Convergence of Min-Sum Message-Passing for Convex Optimization

Ciamac C. Moallemi, *Member, IEEE*, and Benjamin Van Roy, *Senior Member, IEEE*

***Abstract*—We establish that the min-sum message-passing algorithm and its asynchronous variants converge for a large class of unconstrained convex optimization problems, generalizing existing results for pairwise quadratic optimization problems. The main sufficient condition is that of scaled diagonal dominance. This condition is similar to known sufficient conditions for asynchronous convergence of other decentralized optimization algorithms, such as coordinate descent and gradient descent.**

***Index Terms*—Message-passing algorithms, decentralized optimization, convex optimization.**

I. Introduction

CONSIDER an optimization problem of the form

$$\begin{aligned} &\text{minimize} \quad F(x) \triangleq \sum_{C \in \mathcal{C}} f_C(x_C) \\ &\text{subject to} \quad x \in \mathcal{X}^V. \end{aligned} \tag{1}$$

Here, the vector of decision variables x is indexed by a finite set $V = \{1, \ldots, n\}$. Each decision variable takes values in the set $\mathcal{X}$. The set $\mathcal{C}$ is a collection of subsets of the index set V. This collection describes an additive decomposition of the objective function. We associate with each set $C \in \mathcal{C}$ a component function (or *factor*) $f_C \colon \mathcal{X}^C \to \mathbb{R}$, which takes values as a function of those components[1] x_C of the vector x identified by the elements of C.

The min-sum algorithm is a method for optimization problems of the form (1). It is one of a class of methods known as message-passing algorithms. These algorithms have been the subject of considerable research recently across a number of fields, including communications, artificial intelligence, statistical physics, and signal processing. Interest in message-passing algorithms has been sparked by their success in solving certain classes of NP-hard combinatorial optimization problems, such as the decoding of low-density parity-check codes and turbo codes (e.g., [1]–[3]), or the solution of certain classes of satisfiability problems (e.g., [4] and [5]). Despite their successes, message-passing algorithms remain poorly understood. For example, conditions for convergence and accurate resulting solutions are not well characterized.

In this paper, we consider cases where $\mathcal{X} = \mathbb{R}$, and the optimization problem is continuous. A closely related case that has been examined previously in the literature is where the objective is pairwise separable (i.e., $|C| \leq 2$, for all $C \in \mathcal{C}$) and the component functions $\{f_C(\cdot)\}$ are quadratic and convex. Here, the min-sum algorithm is known to compute the optimal solution when it converges [6]–[8], and sufficient conditions for convergence identify a broad class of problems [9], [10]. Also related is a recent line of work examining the min-sum algorithm in the context of matching, b-matching, maximum weight independent set, etc. (e.g., [11]–[15]). This work has considered the min-sum algorithm for specific classes of convex programs that arise from the linear programming relaxations of certain discrete optimization problems. These problems are constrained, however, thus they do not fall into the formulation at hand.

Our main contribution is the analysis of unconstrained convex optimization in the case where the functions are convex but not necessarily quadratic. We establish that the min-sum algorithm and its asynchronous variants converge for a large class of such problems. Our work are generalizes existing results for pairwise quadratic optimization problems. The main sufficient condition is that of *scaled diagonal dominance*. This condition is similar to known sufficient conditions for asynchronous convergence of other decentralized optimization algorithms, such as coordinate descent and gradient descent.

Analysis of the convex case has been an open challenge and its resolution advances the state of understanding in the growing literature on message-passing algorithms. Further, it builds a bridge between this emerging research area and the better established fields of convex analysis and optimization.

This paper is organized as follows. The next section studies the min-sum algorithm in the context of pairwise separable convex programs, establishing convergence for a broad class of such problems. Section III extends this result to more general separable convex programs, where each factor can be a function of more than two variables. In Section IV, we discuss how our convergence results hold even with a totally asynchronous model of computation. When applied to a continuous optimization problem, messages computed and stored by the min-sum algorithm are functions over continuous domains. Except in very special cases, this is not feasible for digital computers, and in Section V, we discuss implementable approaches to approximating the behavior of the min-sum algorithm. We close by discussing possible extensions and open issues in Section VI.

Manuscript received January 01, 2009; revised December 01, 2009. Current version published March 17, 2010. The work of C. C. Moallemi was supported by a Benchmark Stanford Graduate Fellowship. This work was supported in part by the National Science Foundation by Grant CMMI-0653876.

C. C. Moallemi is with the Graduate School of Business, Columbia University, New York, NY 10027 USA (e-mail: ciamac@gsb.columbia.edu).

B. Van Roy is with the Department of Management Science and Engineering and the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: bvr@stanford.edu).

Communicated by H.-A. Loeliger, Associate Editor for Coding Techniques.

Digital Object Identifier 10.1109/TIT.2010.2040863

[1]Given a vector $x \in \mathcal{X}^V$ and a subset $A \subset V$, we use the notation $x_A = (x_i,\ i \in A) \in \mathcal{X}^A$ for the vector of components of x specified by the set A.

II. Pairwise Separable Convex Programs

Consider first the case of pairwise separable programs. These are programs of the form (1), where $|C| \leq 2$, for all $C \in \mathcal{C}$. In this case, we can define an undirected graph (V, E) based on the objective function. This graph has a vertex set V corresponding to the decision variables, and an edge set E defined by the pairwise factors, i.e.

$$E \triangleq \{C \in \mathcal{C} \,:\, |C| = 2\}.$$

Definition 1 (Pairwise Separable Convex Program): A pairwise separable convex program is an optimization problem of the form

$$\begin{aligned} &\text{minimize} \quad F(x) \triangleq \sum_{i \in V} f_i(x_i) + \sum_{(i,j) \in E} f_{ij}(x_i, x_j) \\ &\text{subject to} \qquad\qquad x \in \mathbb{R}^V \end{aligned} \tag{2}$$

where the factors $\{f_i(\,\cdot\,)\}$ are strictly convex, coercive,[2] and twice continuously differentiable, the factors $\{f_{ij}(\cdot,\cdot)\}$ are convex and twice continuously differentiable, and

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

Under this definition, the objective function $F(\,\cdot\,)$ is strictly convex and coercive. Hence, we can define $x^* \in \mathbb{R}^V$ to be the unique optimal solution. We will see shortly that this definition also guarantees that the update equations of the min-sum algorithm correspond to convex optimization problems.

A. The Min-Sum Algorithm

The min-sum algorithm attempts to minimize the objective function $F(\,\cdot\,)$ by an iterative, message-passing procedure. For each vertex $i \in V$, denote the set of neighbors of i in the graph by

$$N(i) \triangleq \{j \in V \,:\, (i,j) \in E\}.$$

Denote the set of edges with direction distinguished by

$$\vec{E} \triangleq \{(i,j) \in V \times V \,:\, i \in N(j)\}.$$

At time t, each vertex i keeps track of a "message" from each neighbor $u \in N(i)$. This message takes the form of a function $J_{u \to i}^{(t)} \colon \mathbb{R} \to \mathbb{R}$. These incoming messages are combined to compute new outgoing messages for each neighbor. The message $J_{i \to j}^{(t+1)}(\,\cdot\,)$ from vertex i to vertex $j \in N(i)$ evolves according to

$$J_{i \to j}^{(t+1)}(x_j) = \min_{y_i} \; f_i(y_i) + f_{ij}(y_i, x_j) + \sum_{u \in N(i) \setminus j} J_{u \to i}^{(t)}(y_i) + \kappa_{i \to j}^{(t+1)}. \tag{3}$$

[2]A function $h \colon \mathbb{R}^n \to \mathbb{R}$ is coercive if, for every sequence $\{x_k\} \subset \mathbb{R}^n$ with $\|x_k\| \to \infty, h(x_k) \to \infty$. Note that if $h(\,\cdot\,)$ is a coercive and continuous function, then a global minimum of $h(\,\cdot\,)$ must exist [16].

Here, $\kappa_{i \to j}^{(t+1)}$ represents an arbitrary offset term that varies from message to message. Only the relative values of the function $J_{i \to j}^{(t+1)}(\,\cdot\,)$ matter, so the choice of $\kappa_{i \to j}^{(t+1)}$ does not influence relevant information.

At each time $t > 0$, a local objective function $b_i^{(t)}(\,\cdot\,)$ is defined for each variable x_i by

$$b_i^{(t)}(x_i) = f_i(x_i) + \sum_{u \in N(i)} J_{u \to i}^{(t)}(x_i). \tag{4}$$

An estimate $x_i^{(t)}$ can be obtained for the optimal value of the variable x_i by minimizing the local objective function:

$$x_i^{(t)} = \operatorname*{argmin}_{y_i} \; b_i^{(t)}(y_i). \tag{5}$$

The min-sum algorithm requires an initial set of messages $\{J_{i \to j}^{(0)}(\,\cdot\,)\}$ at time $t = 0$. We make the following assumption regarding these messages.

Assumption 1 (Min-Sum Initialization): Assume that the initial messages $\{J_{i \to j}^{(0)}(\,\cdot\,)\}$ are chosen to be twice continuously differentiable and so that, for each message $J_{i \to j}^{(0)}(\,\cdot\,)$, there exists some $z_{i \to j} \in \mathbb{R}$ with

$$\frac{d^2}{dx_j^2} J_{i \to j}^{(0)}(x_j) \geq \frac{\partial^2}{\partial x_j^2} f_{ij}(z_{i \to j}, x_j), \quad \forall\, x_j \in \mathbb{R}. \tag{6}$$

Assumption 1 guarantees that the messages at time $t = 0$ are convex functions. Examining the update (3), it is clear that, by induction, this implies that all future messages are also convex functions. This is because of the fact that, if $g(\cdot,\cdot)$ is a convex function (in both arguments), then $h(x) \triangleq \inf_y g(x,y)$ is also a convex function [17]. Similarly, since the functions $\{f_i(\,\cdot\,)\}$ are strictly convex and coercive, and the functions $\{f_{ij}(\cdot,\cdot)\}$ are convex, it follows that the optimization problem in the update (3) has a strictly convex and coercive objective, so that a global minimum exists and is unique. Finally, each local objective function $b_i^{(t)}(\,\cdot\,)$ must strictly convex and coercive, and hence each estimate $x_i^{(t)}$ is uniquely defined by (5).

Assumption 1 also requires that the initial messages be sufficiently convex, in the sense of (6). As we will shortly demonstrate, this will be an important condition for our convergence results. For the moment, however, note that it is easy to select a set of initial messages satisfying Assumption 1. For example, one might choose

$$J_{i \to j}^{(0)}(x_j) = f_{ij}(0, x_j).$$

B. Convergence

Our goal is to understand conditions under which the min-sum algorithm converges to the optimal solution x^*, i.e.

$$\lim_{t \to \infty} x^{(t)} = x^*.$$

Consider the following diagonal dominance condition:

Definition 2 (Scaled Diagonal Dominance): An objective function $F \colon \mathbb{R}^V \to \mathbb{R}$ is (λ, w)-scaled diagonally dominant if λ

is a scalar with $0 < \lambda < 1$ and $w \in \mathbb{R}^V$ is a vector with $w > 0$, so that for each $i \in V$ and all $x \in \mathbb{R}^V$,

$$\sum_{j \in V \setminus i} w_j \left| \frac{\partial^2}{\partial x_i \partial x_j} F(x) \right| \leq \lambda w_i \frac{\partial^2}{\partial x_i^2} F(x).$$

Our main convergence result, whose proof is provided in Section II-D, is as follows.

Theorem 1: Consider a pairwise separable convex program with an objective function that is (λ, w)-scaled diagonally dominant. Assume that the min-sum algorithm is initialized in accordance with Assumption 1. Define the constant

$$K \triangleq \frac{1}{M} \frac{\max_u w_u}{\min_u w_u}.$$

Then, the iterates of the min-sum algorithm satisfy

$$\|x^{(t)} - x^*\|_\infty \leq K \frac{\lambda^t}{1-\lambda} \times \sum_{(u,v) \in \vec{E}} \left| \frac{d}{dx_v} J^{(0)}_{u \to v}(x^*_v) - \frac{\partial}{\partial x_v} f_{uv}(x^*_u, x^*_v) \right|.$$

Hence

$$\lim_{t \to \infty} x^{(t)} = x^*.$$

We can compare Theorem 1 to existing results on min-sum convergence in the case of where the objective function $F(\cdot)$ is quadratic. Rusmevichientong and Van Roy [7] developed abstract conditions for convergence, but these conditions are difficult to verify in practical instances. Convergence has also been established in special cases arising in certain applications [18], [19].

More closely related to our current work is that of Weiss and Freeman [6]. They established convergence when the factors $\{f_i(\cdot), f_{ij}(\cdot,\cdot)\}$ are quadratic, the single-variable factors $\{f_i(\cdot)\}$ are strictly convex, and the pairwise factors $\{f_{ij}(\cdot,\cdot)\}$ are convex and diagonally dominated, i.e.

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

for all $(i,j) \in E$ and $x_j, x_j \in \mathbb{R}$. It is not difficult to see that this is a special case of (λ, w)-scaled diagonal dominance, with $w = \mathbf{1}$.

The work of Malioutov *et al.* [10] relaxes the assumption of diagonal dominance, replacing it with the more general assumption of "walk-summability." Here, given a positive definite quadratic objective function $F(\cdot)$ with Hessian $\nabla^2 F$, define the matrix $R \in \mathbb{R}^{V \times V}$ by

$$R \triangleq I - D^{-1/2} \nabla^2 F \, D^{-1/2} \tag{7}$$

where $D \triangleq \operatorname{diag}(\nabla^2 F)$ is a matrix with diagonal entries from the Hessian. Denote by $|R|$ the matrix of component-wise absolute values of R. The function $F(\cdot)$ is said to be *walk-summable* if $\rho(|R|) < 1$, i.e., the matrix $|R|$ has spectral radius less than 1. The following theorem shows that this is, in fact, equivalent to scaled diagonal dominance.

Theorem 2: A positive definite quadratic objective function $F(\cdot)$ is walk-summable if and only if it is (λ, w)-scaled diagonally dominant.

Proof: We assume, without loss of generality, assume that the graph (V, E) is connected (otherwise, each connected component can be considered separately). In this case, the matrix $|R|$ is irreducible and nonnegative.

Suppose that $F(\cdot)$ is walk-summable. By the Perron-Frobenius theorem [20, Theorem 8.4.4], there exists a vector $v \in \mathbb{R}^V$ with $v > 0$, and a scalar $\lambda = \rho(|R|) > 0$, so that

$$|R| v = \lambda v.$$

Define $w \triangleq D^{-1/2} v > 0$, and note that $0 < \lambda < 1$. Examining (7), we have precisely that $F(\cdot)$ is (λ, w)-scaled diagonally dominant.

Conversely, assume that $F(\cdot)$ is (λ, w)-scaled diagonally dominant. Define $v \triangleq D^{1/2} w$. Then, we have that $|R| v \leq \lambda v$, with $v > 0$. It follows that $\rho(|R|) \leq \lambda$ [10, Corollary 8.1.29]. Since $\lambda < 1$, $F(\cdot)$ is walk-summable. □

Similarly, in our prior work [9], we establish convergence for quadratic objective functions that decompose into factors so that the single-variable factors are quadratic and strictly convex, and the pairwise factors are quadratic convex. This is equivalent to walk-summability [10, Proposition 13], and, hence, by Theorem 2, also to scaled diagonal dominance. It is worth pointing out, however, that our convergence result in [9] allows for more general initial conditions than Assumption 1.

Finally, as we will see in Section III, Theorem 1 also generalizes beyond pairwise decompositions.

C. The Computation Tree

In order to prove Theorem 1, we first introduce the notion of the *computation tree*. This is a useful device in the analysis of message-passing algorithms, an early instance of which is the work of Wiberg [21]. Given a vertex $r \in V$ and a time t, the computation tree defines an optimization problem that is constructed by "unrolling" all the optimizations involved in the computation of the min-sum estimate $x_r^{(t)}$.

Formally, the computation tree is a graph $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ where each vertex $i \in \mathcal{V}$ is labeled by a corresponding vertex $\tilde{i} \in V$ in the original graph, through a mapping $\sigma \colon \mathcal{V} \to V$. This mapping is required to preserve the edge structure of the graph, so that if $(i,j) \in \mathcal{E}$, then $(\sigma_i, \sigma_j) \in E$. Given a vertex $i \in \mathcal{V}$, we will abuse notation and refer to the corresponding vertex $\sigma_i \in V$ in the original graph simply by i.

Fixing a vertex $r \in V$ and a time t, the computation tree rooted at r and of depth t is defined in an iterative fashion. Initially, the tree consists of a root single vertex corresponding to r. At each subsequent step, the leaves in the computation tree are examined. Given a leaf i with a parent j, a vertex u and an edge (u,i) are added to the computation tree corresponding to each neighbor of i excluding j in the original graph. This process is

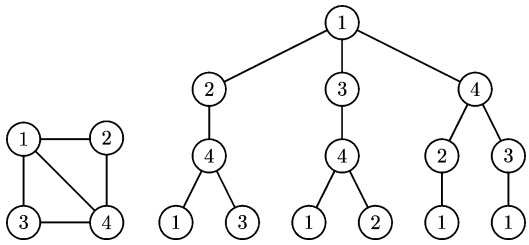

Fig. 1. A graph, on the left, and the corresponding computation tree, on the right, rooted at vertex 1 and of depth $t = 3$. The vertices in the computation tree are labeled according to the corresponding vertices in the original graph.

repeated for t steps. An example of the resulting graph is illustrated in Fig. 1.

Given the graph $\mathcal{T} = (\mathcal{V}, \mathcal{E})$, and the correspondence mapping σ, define a decision variable x_i for each vertex $i \in \mathcal{V}$. Define a pairwise separable objective function $F_{\mathcal{T}}: \mathbb{R}^{\mathcal{V}} \to \mathbb{R}$, by considering factors of the form:

1) For each $i \in \mathcal{V}$, add a single-variable factor $f_i(x_i)$ by setting $f_i(x_i) \triangleq f_{\sigma_i}(x_i)$;
2) For each $(i,j) \in \mathcal{V}$, add a pairwise factor $f_{ij}(x_i, x_j)$ by setting $f_{ij}(x_i, x_j) \triangleq f_{\sigma_i \sigma_j}(x_i, x_j)$;
3) For each $i \in \mathcal{V}$ that is a leaf vertex with parent j, add a single-variable factor $J^{(0)}_{u \to \sigma_i}(x_i)$, for each neighbor $u \in N(\sigma_i) \setminus \sigma_j$ of i in the original graph, excluding j.

Now, let $\tilde{x}$ be the optimal solution to the minimization of the computation tree objective $F_{\mathcal{T}}(\cdot)$. By inductively examining the operation of the min-sum algorithm, it is easy to establish that the component $\tilde{x}_r$ of this solution at the root of the tree is precisely the min-sum estimate $x_r^{(t)}$.

The following lemma establishes that the computation tree inherits the scaled diagonal dominance property from the original objective function.

Lemma 1: Consider a pairwise separable convex program with an objective function that is (λ, w)-scaled diagonally dominant. Assume that the min-sum algorithm is initialized in accordance with Assumption 1, and let $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ be a computation tree associated with this program. Then, the computation tree objective function $F_{\mathcal{T}}(\cdot)$ is also (λ, w)-scaled diagonally dominant.

Proof: Given a vertex $i \in \mathcal{V}$, let $N^{\mathcal{V}}(i)$ be the neighborhood in the computation tree, and let $N(i)$ be the neighborhood of the corresponding vertex in the original graph. If $i \in \mathcal{V}$ is an interior vertex of the computation tree, then

$$\begin{aligned}
&\sum_{u \in \mathcal{V} \setminus i} w_u \left| \frac{\partial^2}{\partial x_i \partial x_u} F_{\mathcal{T}}(x) \right| \\
&\quad = \sum_{u \in N^{\mathcal{V}}(i)} w_u \left| \frac{\partial^2}{\partial x_i \partial x_u} f_{iu}(x_i, x_u) \right| \\
&\quad \le \lambda w_i \left(\frac{\partial^2}{\partial x_i^2} f_i(x_i) + \sum_{u \in N^{\mathcal{V}}(i)} \frac{\partial^2}{\partial x_i^2} f_{iu}(x_i, x_u) \right) \\
&\quad = \lambda w_i \frac{\partial^2}{\partial x_i^2} F_{\mathcal{T}}(x)
\end{aligned}$$

where the inequality follows from the scaled diagonal dominance of the original objective function $F(\cdot)$.

Similarly, if i is a leaf vertex with parent j,

$$\begin{aligned}
&\sum_{u \in \mathcal{V} \setminus i} w_u \left| \frac{\partial^2}{\partial x_i \partial x_u} F_{\mathcal{T}}(x) \right| \\
&\quad = w_j \left| \frac{\partial^2}{\partial x_i \partial x_j} f_{ij}(x_i, x_j) \right| \\
&\quad \le w_j \left| \frac{\partial^2}{\partial x_i \partial x_j} f_{ij}(x_i, x_j) \right| \\
&\qquad + \sum_{u \in N(i) \setminus j} w_u \left| \frac{\partial^2}{\partial x_i \partial x_u} f_{iu}(x_i, z_{u \to i}) \right| \\
&\quad \le \lambda w_i \left(\frac{\partial^2}{\partial x_i^2} f_i(x_i) + \frac{\partial^2}{\partial x_i^2} f_{ij}(x_i, x_j) \right. \\
&\qquad \left. + \sum_{u \in N(i) \setminus j} \frac{\partial^2}{\partial x_i^2} f_{iu}(x_i, z_{u \to i}) \right) \\
&\quad \le \lambda w_i \left(\frac{\partial^2}{\partial x_i^2} f_i(x_i) + \frac{\partial^2}{\partial x_i^2} f_{ij}(x_i, x_j) \right. \\
&\qquad \left. + \sum_{u \in N(i) \setminus j} \frac{\partial^2}{\partial x_i^2} J^{(0)}_{u \to i}(x_i) \right) \\
&\quad = \lambda w_i \frac{\partial^2}{\partial x_i^2} F_{\mathcal{T}}(x).
\end{aligned}$$

Here, the second inequality follows from the scaled diagonal dominance of the original objective function $F(\cdot)$, and the third inequality follows from Assumption 1. □

D. Proof of Theorem 1

In order to prove Theorem 1, we will study the evolution of the min-sum algorithm under a set of linear perturbations. Consider an arbitrary vector $p \in \mathbb{R}^{\vec{E}}$ with one component $p_{i \to j}$ for each $i \in V$ and $j \in N(i)$. Given an arbitrary vector p, define $\{J^{(t)}_{i \to j}(\cdot, p)\}$ to be the set of messages that evolve according to

$$\begin{aligned}
J^{(0)}_{i \to j}(x_j, p) &= J^{(0)}_{i \to j}(x_j) + p_{i \to j} x_j, \\
J^{(t+1)}_{i \to j}(x_j, p) &= \min_{y_i} \; f_i(y_i) + f_{ij}(y_i, x_j) \\
&\qquad + \sum_{u \in N(i) \setminus j} J^{(t)}_{u \to i}(y_i, p) + \kappa^{(t+1)}_{i \to j}. \qquad (8)
\end{aligned}$$

Similarly, define $\{b_i^{(t)}(\cdot, p)\}$ and $\{x_i^{(t)}(p)\}$ to be the resulting local objective functions and optimal value estimates under this perturbation

$$\begin{aligned}
b_i^{(t)}(x_i, p) &= f_i(x_i) + \sum_{u \in N(i)} J^{(t)}_{u \to i}(x_i, p), \\
x_i^{(t)}(p) &= \operatorname*{argmin}_{y_i} \; b_i^{(t)}(y_i, p).
\end{aligned}$$

The following simple lemma gives a particular choice of p for which the min-sun algorithm yields the optimal solution at every time.

Lemma 2: Define the vector $p^* \in \mathbb{R}^{\vec{E}}$ by setting, for each $i \in V$ and $j \in N(i)$

$$p^*_{i\to j} \triangleq \frac{\partial}{\partial x_j} f_{ij}(x^*_i, x^*_j) - \frac{d}{dx_j} J^{(0)}_{i\to j}(x^*_j).$$

Then, at every time $t \geq 0$,

$$\frac{\partial}{\partial x_j} J^{(t)}_{i\to j}(x^*_j, p^*) = \frac{\partial}{\partial x_j} f_{ij}(x^*_i, x^*_j) \tag{9}$$

and $x^{(t)}_j(p^*) = x^*_j$.

Proof: Note that the first-order optimality conditions for $F(\,\cdot\,)$ at x^* imply that, for each $j \in V$

$$\frac{d}{dx_j} f_j(x^*_j) + \sum_{i\in N(j)} \frac{\partial}{\partial x_j} f_{ij}(x^*_i, x^*_j) = 0.$$

If (9) holds at time t, this is exactly the first-order optimality condition for the minimization of $b^{(t)}_j(\cdot, p^*)$, thus $x^{(t)}_j(p^*) = x^*_j$.

Clearly (9) holds at time $t = 0$. Assume it holds at time $t \geq 0$. Then, when $x_j = x^*_j$, the minimizing value of y_i in (8) is x^*_i. Hence, (9) holds at time $t + 1$. □

Next, we will bound the sensitivity of the estimate $x^{(t)}_i(p)$ to the choice of p. The main technique employed here is analysis of the computation tree described in Section II-C. In particular, the perturbation p impacts the computation tree only through the leaf vertices at depth t. The scaled diagonal dominance property of the computation tree, provided by Lemma 1, can then be used to guarantee *correlation decay*—we guarantee that the impact of the leaves to the root is diminishing in t.

Lemma 3: We have, for all $p \in \mathbb{R}^{\vec{E}}$, $r \in V$, $(u, v) \in \vec{E}$, and $t \geq 0$

$$\left| \frac{\partial}{\partial p_{u\to v}} x^{(t)}_r(p) \right| \leq K \frac{\lambda^t}{1-\lambda}.$$

Proof: Fix $r \in V$, and let $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ be the computation tree rooted at r after t time steps. Let $F_{\mathcal{T}}(x, p)$ be the objective value of this computation tree, and let

$$\tilde{x}(p) \triangleq \operatorname*{argmin}_x \; F_{\mathcal{T}}(x, p)$$

so that

$$\tilde{x}_r(p) = x^{(t)}_r(p).$$

By the first-order optimality conditions, for any $j \in \mathcal{V}$

$$\frac{\partial}{\partial x_j} F_{\mathcal{T}}(\tilde{x}(p), p) = 0.$$

If j is an interior vertex of $\mathcal{T}$, this becomes

$$\frac{d}{dx_j} f_j(\tilde{x}_j(p)) + \sum_{i\in N(j)} \frac{\partial}{\partial x_j} f_{ij}(\tilde{x}_i(p), \tilde{x}_j(p)) = 0. \tag{10}$$

If j is a leaf with parent u, we have

$$\frac{d}{dx_j} f_j(\tilde{x}_j(p)) + \frac{\partial}{\partial x_j} f_{uj}(\tilde{x}_u(p), \tilde{x}_j(p)) + \sum_{i\in N(j)\setminus u} \left(\frac{\partial}{\partial x_j} J^{(0)}_{i\to j}(\tilde{x}_j(p)) + p_{i\to j} \right) = 0. \tag{11}$$

Now, fix some directed edge (a, b), and differentiate (10)–(11) with respect to $p_{a\to b}$. We have, for an interior vertex j

$$\begin{aligned} 0 = {} & \frac{d^2}{dx_j^2} f_j(\tilde{x}_j(p)) \frac{\partial}{\partial p_{a\to b}} \tilde{x}_j(p) \\ & + \sum_{i\in N(j)} \frac{\partial^2}{\partial x_j^2} f_{ij}(\tilde{x}_i(p), \tilde{x}_j(p)) \frac{\partial}{\partial p_{a\to b}} \tilde{x}_j(p) \\ & + \sum_{i\in N(j)} \frac{\partial^2}{\partial x_i \partial x_j} f_{ij}(\tilde{x}_i(p), \tilde{x}_j(p)) \frac{\partial}{\partial p_{a\to b}} \tilde{x}_i(p) \end{aligned}$$

and for a leaf vertex j with parent u,

$$\begin{aligned} 0 = {} & \frac{d^2}{dx_j^2} f_j(\tilde{x}_j(p)) \frac{\partial}{\partial p_{a\to b}} \tilde{x}_j(p) \\ & + \frac{\partial^2}{\partial x_j^2} f_{uj}(\tilde{x}_u(p), \tilde{x}_j(p)) \frac{\partial}{\partial p_{a\to b}} \tilde{x}_j(p) \\ & + \frac{\partial^2}{\partial x_u \partial x_j} f_{uj}(\tilde{x}_u(p), \tilde{x}_j(p)) \frac{\partial}{\partial p_{a\to b}} \tilde{x}_u(p) \\ & + \sum_{i\in N(j)\setminus u} \left(\frac{\partial^2}{\partial x_j^2} J^{(0)}_{i\to j}(\tilde{x}_j(p)) \frac{\partial}{\partial p_{a\to b}} \tilde{x}_j(p) \right. \\ & \left. \quad + \mathbb{I}_{\{(a,b)=(i,j)\}} \right). \end{aligned}$$

We can write this system of equations in matrix form, as

$$\Gamma v^{a\to b} + h^{a\to b} = 0. \tag{12}$$

Here, $v^{a\to b} \in \mathbb{R}^{\mathcal{V}}$ is a vector with components

$$v^{a\to b}_j \triangleq \frac{\partial}{\partial p_{a\to b}} \tilde{x}_j(p).$$

The vector $h^{a\to b} \in \mathbb{R}^{\mathcal{V}}$ has components

$$h^{a\to b}_j \triangleq \mathbb{I}_{\{j \text{ is a leaf vertex of type } a \text{ with a parent of type } b\}}.$$

The symmetric matrix $\Gamma \in \mathbb{R}^{\mathcal{V}\times\mathcal{V}}$ has components as follows:

1) If j is an interior vertex

$$\Gamma_{jj} \triangleq \frac{d^2}{dx_j^2} f_j(\tilde{x}_j(p)) + \sum_{i\in N(j)} \frac{\partial^2}{\partial x_j^2} f_{ij}(\tilde{x}_i(p), \tilde{x}_j(p)).$$

2) If j is an interior vertex and $i \in N(j)$,

$$\Gamma_{ij} \triangleq \frac{\partial^2}{\partial x_i \partial x_j} f_{ij}(\tilde{x}_i(p), \tilde{x}_j(p)).$$

3) If j is a leaf vertex with parent u,

$$\Gamma_{jj} \triangleq \frac{d^2}{dx_j^2} f_j(\tilde{x}_j(p)) + \frac{\partial^2}{\partial x_j^2} f_{uj}(\tilde{x}_u(p), \tilde{x}_j(p)) + \sum_{i \in N(j) \setminus u} \frac{\partial^2}{\partial x_j^2} J_{i \to j}^{(0)}(\tilde{x}_j(p)),$$
$$\Gamma_{uj} \triangleq \frac{\partial^2}{\partial x_u \partial x_j} f_{uj}(\tilde{x}_u(p), \tilde{x}_j(p)).$$

4) All other entries of Γ are zero.

Note that $\Gamma = \nabla_x^2 F_{\mathcal{T}}(\tilde{x}(p), p)$. Then, Lemma 1 implies that

$$\sum_{i \in \mathcal{V} \setminus j} w_i |\Gamma_{ij}| \leq \lambda w_j \Gamma_{jj}. \tag{13}$$

Define, for vectors $x \in \mathbb{R}^{\mathcal{V}}$, the weighted sup-norm

$$\|x\|_\infty^w \triangleq \max_{j \in \mathcal{V}} \frac{|x_j|}{w_j}.$$

For a linear operator $A: \mathbb{R}^{\mathcal{V}} \to \mathbb{R}^{\mathcal{V}}$, the corresponding induced operator norm is given by

$$\|A\|_\infty^w \triangleq \max_{j \in \mathcal{V}} \frac{1}{w_j} \sum_{i \in \mathcal{V}} w_i |A_{ji}|.$$

Define the matrices

$$D \triangleq \operatorname{diag}(\Gamma),$$
$$R \triangleq I - D^{-1}\Gamma.$$

Then, (13) implies that

$$\|R\|_\infty^w \leq \lambda < 1.$$

Hence, the matrix $I - R = D^{-1}\Gamma$ is invertible, and

$$(D^{-1}\Gamma)^{-1} = (I - R)^{-1} = \sum_{s=0}^{\infty} R^s.$$

Examining the linear equation (12), we have

$$v^{a \to b} = -\Gamma^{-1} h^{a \to b} = -(I - R)^{-1} D^{-1} h^{a \to b} = -\sum_{s=0}^{\infty} R^s D^{-1} h^{a \to b}.$$

We are interested in bounding the value of the component $v_r^{a \to b}$ (recall that $v_r^{a \to b} = \partial x_r^{(t)}(p) / \partial p_{a \to b}$). Hence, we have

$$v_r^{a \to b} = -\sum_{s=0}^{\infty} \left[R^s D^{-1} h^{a \to b} \right]_r.$$

Since $h^{a \to b}$ is zero on interior vertices, and any leaf vertex is distance t from the root r, we have

$$[R^s D^{-1} h^{a \to b}]_r = 0, \quad \forall\, s < t.$$

Thus,

$$v_r^{a \to b} = -\sum_{s=t}^{\infty} [R^s D^{-1} h^{a \to b}]_r.$$

Then,

$$\begin{aligned} \frac{|v_r^{a \to b}|}{w_r} &\leq \left\| \sum_{s=t}^{\infty} R^s D^{-1} h^{a \to b} \right\|_\infty^w \\ &\leq \sum_{s=t}^{\infty} \|R^s\|_\infty^w \|D^{-1} h^{a \to b}\|_\infty^w \\ &\leq \frac{\lambda^t}{1 - \lambda} \|D^{-1} h^{a \to b}\|_\infty^w \\ &\leq \frac{\lambda^t}{1 - \lambda} \max_{i \in V} \sup_{x \in \mathbb{R}^V} \left(w_i \frac{\partial^2}{\partial x_i^2} F(x) \right)^{-1} \\ &\leq M \frac{\lambda^t}{1 - \lambda} \max_{i \in \mathcal{V}} \frac{1}{w_i}. \end{aligned}$$

□

The following lemma combines the results from Lemmas 2 and 3. Theorem 1 follows by taking $p = 0$.

Lemma 4: Given an arbitrary vector $p \in \mathbb{R}^{\vec{E}}$

$$\|x^{(t)}(p) - x^*\|_\infty \leq K \frac{\lambda^t}{1 - \lambda} \sum_{(u,v) \in \vec{E}} |p_{u \to v} - p_{u \to v}^*|.$$

Proof: For any $j \in V$, define

$$g_j^{(t)}(\theta) = x_j^{(t)}(\theta p + (1 - \theta) p^*).$$

We have, from Lemma 2

$$x_j^{(t)}(p) - x_j^* = x_j^{(t)}(p) - x_j^{(t)}(p^*) = g_j^{(t)}(1) - g_j^{(t)}(0).$$

By the mean value theorem and Lemma 3

$$\begin{aligned} &|x_j^{(t)}(p) - x_j^*| \\ &\quad \leq \sup_{\theta \in [0,1]} \left| \frac{d}{d\theta} g_j^{(t)}(\theta) \right| \\ &\quad \leq \sup_{\theta \in [0,1]} \sum_{(u,v) \in \vec{E}} \left| \frac{\partial}{\partial p_{u \to v}} x_j^{(t)}(\theta p + (1 - \theta) p^*) \right| \\ &\qquad \times |p_{u \to v} - p_{u \to v}^*| \\ &\quad \leq K \frac{\lambda^t}{1 - \lambda} \sum_{(u,v) \in \vec{E}} |p_{u \to v} - p_{u \to v}^*|. \end{aligned}$$

□

III. General Separable Convex Programs

In this section we will consider convergence of the min-sum algorithm for more general separable convex programs. In particular, consider a vector of real-valued decision variables $x \in$

$\mathbb{R}^V$, indexed by a finite set V, and a hypergraph $(V, \mathcal{C})$, where the set $\mathcal{C}$ is a collection of subsets (or, "hyperedges") of the vertex set V.

Definition 3 (General Separable Convex Program): A general separable convex program is an optimization problem of the form

$$\begin{aligned} \text{minimize} \quad & F(x) \triangleq \sum_{i \in V} f_i(x_i) + \sum_{C \in \mathcal{C}} f_C(x_C) \\ \text{subject to} \quad & x \in \mathbb{R}^V, \end{aligned} \tag{14}$$

where the factors $\{f_i(\cdot)\}$ are strictly convex, coercive, and twice continuously differentiable, the factors $\{f_C(\cdot)\}$ are convex and twice continuously differentiable, and

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

As in the pairwise case, under this definition, the objective function $F(\cdot)$ is strictly convex and coercive. Hence, we can define $x^* \in \mathbb{R}^V$ to be the unique optimal solution.

In this setting, the min-sum algorithm operates by passing messages between vertices and hyperedges. In particular, denote the set of neighbor hyperedges to a vertex $i \in V$ by

$$N_f(i) \triangleq \{C \in \mathcal{C} \,:\, i \in C\}$$

The min-sum update equations take the form

$$\begin{aligned} J_{i \to C}^{(t+1)}(x_i) &= f_i(x_i) + \sum_{C' \in N_f(i) \setminus C} J_{C' \to i}^{(t)}(x_i) + \kappa_{i \to C}^{(t+1)}, \\ J_{C \to i}^{(t+1)}(x_i) &= \min_{y_{C \setminus i}} f_C(x_i, y_{C \setminus i}) + \sum_{i' \in C \setminus i} J_{i' \to C}^{(t+1)}(y_{i'}) \\ &\quad + \kappa_{C \to i}^{(t+1)}. \end{aligned} \tag{15}$$

Local objective functions and estimates of the optimal solution are defined by

$$\begin{aligned} b_i^{(t)}(x_i) &= f_i(x_i) + \sum_{C \in N_f(i)} J_{C \to i}^{(t)}(x_i), \\ x_i^{(t)} &= \operatorname*{argmin}_{y_i} b_i^{(t)}(y_i). \end{aligned}$$

We will make the following assumption on the initial messages.

Assumption 2 (Min-Sum Initialization): Assume that the initial messages $\{J_{C \to j}^{(0)}(\cdot)\}$ are chosen to be twice continuously differentiable and so that, for each message $J_{C \to j}^{(0)}(\cdot)$, there exists some $z_{C \to j} \in \mathbb{R}^{C \setminus i}$ with

$$\frac{d^2}{dx_j^2} J_{C \to j}^{(0)}(x_j) \geq \frac{\partial^2}{\partial x_j^2} f_C(x_j, z_{C \to j}), \quad \forall\, x_j \in \mathbb{R}.$$

As in the pairwise case, this assumption guarantees that the messages $\{J_{C \to j}^{(t)}(\cdot)\}$ are strictly convex and coercive for $t \geq 1$. Furthermore, it follows that the optimization problem in the update (15) has a strictly convex and coercive objective, so that a global minimum exists and is unique. Finally, each local objective function $b_i^{(t)}(\cdot)$ must strictly convex and coercive, and hence each estimate $x_i^{(t)}$ is uniquely defined.

Then, we have the following analog of Theorem 1.

Theorem 3: Consider a general separable convex program. Assume that either:

a) The objective function $F(\cdot)$ is (w, λ)-scaled diagonally dominant, and each pair of vertices $i, j \in V$ participate in at most one common factor. That is

$$\left|\{C \in \mathcal{C} \,:\, (i,j) \subset C\}\right| \leq 1, \quad \forall\, i, j \in V.$$

b) The factors $\{f_C(\cdot)\}$ are individually (w, λ)-scaled diagonally dominant, in the sense that exists a scalar $\lambda \in (0,1)$ and a vector $w \in \mathbb{R}^V$, with $w > 0$, so that for all $C \in \mathcal{C}$, $i \in C$, and $x_C \in \mathbb{R}^C$

$$\sum_{j \in C \setminus i} w_j \left| \frac{\partial^2}{\partial x_i \partial x_j} f_C(x_C) \right| \leq \lambda w_i \frac{\partial^2}{\partial x_i^2} f_C(x_C).$$

Assume that the min-sum algorithm is initialized in accordance with Assumption 2. Define the constant

$$K \triangleq \frac{1}{M} \frac{\max_u w_u}{\min_u w_u}.$$

Then, the iterates of the min-sum algorithm satisfy

$$\begin{aligned} &\|x^{(t)} - x^*\|_\infty \\ &\quad \leq K \frac{\lambda^t}{1 - \lambda} \sum_{C \in \mathcal{C}} \sum_{v \in C} \left| \frac{d}{dx_v} J_{C \to v}^{(0)}(x_v^*) - \frac{\partial}{\partial x_v} f_C(x_C^*) \right|. \end{aligned}$$

Hence

$$\lim_{t \to \infty} x^{(t)} = x^*.$$

Proof: This result can be proved using the same method as Theorem 1. The main modification required is the development of a suitable analog of Lemma 1. In the general case, scaled diagonal dominance of the computation tree *does not* follow from scaled diagonal dominance of the objective function $F(\cdot)$. This is because a pair of variables can participate in multiple common factors in the objective function, while, in the unrolled computation tree, any pair of variables participates in at most a single common factor.

The hypotheses (a) and (b) are sufficient conditions to guarantee scaled diagonal dominance of the computation tree—in the pairwise case of Lemma 1, (a) implicitly holds. To see this, in what follows, consider a computation tree $\mathcal{T} = (\mathcal{V}, \tilde{\mathcal{C}})$. If $i \in \mathcal{V}$ is a vertex in the computation tree with a neighboring vertex $j \in N^{\mathcal{V}}(i)$, define $\tilde{C}(i,j)$ to be the unique common factor.

Suppose that (a) holds. Then, if $i \in \mathcal{V}$ is an interior vertex of the computation tree

$$\begin{aligned}
&\sum_{u \in \mathcal{V} \setminus i} w_u \left| \frac{\partial^2}{\partial x_i \partial x_u} F_{\mathcal{T}}(x) \right| \\
&\quad = \sum_{u \in N^{\mathcal{V}}(i)} w_u \left| \frac{\partial^2}{\partial x_i \partial x_u} f_{\tilde{C}(i,u)} \left(x_{\tilde{C}(i,u)} \right) \right| \\
&\quad \leq \lambda w_i \left(\frac{\partial^2}{\partial x_i^2} f_i(x_i) + \sum_{C \in \tilde{\mathcal{C}}:\, i \in C} \frac{\partial^2}{\partial x_i^2} f_C(x_C) \right) \\
&\quad = \lambda w_i \frac{\partial^2}{\partial x_i^2} F_{\mathcal{T}}(x)
\end{aligned}$$

where the inequality follows from (a). By similarly considering the case of a leaf vertex, it is clear that $F_{\mathcal{T}}(\,\cdot\,)$ is (λ, w)-scaled diagonally dominant.

Alternatively, suppose that (b) holds. Then, if $i \in \mathcal{V}$ is an interior vertex of the computation tree

$$\begin{aligned}
&\sum_{u \in \mathcal{V} \setminus i} w_u \left| \frac{\partial^2}{\partial x_i \partial x_u} F_{\mathcal{T}}(x) \right| \\
&\quad = \sum_{u \in N^{\mathcal{V}}(i)} w_u \left| \frac{\partial^2}{\partial x_i \partial x_u} f_{\tilde{C}(i,u)} \left(x_{\tilde{C}(i,u)} \right) \right| \\
&\quad = \sum_{C \in \tilde{\mathcal{C}}:\, i \in C} \sum_{u \in C \setminus i} w_u \left| \frac{\partial^2}{\partial x_i \partial x_u} f_C(x_C) \right| \\
&\quad \leq \sum_{C \in \tilde{\mathcal{C}}:\, i \in C} \lambda w_i \frac{\partial^2}{\partial x_i^2} f_C(x_C) \\
&\quad < \lambda w_i \left(\frac{\partial^2}{\partial x_i^2} f_i(x_i) + \sum_{C \in \tilde{\mathcal{C}}:\, i \in C} \frac{\partial^2}{\partial x_i^2} f_C(x_C) \right) \\
&\quad = \lambda w_i \frac{\partial^2}{\partial x_i^2} F_{\mathcal{T}}(x),
\end{aligned}$$

where the first inequality follows from (b), and the second inequality follows since $f_i(\,\cdot\,)$ is strictly convex. By analogous consideration of the case of a leaf vertex, it is clear that $F_{\mathcal{T}}(\,\cdot\,)$ is (λ, w)-scaled diagonally dominant in this case also.

Thus, either of the hypotheses (a) or (b) imply scaled diagonal dominance of the computation tree. The balance of the proof proceeds as in Section II-D. □

IV. Asynchronous Convergence

The convergence results of Theorems 1 and 3 assumed a synchronous model of computation. That is, each message is updated at every time step in parallel. The min-sum update (3) and (15) are naturally decentralized, however. If we consider the application of the min-sum algorithm in distributed contexts, it is necessary to consider convergence under an *asynchronous* model of computation. In this section, we will establish that Theorems 1 and 3 extend to an asynchronous setting.

Without loss of generality, consider the pairwise case. Assume that there is a processor associated with each vertex i in the graph, and that this processor is responsible for computing the message $J_{i \to j}(\,\cdot\,)$, for each neighbor j of vertex i. Each processor occasionally communicates its messages to neighboring processors, and occasionally computes new messages based on the most recent messages it has received. Define the T^i to be the set of times at which new messages are computed. Define $0 \leq \tau_{j \to i}(t) \leq t$ to be the last time the processor at vertex j communicated to the processor at vertex i. Then, the messages evolve according to

$$\begin{aligned}
J_{i \to j}^{(t+1)}(x_j) = \min_{y_i} \; & f_i(y_i) + f_{ij}(y_i, x_j) \\
& + \sum_{u \in N(i) \setminus j} J_{u \to i}^{(\tau_{u \to i}(t))}(y_i) + \kappa_{i \to j}^{(t+1)}
\end{aligned}$$

if $t \in T^i$, and

$$J_{i \to j}^{(t+1)}(x_j) = J_{i \to j}^{(t)}(x_j)$$

otherwise.

We will make the following assumption [22].

Assumption 3 (Total Asynchronism): Suppose that each set T^i is infinite, and that if $\{t_k\}$ is a sequence in T^i tending to infinity, then

$$\lim_{k \to \infty} \tau_{i \to j}(t_k) = \infty$$

for each neighbor $j \in N(i)$.

Total asynchronism is a very mild assumption. It guarantees that each component is updated infinitely often, and that processors eventually communicate with neighboring processors. It allows for arbitrary delays in communication, and even the out-of-order arrival of messages between processors.

Theorem 1 can be extended to the totally asynchronous setting. To see this, note that we can repeat the construction of the computation tree in Section II-C. As in the synchronous case, the initial messages only impact the leaves of computation tree. The total asynchronism assumption guarantees that these leaves are, eventually, arbitrarily far away from the root of the computation tree. The arguments in Lemma 3 then imply that the optimal value at the root of the computation tree is insensitive to the choice of initial messages. Convergence follows, as in Section II-D.

The scaled diagonal dominance requirement of our convergence result is similar to conditions required for the totally asynchronous convergence of other optimization algorithms. Consider, for example, a decentralized coordinate descent algorithm. Here, the processor associated with vertex i maintains an estimate $x_i^{(t)}$ of the ith component of the optimal solution at time t. These estimates are updated according to

$$x_i^{(t+1)} = \operatorname*{argmin}_{y_i} \; f_i(y_i) + \sum_{u \in N(i)} f_{oi}\left(x_u^{(\tau_{u \to i}(t))}, y_i \right)$$

if $t \in T^i$, and $x_i^{(t+1)} = x_i^{(t)}$, otherwise. Similarly, consider a decentralized gradient method, where

$$x_i^{(t+1)} = x_i^{(t)} - \alpha \frac{\partial}{\partial x_i} \left(f_i\left(x_i^{(t)}\right) + \sum_{u \in N(i)} f_{ui}\left(x_u^{(\tau_{u \to i}(t))}, x_i^{(t)}\right) \right)$$

if $t \in T^i$, and $x_i^{(t+1)} = x_i^{(t)}$, otherwise, for some small positive step size α. These methods are not guaranteed to converge for arbitrary pairwise separable convex optimization problems. One sufficient condition is that the updates of each algorithm are contraction mappings under a weighted maximum norm, and this is can be established by assuming some sort of scaled diagonal dominance [22]. This is similar in spirit to the correlation decay argument provided in Lemma 3.

V. Implementation

The convergence theory we have presented elucidates properties of the min-sum algorithm and builds a bridge to the more established areas of convex analysis and optimization. However, except in very special cases, the algorithm as we have formulated it can not be implemented on a digital computer because the messages that are computed and stored are functions over continuous domains. In this section, we present two variations that can be implemented to approximate behavior of the min-sum algorithm. Note that our convergence results do not apply to the approximate algorithms described here. The study of the convergence of such approximate variations is an interesting open question. For simplicity, in what follows, we restrict attention to the case of the synchronous min-sum algorithm for pairwise separable convex programs.

First, consider an approach which approximates messages using quadratic functions. This can be viewed as a hybrid between the min-sum algorithm and Newton's method. It is easy to show that, if the single-variable factors $\{f_i(\cdot)\}$ are positive definite quadratics and the pairwise factors $\{f_{ij}(\cdot,\cdot)\}$ are positive semidefinite quadratics, then min-sum updates map quadratic messages to quadratic messages. The algorithm we propose here maintains a running estimate $\tilde{x}^{(t)}$ of the optimal solution, and at each time approximates each factor by a second-order Taylor expansion. In particular, let $\tilde{f}_i^{(t)}(\cdot)$ be the second-order Taylor expansion of $f_i(\cdot)$ around $\tilde{x}_i^{(t)}$ and let $\tilde{f}_{ij}^{(t)}(\cdot,\cdot)$ be the second-order Taylor expansion of $f_{ij}(\cdot,\cdot)$ around $\left(\tilde{x}_i^{(t)}, \tilde{x}_j^{(t)}\right)$. Quadratic messages are updated according to

$$J_{i \to j}^{(t+1)}(x_j) = \min_{y_i} \tilde{f}_i^{(t)}(y_i) + \tilde{f}_{ij}^{(t)}(y_i, x_j) + \sum_{u \in N(i) \setminus j} J_{u \to i}^{(t)}(y_i) + \kappa_{i \to j}^{(t+1)} \quad (16)$$

where running estimates of the optimal solution are generated according to

$$\tilde{x}_i^{(t+1)} = \operatorname*{argmin}_{y_i} \left(\tilde{f}_i^{(t+1)}(y_i) + \sum_{u \in N(i)} J_{u \to i}^{(t+1)}(y_i) \right). \quad (17)$$

Note that the message update (16) takes the form of a Ricatti equation for a scalar system, which can be carried out efficiently. Further, each optimization problem (17) is a scalar unconstrained convex quadratic program.

A second approach makes use of a piecewise-linear approximation to each message. Let us assume knowledge that the optimal solution x^* is in a closed bounded set $[-B, B]^n$. Let $\mathcal{S} = \{\hat{x}_1, \ldots, \hat{x}_m\} \subset [-B, B]$, with $-B = \hat{x}_1 < \cdots < \hat{x}_m = B$, be a set of points where the linear pieces begin and end. Our approach applies the min-sum update equation to compute values at these points. Then, an approximation to the min-sum message is constructed via linear interpolation between consecutive points or extrapolation beyond the end points. In particular, the algorithm takes the form

$$J_{i \to j}^{(t+1)}(x_j) = \min_{y_i \in [-B, B]} f_i(y_i) + f_{ij}(y_i, x_j) + \sum_{u \in N(i) \setminus j} J_{u \to i}^{(t)}(y_i) + \kappa_{i \to j}^{(t+1)}$$

for $x_j \in \mathcal{S}$, where

$$J_{u \to i}^{(t)}(x_i) = \max_{1 \le k \le m-1} \frac{(\hat{x}_{k+1} - x_i) J_{u \to i}^{(t)}(\hat{x}_{k+1})}{\hat{x}_{k+1} - \hat{x}_k} + \frac{(x_i - \hat{x}_k) J_{u \to i}^{(t)}(\hat{x}_k)}{\hat{x}_{k+1} - \hat{x}_k}$$

for all $x_i \in \mathbb{R}$. As opposed to the case of quadratic approximations, where each message is parameterized by two numerical values, the number of parameters for each piecewise linear message grows with m. Hence, we anticipate that for fine-grain approximations, our second approach is likely to require greater computational resources. On the other hand, piecewise linear approximations may extend more effectively to non-convex problems, since non-convex messages are unlikely to be well-approximated by convex quadratic functions.

VI. Open Issues

There are many open questions in the theory of message passing algorithms. They fuel a growing research community that cuts across communications, artificial intelligence, statistical physics, signal processing, and operations research. This paper has focused on application of the min-sum message passing algorithm to convex programs, and even in this context a number of interesting issues remain unresolved.

Our proof technique establishes convergence under total asynchronism assuming a scaled diagonal dominance condition. With such a flexible model of asynchronous computation, convergence results for gradient descent and coordinate descent also require similar diagonal dominance assumptions. On the other hand, for the *partially asynchronous* setting, where communication delays and times between successive updates are

bounded, such assumptions are no longer required to guarantee convergence of these two algorithms. It would be interesting to see whether convergence of the min-sum algorithm under partial asynchronism can be established in the absence of scaled diagonal dominance.

Another direction will be to assess practical value of the min-sum algorithm for convex optimization problems. This calls for theoretical or empirical analysis of convergence and convergence times for implementable variants as those proposed in the previous section. Some convergence time results for a special case reported in [18] may provide a starting point. Our expectation is that for most relevant centralized optimization problems, the min-sum algorithm will be more efficient than gradient descent or coordinate descent but fall short of Newton's method. On the other hand, Newton's method does not decentralize gracefully, so in applications that call for decentralized solution, the min-sum algorithm may prove to be useful.

Finally, it would be interesting to explore whether ideas from this paper can be helpful in analyzing behavior of the min-sum algorithm for non-convex programs. It is encouraging that convex optimization theory has more broadly proved to be useful in designing and analyzing approximation methods for non-convex programs.

REFERENCES

[1] R. G. Gallager, *Low-Density Parity Check Codes*. Cambridge, MA: MIT Press, 1963.

[2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding," in *Proc. Int. Commun. Conf.*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[3] 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.

[4] M. Mézard, G. Parisi, and R. Zecchina, "Analytic and algorithmic solutions to random satisfiability problems," *Science*, vol. 297, no. 5582, pp. 812–815, 2002.

[5] A. Braunstein, M. Mézard, and R. Zecchina, "Survey propagation: An algorithm for satisfiability," *Random Struct. Algorithms*, vol. 27, no. 2, pp. 201–226, 2005.

[6] Y. Weiss and W. T. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," *Neural Computat.*, vol. 13, pp. 2173–2200, 2001.

[7] P. Rusmevichientong and B. Van 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.

[8] M. J. Wainwright, T. Jaakkola, and A. S. Willsky, "Tree-based reparameterization framework for analysis of sum-product and related algorithms," *IEEE Trans. Inf. Theory*, vol. 49, no. 5, pp. 1120–1146, 2003.

[9] 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, May 2009.

[10] 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, Oct. 2006.

[11] M. Bayati, C. Borgs, J. Chayes, and R. Zecchina, "On the exactness of the cavity method for weighted b-matchings on arbitrary graphs and its relation to linear programs," *J. Statist. Mechan.*, vol. 6, p. L06001, 2008.

[12] M. Bayati, D. Shah, and M. Sharma, "Max-product for maximum weight matching: Convergence, correctness, and LP duality," *IEEE Trans. Inf. Theory*, vol. 54, no. 3, pp. 1241–1251, Mar. 2008.

[13] B. Huang and T. Jebara, "Loopy belief propagation for bipartite maximum weight b-matching," in *Proc. Artif. Intell. Statist. (AISTATS) Conf.*, 2007.

[14] S. Sanghavi, D. Malioutov, and A. Willsky, Belief Propagation and LP Relaxation for Weighted Matching in General Graphs 2008, working paper.

[15] S. Sanghavi, D. Shah, and A. Willsky, "Message passing for maximum weight independent set," *IEEE Trans. Inf. Theory*, 2008, to be published.

[16] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 1995.

[17] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K. : Cambridge Univ. Press, 2004.

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

[19] A. Montanari, B. Prabhakar, and D. Tse, "Belief propagation based multi-user detection," in *Proc. Allerton Conf. Commun., Control, and Comput.*, 2005.

[20] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1990.

[21] N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation, Linköping Univ., Linköping, Sweden, 1996.

[22] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Belmont, MA: Athena Scientific, 1997.

Ciamac C. Moallemi (M'07) received the S.B. degrees in electrical engineering and computer science and in mathematics from the Massachusetts Institute of Technology, Cambridge, in 1996. He received the Certificate of Advanced Study in Mathematics, with distinction in 1997 from the University of Cambridge, U.K. He received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 2007.

He joined the Graduate School of Business, Columbia University, New York, in 2007, where he is currently an Assistant Professor.

Dr. Moallemi is a member of INFORMS. He is the recipient of a British Marshall Scholarship (1996) and a Benchmark Stanford Graduate Fellowship (2003).

Benjamin Van Roy (M'99–SM'07) received the S.B. degree in computer science and engineering and the S.M. and Ph.D. degrees in electrical engineering and computer science, all from the Massachusetts Institute of Technology (MIT), Cambridge, in 1993, 1995, and 1998, respectively.

He is an Associate Professor of Management Science and Engineering, Electrical Engineering, and, by courtesy, Computer Science, at Stanford University, Stanford, CA. He has held visiting positions as the Wolfgang and Helga Gaul Visiting Professor with the University of Karlsruhe and as the Chin Sophonpanich Foundation Professor of Banking and Finance, Chulalongkorn University.

Dr. Van Roy is a member of INFORMS. He has served on the editorial boards of Discrete Event Dynamic Systems, Machine Learning, Mathematics of Operations Research, and Operations Research. He has been a recipient of the MIT George C. Newton Undergraduate Laboratory Project Award in 1993, the MIT Morris J. Levin Memorial Master's Thesis Award in 1995, the MIT George M. Sprowls Doctoral Dissertation Award in 1998, the NSF CAREER Award in 2000, and the Stanford Tau Beta Pi Award for Excellence in Undergraduate Teaching in 2003. He has been a Frederick E. Terman Fellow and a David Morgenthaler II Faculty Scholar.