k = 0, 1, \dots, \quad (13)$$

there exist $0 \leq \lambda_k < 1$, $k = 1, \dots, t$, such that the estimation error at time t is bounded by

$$\|\hat{x}^{(t)} - x_{edge}^*\|_\infty \leq \left(\prod_{k=1}^t \lambda_k \right) \|\hat{x}^{(0)} - x_{edge}^*\|_\infty. \quad (14)$$

Hence, $\lim_{t \rightarrow \infty} \hat{x}^{(t)} = x_{edge}^*$.

The main idea of the proof is to show that the impact of the initial estimate $\hat{x}^{(0)}$ on $\hat{x}^{(t)}$ decreases as the algorithm evolves. As a result, the estimate $\hat{x}^{(t)}$ converges to a fixed point. The parameters λ_k correspond to the infinite norms of some matrices constructed by

the second derivatives of $f(x)$ and the weighting factors. We will present the proof in [11].

4. APPLICATION TO THE AVERAGING PROBLEM IN SENSOR NETWORK

The averaging problem in sensor networks has received intensive attentions in recent years (see [12] for an overview). This is because the average operation can serve as a basis to solve more complicated problems. In this section we consider applying the GLiCD algorithm to the averaging problem in sensor networks.

We first formulate the averaging problem mathematically. Suppose node $i \in V$ obtains a scalar measurement $y_i \in \mathbb{R}$. We use \bar{y} to denote the average of all the measurements, which is computed as

$$\bar{y} = \frac{1}{|V|} \sum_{i \in V} y_i.$$

The research goal is to design a decentralized message-passing algorithm such that in the end the average number \bar{y} is available at every node.

We note that Moallemi and Roy applied the min-sum algorithm to solve the averaging problem [4]. The basic idea of their work is to convert the averaging problem into a convex optimization problem. The min-sum algorithm is then applied to solve the new problem. In particular, in [4], the self and edge potentials are defined as

$$f_{ij}(x_i, x_j) \triangleq \frac{\alpha}{2} (x_i - x_j)^2 \quad i \in V \quad (15)$$

$$f_i(x_i) \triangleq \frac{1}{2} (x_i - y_i)^2 \quad (i, j) \in E, \quad (16)$$

where α is a free parameter. Denote the optimal solution that minimizes the objective function $f(x)$ constructed by (15)-(16) as x^α . It was shown in [4] that $\sum_i x_i^\alpha / |V| = \bar{y}$ and $\lim_{\alpha \uparrow \infty} x_i^\alpha = \bar{y}$, for all $i \in V$. The parameter α determines the accuracy of $\{x_i^\alpha, i \in V\}$ with respect to \bar{y} . Large α improves the accuracy for computing \bar{y} , while at same time may slow down the convergence of the min-sum algorithm.

In this paper, we consider the same objective function with the self and edge potentials defined in (15)-(16), which is obviously pairwise diagonal dominant. In order to guarantee the convergence of the GLiCD algorithm, we have to choose proper weighting factors $\{\eta_u^{v(k)}\}$. Inserting (15) into (13) yields

$$\eta_u^{v(k)} \geq \alpha, \quad \forall [v, u] \in \vec{E}, k = 0, 1, \dots \quad (17)$$

Equation (17) implies that the lower bound is a constant for all the weighting factors, which is due to the simplicity of the edge potentials (15). It is clear from Theorem 3.1 that as long as (17) holds, the GLiCD algorithm would converge to the optimal solution x^α for any initialization $\hat{x}^{(0)}$.

It should be noted that the inequality (17) is only sufficient for the convergence of the GLiCD algorithm. In other words, the lower bound α for the weighting factors may not be tight. We show in the following by experiment that there exist weighting factors smaller than α for which the algorithm still converges.

4.1. Experimental comparison

It is clear that both the min-sum and the GLiCD algorithm converge for the functional construction (15)-(16) in solving the averaging problem. In this subsection we investigate the efficiency of the GLiCD algorithm with the min-sum algorithm as a reference.

We test two graphical models in the experiment. The first one is a planar grid of size 10×10 . Correspondingly, there are 100 nodes in total. Depending on the location of the nodes, each one of them

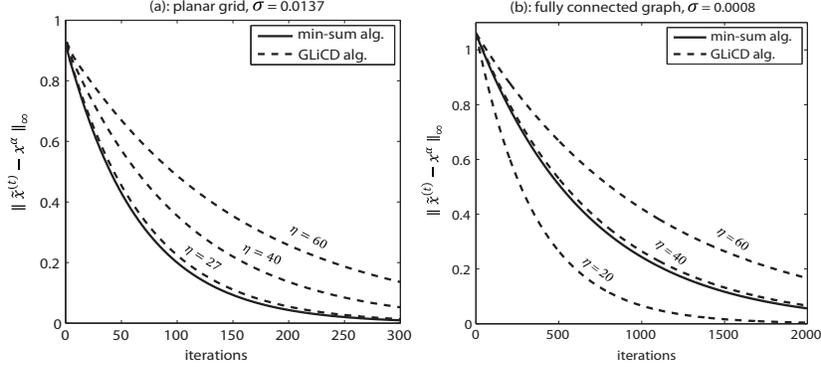


Fig. 1. Performance comparison of the two algorithms for the two graph models, where $\alpha = 40$.

may have two, three or four neighbors. The second one is a fully connected graph with $|V| = 20$ and $N(i) = 19, \forall i \in V$. Thus, the second graph is more dense. In the two graphical models, the scalar measurements $\{y_i, i \in V\}$ were generated from a uniform distribution $\mathcal{U}[0, 2]$.

Next we consider the implementation of the two algorithms. For the GLiCD algorithm, we let the initial estimate $\hat{x}^{(0)} = \mathbf{0}$. Correspondingly, $z_u^{v(0)} = 0, \forall [v, u] \in \vec{E}$. The initialization for the min-sum algorithm is described in detail in [10]. In our experiment, we let $\tilde{m}_{ij}^{(0)}(x_j) = f_{ij}(0, x_j), \forall [i, j] \in \vec{E}$. Since all the edge potentials are in a quadratic form, the messages $\{\tilde{m}_{ij}^{(k)}\}$ are quadratic functions.

For each graphical model, we study the convergence speed of the two algorithms for $\alpha = 40$. The parameter α determines the distance of $x_i^\alpha, i \in V$, to \bar{y} . We use σ to denote the standard derivation of x^α , i.e., $\sigma = \sqrt{\frac{1}{|V|} \sum_{i \in |V|} (x_i^\alpha - \bar{y})^2}$. For the min-sum algorithm, we evaluate the error $\|\hat{x}^{(t)} - x^\alpha\|_\infty$ over time, where $\hat{x}^{(t)}$ is the estimation vector of x^α in $|V|$ -dimensional space. On the other hand, the estimate $\hat{x}^{(t)}$ in the GLiCD algorithm has different dimensionality than that of x^α . To make a fair comparison with the min-sum algorithm, we compute a new estimation vector $\tilde{x}^{(t)}$ using $\hat{x}^{(t)}$. That is for each node $i \in V$, we obtain $\tilde{x}_i^{(t)} = \frac{1}{|N(i)|} \sum_{u \in N(i)} \hat{x}_i^{u(t)}$. With the new estimate, we then compute the error in the infinite norm. The experimental results for the two graphical models are provided in Fig. 1.

It is seen from the figure that the weighting factor η influences the convergence speed of the GLiCD algorithm significantly. Small weighting factor results in high convergence speed. For the two graphical models, we found that there exist weighting factor η smaller than α where the GLiCD algorithm converges. For the planar grid model, the min-sum algorithm converges a bit faster than the GLiCD algorithm, however, at the expense of high computation and storage complexities. For the fully connected model, the two algorithms are comparable w.r.t. the convergence speed. This suggests that if a graph becomes more dense by introducing new edges, the performance of the GLiCD algorithm is increasingly efficient. Due to the quadratic form of the local functions, the min-sum algorithm always doubles the message-transmission bandwidth of the GLiCD algorithm for each iteration. Thus, the GLiCD algorithm may be a better choice for some sensor-network topologies.

5. CONCLUSION

In this paper, we have proposed the GLiCD algorithm for the general convex problems. Differently from the min-sum algorithm, the new algorithm always has linear message form irrespective of the objec-

tive function. Therefore, GLiCD algorithm has lower computational complexity and requires less storage capacity than the min-sum algorithm. We have provided a sufficient condition for the convergence of the GLiCD algorithm. We then successfully applied the GLiCD algorithm to solve the averaging problem in sensor networks.

6. REFERENCES

- [1] T. Richardson and R. Urbanke, "The Capacity of Low-Density Parity Check Codes Under Message-Passing Decoding," *IEEE Trans. Inf. Theory*, vol. 47, pp. 599–618, 2001.
- [2] N. Shental, A. Zomet, T. Hertz, and Y. Weiss, "Learning and Inferring Image Segmentations Using the GBP Typical Cut Algorithm," in *Proc. Int. Conf. Computer Vision*, 2003.
- [3] A. Montanari, B. Prabhakar, and D. Tse, "Belief Propagation Based Multi-User Detection," in *Proc. 43rd Allerton Conf. on Communications, Control and Computing*, 2005.
- [4] C. C. Moallemi and B. Van Roy, "Consensus Propagation," *IEEE Trans. Inf. Theory*, vol. 52, no. 11, pp. 4753–4766, 2006.
- [5] 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.
- [6] P. Rusmevichientong and B. Ban 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.
- [7] C. C. Moallemi and B. Van Roy, "Convergence of Min-Sum Message Passing for Quadratic Optimization," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2413–2423, 2009.
- [8] 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*, Cambridge, MA: MIT Press, 2006, vol. 18.
- [9] G. Zhang and R. Heusdens, "Linear Coordinate-Descent Message-Passing for Quadratic Optimization," *ICASSP*, 2012.
- [10] C. C. Moallemi and B. Van Roy, "Convergence of Min-Sum Message Passing for Convex Optimization," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 2041–2050, 2010.
- [11] G. Zhang and R. Heusdens, "Generalized Linear Coordinate-Descent Message-Passing for Convex Optimization," in preparation for submission to *IEEE Trans. Inform. Theory*.
- [12] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," Available: <http://arxiv.org/abs/1003.5309>, Mar. 2010.