Journal of Machine Learning Research 20 (2019) 1-37 Submitted 5/17; Revised 12/18; Published 2/19

A New Approach to Laplacian Solvers and Flow Problems

Patrick Rebeschini PATRICK.REBESCHINI@STATS.OX.AC.UK
Department of Statistics
University of Oxford
Oxford, OX1 3LB, UK

Sekhar Tatikonda SEKHAR.TATIKONDA@YALE.EDU
Department of Statistics and Data Science
Yale University
New Haven, CT 06511, USA

Editor: Francis Bach

Abstract

This paper investigates the behavior of the Min-Sum message passing scheme to solve systems of linear equations in the Laplacian matrices of graphs and to compute electric flows. Voltage and flow problems involve the minimization of quadratic functions and are fundamental primitives that arise in several domains. Algorithms that have been proposed are typically centralized and involve multiple graph-theoretic constructions or sampling mechanisms that make them difficult to implement and analyze. On the other hand, message passing routines are distributed, simple, and easy to implement. In this paper we establish a framework to analyze Min-Sum to solve voltage and flow problems. We characterize the error committed by the algorithm on general weighted graphs in terms of hitting times of random walks defined on the computation trees that support the operations of the algorithms with time. For d-regular graphs with equal weights, we show that the convergence of the algorithms is controlled by the total variation distance between the distributions of non-backtracking random walks defined on the original graph that start from neighboring nodes. The framework that we introduce extends the analysis of Min-Sum to settings where the contraction arguments previously considered in the literature (based on the assumption of walk summability or scaled diagonal dominance) can not be used, possibly in the presence of constraints.

Keywords: Min-Sum, message passing, decentralized algorithms, Laplacian solver, network flows.

1. Introduction

Voltage and flow problems. Laplacian matrices of graphs are fundamental. Solving linear systems of equations in these matrices constitutes the back-bone of many algorithms with applications to a wide variety of fields. See Spielman (2011); Vishnoi (2013). When a weighted graph is viewed as an electric circuit, where weighted edges represent resistors and vertices represent nodes to which some external current is applied, solving linear systems of equations in the Laplacian matrix of the graph corresponds to computing the voltages induced to each node by the external currents. For this reason, we refer to the Laplacian problem as the "voltage problem." Along with the problem of computing the voltages is the problem of computing the current induced across each resistor, which we refer to as the

©2019 Patrick Rebeschini and Sekhar Tatikonda.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/17-286.html.

"flow problem." The two problems are related by Ohm's law—which says that electric flows are induced by voltage differences across the nodes—and they both can be cast in the form of quadratic optimization problems. The voltage problem is an unconstrained problem to find the voltage assignment that maximizes the quadratic Laplacian form. The flow problem is a constrained problem to find the flow assignment that minimizes the circuit energy and satisfies Kirchhoff's law. The voltage problem is the dual of the flow problem.

Previous literature on Laplacian solvers. The extensive literature on approximately solving the voltage and flow problems typically involves algorithms that rely on multiple graph-theoretic constructions or sampling mechanisms which make them difficult to implement and analyze. The first quasi-linear time solver was given in Spielman and Teng (2004, 2014), and it achieves a running time of $\tilde{O}(m \log^c n \log 1/\varepsilon)$ in the Laplacian-modified ℓ_2 norm. Here, n and m are, respectively, the number of vertices and edges in the graph, c is a constant close to 100, and the $\tilde{O}$ notation hides $\log \log n$ factors. Since this result, many algorithms have been proposed and showed to yield smaller values of c. The work in Koutis et al. (2011) has reduced the constant c to 1. Currently, the best running time is in Cohen et al. (2014) where $c = 1/2$. These algorithms all rely on the same general architecture, which combines the Chebyshev method or the conjugate gradient algorithm with the use of recursive preconditioning, spectral sparsifiers, low-stretch spanning trees, and expander graphs. Despite their theoretical guarantees, these algorithms present noticeable difficulties to practitioners due to the involved machinery they rely upon. Lately, new algorithms have been proposed that depart from this framework and use fewer graph-theoretic constructions. The method in Kelner et al. (2013), for instance, which operates in the flow domain and yields a fast solver with $c = 2$, using very little of the previous machinery—although it still requires low-stretch spanning trees. More recently, the work in Kyng and Sachdeva (2016), which relies on sparse Gaussian elimination for Laplacian matrices to yield a fast solver with $c = 3$, without the use of any graph-theoretic construction. While more appealing, these new algorithms still rely on random sampling so the convergence analysis is only performed in expectation, or with high probability. Moreover, all these algorithms are centralized. There has also been some work in parallelizing these solvers, with the first parallel solver with near linear work and poly-logarithmic depth presented in Peng and Spielman (2014). These solvers work in the shared memory model and are not completely distributed.

Min-Sum algorithm. The aforementioned limitations prompted us to investigate the behavior of the Min-Sum algorithm to solve both the voltage and the flow problem. Min-Sum is a distributed, deterministic (i.e., not randomized), general-purpose message passing algorithm that is used to optimize objective functions that are sums of component functions supported on a given graph. Min-Sum has emerged as a canonical procedure to address large scale problems in a variety of domains. Its success comes from the fact that the algorithm is simple and easy to implement, require minimal data structures and low communication per iteration, and in many applications it has been shown to converge to the problem solution. Interest in Min-Sum has been triggered by its success in solving certain classes of challenging combinatorial optimization problems, such as decoding low-density parity-check codes, see Berrou et al. (1993); Richardson and Urbanke (2001) for instance, and solving certain type of satisfiability problems, see Mezard et al. (2002); Braunstein et al. (2005), for instance.

Despite its successes, the convergence analysis of the Min-Sum algorithm remains limited. For quadratic optimization problems and, more generally, convex optimization prob-

lems, the Min-Sum algorithm has been shown to converge to the problem solution for *any* graph topology under the assumption that the objective function is walk-summable (Malioutov et al., 2006), or, equivalently, scaled diagonal dominant (Moallemi and Van Roy, 2010). These conditions allow the use of contraction arguments to establish exponential convergence rates. These conditions do not apply to quadratic objective functions in Laplacian matrices, hence to the voltage problem, and they do not apply to constrained optimization problems, hence to the flow problem. At the same time, on the one hand quadratic approximations of the Min-Sum algorithm leading to the Approximate Message Passing (AMP) algorithm are becoming increasingly popular in high-dimensional statistics and machine learning (Donoho et al., 2009; Montanari, 2012; Barbier et al., 2018). The AMP algorithm is typically analysed from an asymptotical point of view and this analysis is *not* based on contraction arguments. The recent success of the AMP algorithm prompts for improving the understanding of the convergence and correctness of the Min-Sum algorithm directly applied to quadratic models, where previously-considered contraction-based assumptions do not hold. On the other hand, while some work has been done to study the behavior of Min-Sum in constrained settings, this work seems to have focused on linear objective functions and a general theory is missing. See Gamarnik et al. (2012) and references therein.

Results. This paper investigates the convergence behavior of the Min-Sum algorithm applied to the voltage and flow problems and establishes a general framework for the analysis of message-passing algorithms that can accommodate for non contraction-based assumptions and for constrained-based optimization problems. The algorithms that we analyze—Algorithm 3 for the voltage problem and Algorithm 5 for the flow problem—are easy to implement, with a pseudo-code that fits two lines. We investigate the behavior of these algorithms on general weighted graphs, and we show that the error they incur can be characterized in terms of hitting times of ordinary diffusion random walks defined on the computation trees that are obtained by unraveling the operations of the algorithms with time (Weiss and Freeman, 2001). We specialize the analysis to the case of d-regular graphs. When $d = 2$, we show that the flow algorithm converges to the true solution in the ℓ_∞ norm with rate $O(1/t)$, where t is the iteration time, independently of the graph dimensions n and m, yielding a ε-approximate solution with a running time $O(m/\varepsilon)$, linear in m (Corollary 16). On the other hand, when $d = 2$, we show that the voltage algorithm does *not* converge, as it keeps oscillating (Corollary 10). For graphs with equal weights, when $d \geq 3$, we show that the error both algorithms commit is characterized by quantities involving the difference of non-backtracking random walks on the original graph that originate from neighbor nodes (Lemma 9). We show that the convergence rate of the algorithms (with respect to various norms) can be controlled by uniform bounds on the total variation distance between the distributions of these walks. For graphs where these bounds decay exponentially $O(e^{-t})$ (such as complete graphs, as we show numerically) or polynomially $O(1/t^\beta)$ with $\beta > 0$ (such as cycles and tori, where $\beta = 1/2$ as we show numerically), possibly under additional assumptions on the problem instance (depending on the norm chosen for the analysis), the Min-Sum algorithm yield ε-approximate solutions for both voltage and flow problem with time $O(m \log 1/\varepsilon)$ or $O(m/\varepsilon^{1/\beta})$, respectively, linear in m (Theorem 15 for voltages, and Theorem 18 for flows).

Novelty in the analysis of message passing. Our analysis of the Min-Sum algorithm presents two key contributions that go beyond the specifics of the problems here addressed.

1. Beyond contraction arguments. Typically, the convergence analysis of the Min-Sum scheme exploits a decay of correlation property in the computation tree established upon imposing *local* assumptions on the original problem, such as the assumption of scaled diagonal dominance. These conditions essentially yield a bound on the local interactions in the computation tree by a quantity strictly less than one. In this case, a contraction argument can be performed at each level of the tree to establish an *exponential* decay of correlation from root to leaves, which translates into an exponential convergence for the algorithm. On the other hand, the analysis of message passing that we present in this paper is based on a *global* analysis of the problem supported on the computation tree, via the analogy with electric circuits and the characterization with hitting times of random walks, which is not captured by contraction arguments based on worst-case bounds for local interactions. In fact, as mentioned above, for some graphs this analysis yields a convergence *polynomial* in time, not exponential.
2. Beyond unconstrained optimization. Typically, the convergence analysis of the Min-Sum algorithm relies on finding the fix point(s) of the message updates. For unconstrained optimization problems, the key insight to find fix point(s) is that the first order optimality conditions for the original problem are respected by the problem supported on the computation tree, for every node that is not a leaf (Weiss and Freeman, 2001; Moallemi and Van Roy, 2010). The convergence analysis that we present for the flow problem—a constrained problem—is based on the observation that the Karush-Kuhn-Tucker (KKT) optimality conditions for constrained problems are respected on the computation tree, for every non-leaf node. This insight allows to extend the analysis of the Min-Sum algorithm to more general constrained optimization problems.

Structure of the paper. The structure of the paper is as follows. In Section 2 we define the voltage and flow problems as optimization routines. In Section 3 we introduce the general-purpose Min-Sum algorithm, and we specialize it to the voltage and flow problem, respectively. In Section 4 we present the error characterization for the Min-Sum algorithms for d-regular graphs, and we present the main results on their convergence. In Section 5 we develop the general framework to investigate the Min-Sum algorithm, yielding the characterization of the error in terms of hitting times of random walks on the computation trees, and we present the proofs of the results given in the previous section. In Appendix A we establish a general connection between the inverse of restricted Laplacian matrices and hitting times of ordinary diffusion random walks on graphs—which is the main computational tool used in our analysis. Technical results are in Appendix B.

Remark 1 (Notation) *Given a matrix M, we denote by M^T its transpose, by M^{-1} its inverse, and by M^+ its Moore-Penrose pseudoinverse. Given a set $\mathcal{I}$ and $i \in \mathcal{I}$, we denote by i both the element $i \in \mathcal{I}$ and the subset $\{i\} \subseteq \mathcal{I}$ (e.g., we write $\mathcal{I} \backslash i$ to mean $\mathcal{I} \backslash \{i\}$). Given a function $f : \mathbb{R}^{\mathcal{I}} \to \mathbb{R}$, $i \in \mathcal{I}$, we write $f(x_{\mathcal{I}\backslash i} \cdot)$ to mean the function $z \in \mathbb{R} \to f(x_{\mathcal{I}\backslash i} z)$. We adopt the following taxonomy of notation:*

- *the arrow notation $\vec{\ }$ is used to either refer to quantities related to directed graphs or to refer to probability distributions related to non-backtracking random walks;*
- *the hat notation $\hat{\ }$ is used to refer to the output of algorithms;*
- *the bar notation $\bar{\ }$ is for quantities related to graphs where a node has been removed;*
- *the double-struck notation is for quantities related to computation trees, cf. Section 5.*

2. Voltage and Flow Problems

Throughout, let us consider a simple (i.e., no self-loops, and no multiple edges), undirected, weighted graph G given by the triple (V, E, W), where V is a set of n vertices, E is a set of m edges, and $W \in \mathbb{R}^{V \times V}$ is a symmetric matrix that assigns a positive weight to every edge, namely, $W_{vw} > 0$ if $\{v, w\} \in E$, and $W_{vw} = 0$ if $\{v, w\} \notin E$. The matrix W is called the weighted *adjacency matrix* of the weighed graph G. Define the weighted degree of a vertex v by $d_v := \sum_{w \in V} W_{vw}$, and let $D \in \mathbb{R}^{V \times V}$ be the diagonal matrix defined by $D_{vv} := d_v$. The Laplacian L of the graph G is the matrix defined as $L := D - W$. Without loss of generality, in what follows we assume that the graph G is connected, otherwise we can treat each connected component on its own. It can be verified that in this case the null space of the Laplacian matrix L is spanned by the all-ones vector $1 \in \mathbb{R}^V$, so that the range of L is spanned by the vectors orthogonal to 1.

Voltage problem. We are interested in solving linear equations in the Laplacian matrix L. That is, given b in the range of L, i.e., $1^T b = 0$, our goal is to find ν that satisfies

$$L\nu = b. \tag{1}$$

As it can immediately be checked from first order optimality conditions, the set of solutions to (1) coincides with the set of solutions to the "voltage problem," namely:

$$\text{maximize} \quad -\frac{1}{2}\nu^T L \nu + b^T \nu. \tag{2}$$

We are interested in the unique solution that is orthogonal to the all one vector, namely, $\nu^\star = L^+ b$, where L^+ is the Moore-Penrose pseudoinverse of L. Given an accuracy parameter $\varepsilon > 0$, the fast solvers mentioned in the introduction achieve a quasi-linear running time $\tilde{O}(m \log^c n \log 1/\varepsilon)$ to return an estimate $\hat{\nu}$ that satisfies $\|\nu^\star - \hat{\nu}\|_L \leq \varepsilon \|\nu^\star\|_L$, where $\|\nu\|_L := \sqrt{\nu^T L \nu}$ is the Laplacian-modified ℓ_2 norm, also known as L-norm. It can be seen that $\|\cdot\|_L$ is a pseudo-norm, as $\|\nu\|_L = 0$ for any vector ν orthogonal to the all one vector. In many applications where Laplacian solvers are needed, the L-norm is the natural norm to run the analysis. See Vishnoi (2013), for instance. The L-norm is related to the Euclidean norm as follow: $\|\nu\|_L \leq \kappa \|\nu\|_2$ and $\|\nu\|_2 \leq \kappa \|\nu\|_L$, where κ is the ratio between the the maximum and minimum non-zero eigenvalues of L.

Flow problem. To introduce the flow problem, let us fix an arbitrary orientation of the edges in E. Denote this set of directed edges by $\vec{E}$, and let $\vec{G} = (V, \vec{E})$ be the corresponding directed graph. Let $A \in \mathbb{R}^{V \times \vec{E}}$ be the *signed vertex-edge incidence* matrix of the graph $\vec{G}$, defined as $A_{ve} := 1$ if edge e leaves node v, $A_{ve} := -1$ if edge e enters node v, $A_{ve} := 0$ otherwise. To each directed edge $e = (v, w)$ let $R \in \mathbb{R}^{\vec{E} \times \vec{E}}$ be the diagonal matrix defined by $R_{ee} := 1/W_{vw}$. It is easy to verify that the Laplacian L of the undirected graph G can be expressed as $L = AR^{-1}A^T$. Note that while A depends on the choice of the direction of the edges in $\vec{E}$, L does not. The "flow problem" is defined as follows:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} x^T R x \\ \text{subject to} \quad & Ax = b. \end{aligned} \tag{3}$$

This problem can be interpreted as a minimal-energy electrical network problem. To each directed edge $e \in \vec{E}$ is associated a resistor of value R_{ee} (analogously, a conductor of value $1/R_{ee}$) through which a current x_e flows, with the convention that $x_e > 0$ if the current is in the direction of the edge, $x_e < 0$ if the current is in the direction opposite the edge. The energy of having current x_e on edge e is given by the $\frac{1}{2}R_{ee}x_e^2$. For each node $v \in V$, b_v represents a given external current: $b_v > 0$ represents a source where the current enters node v, whereas $b_v < 0$ represents a sink where the current exits node v. The constraint equations $Ax = b$ embody Kirchhoff's conservation law, by which at each vertex the net sum of the incoming and outgoing internal flows equals the external flow.

Connection between voltage and flow problems. From the KKT optimality conditions it can be shown (see Proposition 31 in Appendix B) that the unique solution of problem (3) reads $x^\star = R^{-1}A^T L^+ b = R^{-1}A^T\nu^\star$. That is, if $e = (v, w) \in \vec{E}$, we have $x_e^\star = W_{vw}(\nu_v^\star - \nu_w^\star)$ which represents Ohm's law. Finally, notice that problem (2) is the dual of (3). Recall that the Lagrangian of (3) is the function $\mathcal{L}$ from $\mathbb{R}^{\vec{E}} \times \mathbb{R}^V$ to $\mathbb{R}$ defined as

$$\mathcal{L}(x, \nu) := \frac{1}{2}x^T Rx + \nu^T(b - Ax). \tag{4}$$

The dual function is $q(\nu) = \min_{x\in\mathbb{R}^{\vec{E}}} \mathcal{L}(x, \nu) = -\frac{1}{2}\nu^T L\nu + b^T\nu$, and the dual problem reads $\max_{\nu\in\mathbb{R}^V} q(\nu)$, which is problem (2).

3. Min-Sum Algorithm

The Min-Sum algorithm is a distributed routine to optimize a cost function that has a graph structure. Let $\mathcal{V}$ and $\mathcal{F}$ be two finite sets, whose elements are respectively referred to as *variables* and *factors*, to be interpreted as two groups of vertices in a certain undirected bipartite graph with edge set $\mathcal{E}$, where each edge in $\mathcal{E}$ connects an element of $\mathcal{V}$ with an element of $\mathcal{F}$. For any variable node $i \in \mathcal{V}$, we use the notation ∂i to denote the set of neighbors of vertex i in the factor graph. Analogously, for any factor node $a \in \mathcal{F}$, we use the notation ∂a to denote the set of neighbors of vertex a in the factor graph. Consider the following optimization problem over $x \in \mathbb{R}^{\mathcal{V}}$:

$$\text{minimize} \quad g(x) := \sum_{i\in\mathcal{V}} g_i(x_i) + \sum_{a\in\mathcal{F}} g_a(x_{\partial a}), \tag{5}$$

for a given set of functions $g_i : \mathbb{R} \to \mathbb{R} \cup \{\infty\}$, for each $i \in \mathcal{V}$, and $g_a : \mathbb{R}^{\partial a} \to \mathbb{R} \cup \{\infty\}$, for each $a \in \mathcal{F}$. The Min-Sum algorithm is an iterative algorithm that proceeds by updating a set of *messages* associated to each edge in the underlying factor graph. Messages are functions of the optimization variables x_i, for $i \in \mathcal{V}$, and each edge in the factor graph is associated with two messages at any given iteration of the algorithm. For any time $s \geq 0$, and $i \in \mathcal{V}, a \in \mathcal{F}$ so that $\{i, a\} \in \mathcal{E}$, $\mu_{i\to a}^s : \mathbb{R} \to \mathbb{R} \cup \{\infty\}$ represents a message from variable to factor, and $\mu_{a\to i}^s : \mathbb{R} \to \mathbb{R} \cup \{\infty\}$ represents a message from factor to variable.

In this work we consider the *synchronous* implementation of the Min-Sum algorithm, where at each iteration all messages are updated. For a given iteration step $t \geq 1$, the algorithm reads as in Algorithm 1.

The iterations of the Min-Sum algorithm correspond to the dynamic programming iterations to solve (5) when the underlying graph is a tree. In general, the Min-Sum algorithm

Algorithm 1: Min-Sum

Input: Initial messages $\{\mu^0_{a\to i}\}$, for each $\{i,a\}\in\mathcal{E}$;
for $s\in\{1,\dots,t\}$ **do**
  Compute, for each $\{i,a\}\in\mathcal{E}$,
  $\mu^s_{i\to a}=g_i+\sum_{b\in\partial i\setminus a}\mu^{s-1}_{b\to i}$, $\quad\mu^s_{a\to i}=\min_{x\in\mathbb{R}^{\mathcal{V}}}\{g_a(x_{\partial a\setminus i}\cdot)+\sum_{j\in\partial a\setminus i}\mu^s_{j\to a}(x_j)\}$;
Compute $\mu^t_i=g_i+\sum_{a\in\partial i}\mu^t_{a\to i}$, for each $i\in\mathcal{V}$;
Output: $\hat{x}^t_i:=\arg\min_{z\in\mathbb{R}}\mu^t_i(z)$, for each $i\in\mathcal{V}$.

is not guaranteed to converge, and even when it converges it is not guaranteed that it converges to an optimal point of problem (5). The Min-Sum algorithm is suited for cooperative distributed multi-agents optimization. In this framework, to each variable $i\in\mathcal{V}$ is associated an agent who has access only to the local objective functions g_i and g_a, for each $a\in\partial i$. At each time step, each agent can only exchange information with his near-neighbors in the factor graph, i.e., the agents who share at least one factor with him. Agents' goal is to cooperate over time to solve problem (5). The Min-Sum algorithm is a mean to achieve this goal: iteratively, each agent i updates a set of messages $\{\mu^s_{i\to a}\}, a\in\partial i$, using the messages that he has received from his neighbors at the previous time step, i.e., $\{\mu^{s-1}_{j\to a}\}, a\in\partial j$, for each neighbor j of i, and using the locally known functions g_i and g_a, for each $a\in\partial i$.

Remark 2 (Belief Propagation) *The Min-Sum algorithm alternates minimizations and summations to compute the minimum of a sum of functions of the form in* (5)*. The main property that is used to derive the local updates of this algorithm (on a tree, where the algorithm computes the correct solution) is the fact that the minimum distributes over the summation. Closely related to the Min-Sum algorithm is the Sum-Product algorithm, where the goal is to compute the sum (or integral) of a product of functions of the following form:*

$$\prod_{i\in\mathcal{V}}g_i(x_i)\prod_{a\in\mathcal{F}}g_a(x_{\partial a}). \tag{6}$$

As the sum (or integral) distributes over the product, the Sum-Product algorithm takes the exact same form as Algorithm 1 with minima replaced by sums and sums replaced by products. In the machine learning community, the Sum-Product algorithm is usually referred to as Belief Propagation, and a classical application involves the computation of the normalization function of graphical models, where random variables $\{X_v\}_{v\in V}$ have a joint density that factorizes as in (6)*. While Min-Sum and Sum-Product are formally related, they are known to produce results that are qualitatively different. In particular, while there are sufficient conditions that guarantee the convergence and correctness of Min-Sum in graphs of arbitrary topology, the Sum-Product algorithm is known to yield correct results only on locally tree-like graphs (Weiss and Freeman, 2001).*

We now specialize the Min-Sum algorithm to the voltage and flow problem, respectively.

3.1. Min-Sum for the Voltage Problem

In order to apply the Min-Sum algorithm to solve the voltage problem (2), throughout the rest of this paper we consider the equivalent minimization form: minimize $\frac{1}{2}\nu^TL\nu-b^T\nu$. We

rewrite this problem in the form (5).[1] To this end, let us identify $\mathcal{V} \leftrightarrow V$ and $\mathcal{F} \leftrightarrow E$. In this context, the notation ∂v indicates the set of edges in E are connected to $v \in V$; analogously, the notation ∂e indicates the pair of vertices that are connected by edge e. For each $v \in V$, define the function $g_v(\nu_v) = -b_v\nu_v$, and for each $e \in E$ define the function $g_e(\nu_{\partial e}) := \frac{1}{2}W_{vw}(\nu_v - \nu_w)^2$, where $\partial e = \{v, w\}$. Recalling that $\nu^T L\nu = \sum_{\{v,w\}\in E} W_{vw}(\nu_v - \nu_w)^2$, the problem above is equivalent to the problem of minimizing the function $\sum_{v\in V} g_v(\nu_v) + \sum_{e\in E} g_e(\nu_{\partial e})$, which is of the form (5). As in the factor graph interpretation each factor node (i.e., edge in E) is adjacent to exactly two variable nodes (i.e., vertices in V), it is convenient to rewrite the Min-Sum algorithm in terms of updates of messages from factors to variables only. At each time step $t \geq 1$, to each edge $e = \{v, w\} \in E$ are associated two messages: $\mu^t_{e\to v}$ and $\mu^t_{e\to w}$. It is easy to verify that in the present setting Algorithm 1 becomes Algorithm 2.

Algorithm 2: Min-Sum, voltage problem

Input: Initial messages $\{\mu^0_{e\to v}\}$, for $e \in E, v \in \partial e$;
for $s \in \{1, \dots, t\}$ **do**
  Compute, for each $e \in E$, $v \in \partial e$, with $w = \partial e \setminus v$
  $\mu^s_{e\to v} = \min_{\nu_w\in\mathbb{R}}\{g_w(\nu_w) + g_e(\nu_w\,\cdot) + \sum_{f\in\partial w\setminus e} \mu^{s-1}_{f\to w}(\nu_w)\}$;
Compute $\mu^t_v = g_v + \sum_{e\in\partial v} \mu^t_{e\to v}$, for each $v \in V$;
Output: $\hat{\nu}^t_v := \arg\min_{z\in\mathbb{R}} \mu^t_v(z)$, for each $v \in V$.

Algorithm 2 applies for any choice of the initial messages $\{\mu^0_{e\to v}\}$, for $e \in E, v \in \partial e$. If we choose initial messages that are quadratic functions, then the messages computed by the algorithm remain quadratic. The following Proposition presents the quadratic updates, without stating formulas for the constant terms as these terms do not influence the output of the Min-Sum algorithm, as we will show next.

Proposition 3 *For a given $s \geq 1$, for each $e \in E$, $v \in \partial e$ let $\mu^{s-1}_{e\to v}(z) = \frac{1}{2}W^{s-1}_{e\to v}z^2 + w^{s-1}_{e\to v}z + c^{s-1}_{e\to v}$. Then, for each $e \in E$, $v \in \partial e$, we have $\mu^s_{e\to v}(z) = \frac{1}{2}W^s_{e\to v}z^2 + w^s_{e\to v}z + c^s_{e\to v}$, where $\{c^s_{e\to v}\}$ is a certain set of constants, and (here $w = \partial e \setminus v$)*

$$W^s_{e\to v} = W_{vw}\frac{\sum_{f\in\partial w\setminus e} W^{s-1}_{f\to w}}{W_{vw} + \sum_{f\in\partial w\setminus e} W^{s-1}_{f\to w}}, \qquad w^s_{e\to v} = W_{vw}\frac{\sum_{f\in\partial w\setminus e} w^{s-1}_{f\to w} - b_w}{W_{vw} + \sum_{f\in\partial w\setminus e} W^{s-1}_{f\to w}}.$$

Proof Fix $s \geq 1$, $e \in E$, $v \in \partial e$, and let $w = \partial e \setminus v$. The Min-Sum update rule reads $\mu^s_{e\to v}(\nu_v) = \min_{z\in\mathbb{R}}\{-b_w z + \frac{1}{2}W_{vw}(\nu_v - z)^2 + \frac{1}{2}z^2\sum_{f\in\partial w\setminus e} W^{s-1}_{f\to w} + z\sum_{f\in\partial w\setminus e} w^{s-1}_{f\to w} + \sum_{f\in\partial w\setminus e} c^{s-1}_{f\to w}\}$. The solution is $z^\star = (W_{vw}\nu_v + b_w - \sum_{f\in\partial w\setminus e} w^{s-1}_{f\to w})/(W_{vw} + \sum_{f\in\partial w\setminus e} W^{s-1}_{f\to w})$, and the proof is immediately concluded by substitution. ■

From Proposition 3 it follows that if we initialize Algorithm 2 with quadratic messages of the form $\mu^0_{e\to v}(z) = \frac{1}{2}W^0_{e\to v}z^2 + w^0_{e\to v}z$, then, modulo the constant terms, we can track the evolution of the quadratic messages computed by the algorithm by evaluating recursively

1. It is well known that different ways of mapping the original optimization problem to the form (5) yield different convergence results. See Moallemi and Roy (2009), for instance.

the set of parameters $\{W^s_{e\to v}\}, \{w^s_{e\to v}\}$ as prescribed by Proposition 3. The belief function associated to vertex v at time t then reads $\mu^t_v(z) = -b_v z + \sum_{e\in\partial v}(\frac{1}{2}W^t_{e\to v}z^2 + w^t_{e\to v}z + c^t_{e\to v})$, so that $\hat{\nu}^t_v = \arg\min_{z\in\mathbb{R}} \mu^t_v(z) = (b_v - \sum_{e\in\partial v} w^t_{e\to v})/(\sum_{e\in\partial v} W^t_{e\to v})$, independent of $\{c^t_{e\to v}\}$. The final procedure, in the case of quadratic initial messages, yields Algorithm 3.

Algorithm 3: Min-Sum, voltage problem, quadratic messages

Input: Initial messages $\{W^0_{e\to v}\}, \{w^0_{e\to v}\}$, $e \in E$,$v \in \partial e$;

for $s \in \{1, \dots, t\}$ **do**

Compute, for each $e \in E$, $v \in \partial e$, with $w = \partial e \setminus v$

$$W^s_{e\to v} = W_{vw}\frac{\sum_{f\in\partial w\setminus e} W^{s-1}_{f\to w}}{W_{vw}+\sum_{f\in\partial w\setminus e} W^{s-1}_{f\to w}}, \qquad w^s_{e\to v} = -W_{vw}\frac{b_w - \sum_{f\in\partial w\setminus e} w^{s-1}_{f\to w}}{W_{vw}+\sum_{f\in\partial w\setminus e} W^{s-1}_{f\to w}};$$

Output: $\hat{\nu}^t_v = \frac{b_v - \sum_{e\in\partial v} w^t_{e\to v}}{\sum_{e\in\partial v} W^t_{e\to v}}$, for each $v \in V$.

Algorithm 3, with the choice of initial conditions $W^0_{e\to v} = W_{vw}$ and $w^0_{e\to v} = 0$, for each $e = \{v, w\} \in E, v \in \partial e$, will be the focus of our investigation for the voltage problem. The convergence analysis will rely on the general formulation of Algorithm 2.

Remark 4 (Gaussian Belief Propagation) *When applied to quadratic objective functions, the Min-Sum algorithm (e.g., Algorithm 3) corresponds to the computation of the mean performed by the Belief Propagation algorithm applied to Gaussian graphical models (Weiss and Freeman, 2001). This connection is due to the fact that in the Gaussian case the mode of a distribution coincides with its mean. Min-Sum can be seen as computing the maximum a posteriori (MAP) estimate of the graphical model (i.e., the mode), which corresponds to the mean computed by Belief Propagation. Typically, the convergence and correctness of the mean of Gaussian Belief Propagation is investigated within the framework of walk-summability (Malioutov et al., 2006), or, equivalently, scaled diagonal dominance (Moallemi and Van Roy, 2010). As the Laplacian L is not positive definite, the existing framework can not be applied to investigate the convergence behavior of Algorithm 3. In this paper, we overcome this limitation by using a different approach based on the global analysis of the computation tree that supports the operations of the algorithm with time.*

3.2. Min-Sum for the Flow Problem

To apply the Min-Sum algorithm to solve the flow problem (3), we first rewrite this problem in the unconstrained form (5). To this end, let us identify $\mathcal{V} \leftrightarrow \vec{E}$ and $\mathcal{F} \leftrightarrow V$. In this context, the notation ∂v indicates the set of edges in $\vec{E}$ that either enter or leave node $v \in V$; analogously, the notation ∂e indicates the pair of vertices that are connected by edge e. For each $e \in \vec{E}$, define the function $g_e(x_e) = \frac{1}{2}R_{ee}x_e^2$, and for each $v \in V$ define the function $g_v(x_{\partial v}) := 0$ if $\sum_{e\in\vec{E}} A_{ve}x_e = b_v$, and $g_v(x_{\partial v}) := \infty$ otherwise. Clearly, problem (3) is equivalent to the problem of minimizing the function $\sum_{e\in\vec{E}} g_e(x_e) + \sum_{v\in V} g_v(x_{\partial v})$, which is of the form (5). As in the factor graph interpretation each variable node (i.e., edge in $\vec{E}$) is adjacent to exactly two factor nodes (i.e., vertices in V), it is convenient to rewrite the Min-Sum algorithm in terms of updates of messages from variables to factors only. At each time step $t \geq 1$, to each edge $e \in \vec{E}$ such that $\partial e = \{v, w\}$ (that is, either $e = (v, w)$

or $e = (w, v)$) are associated two messages: $\mu^t_{e\to v}$ and $\mu^t_{e\to w}$. It can be verified that in the present setting Algorithm 1 becomes Algorithm 4.[2]

Algorithm 4: Min-Sum, flow problem

Input: Initial messages $\{\mu^0_{e\to v}\}$, for $e \in \vec{E}, v \in \partial e$;
for $s \in \{1, \dots, t\}$ **do**
  Compute, for each $e \in \vec{E}$, $v \in \partial e$, with $w = \partial e \setminus v$
  $\mu^s_{e\to v} = g_e + \min_{x\in\mathbb{R}^{\vec{E}}} \left\{ g_w(x_{\partial w\setminus e}\,\cdot) + \sum_{f\in\partial w\setminus e} \mu^{s-1}_{f\to w}(x_f) \right\}$;
Compute $\mu^t_e = \mu^t_{e\to v} + \mu^t_{e\to w} - g_e$, for $e = (v, w) \in \vec{E}$;
Output: $\hat{x}^t_e := \arg\min_{z\in\mathbb{R}} \mu^t_e(z)$, for each $e \in \vec{E}$.

Note that if $\partial w \setminus e = \varnothing$, then the message $\mu^s_{e\to v}$ in Algorithm 4 reads $\mu^s_{e\to v}(z) = \frac{1}{2} R_{ee} b_w^2$ if $z = A_{we} b_w$, and $\mu^s_{e\to v}(z) = \infty$ otherwise, for any $s \geq 1$, irrespective of the choice of the initial messages. This is a consequence of the fact that if node $w \in V$ is a *leaf* of G (i.e., w has degree 1), then by Kirchhoff's law the value x_e of the flow on the unique edge e that is associated to w is $x_e = A_{we} b_w$. Without loss of generality, in what follows we assume that the graph G has no leaves. Otherwise, by removing one leaf at a time as described above, we can reduce the original problem to a new problem supported on a graph with no leaves (if G is a tree we can exactly compute all the components of the solution $x^\star$ in this way).

Algorithm 4 applies for any choice of the initial messages $\{\mu^0_{e\to v}\}$, for $e \in \vec{E}, v \in \partial e$. If we choose initial messages that are quadratic functions, then also the following messages computed by the algorithm remain quadratic. The next Lemma presents the quadratic updates, without stating formulas for the constant terms as these terms do not influence the output of the Min-Sum algorithm, as we will show next.

Proposition 5 *For a given $s \geq 1$, for each $e \in \vec{E}$, $v \in \partial e$ let $\mu^{s-1}_{e\to v}(z) = \frac{1}{2} R^{s-1}_{e\to v} z^2 + r^{s-1}_{e\to v} z + c^{s-1}_{e\to v}$. Then, if the graph G has no leaves, for each $e \in \vec{E}$, $v \in \partial e$, we have $\mu^s_{e\to v}(z) = \frac{1}{2} R^s_{e\to v} z^2 + r^s_{e\to v} z + c^s_{e\to v}$, where $\{c^s_{e\to v}\}$ is a set of constants, and (here $w = \partial e \setminus v$)*

$$R^s_{e\to v} = R_{ee} + \frac{1}{\sum_{f\in\partial w\setminus e} 1/R^{s-1}_{f\to w}}, \qquad r^s_{e\to v} = -A_{we} \frac{\sum_{f\in\partial w\setminus e} A_{wf} r^{s-1}_{f\to w}/R^{s-1}_{f\to w} + b_w}{\sum_{f\in\partial w\setminus e} 1/R^{s-1}_{f\to w}}.$$

Proof Fix $s \geq 1$, $e \in \vec{E}$, $v \in \partial e$, and let $w \in \partial e \setminus v$. As we assume that the graph G has no leaves, then $\partial w \setminus e \neq \varnothing$ and the Min-Sum update rule reads $\mu^s_{e\to v}(z) = g_e(z) + \min_{x\in\mathbb{R}^{\partial w\setminus e} : \tilde{A}x = b_w - A_{we} z} \sum_{f\in\partial w\setminus e} \mu^{s-1}_{f\to w}(x_f)$, where $\tilde{A} := A_{w,\partial w\setminus e}$ is the submatrix of A given by the w-th row and the columns indexed by the elements in $\partial w \setminus e$. The minimization in the definition of $\mu^s_{e\to v}$ can be written as

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} x^T \tilde{R} x + \tilde{r}^T x + 1^T \tilde{c} \\ \text{subject to} \quad & \tilde{A} x = b_w - A_{we} z \end{aligned} \tag{7}$$

2. For convenience of notation, we present Algorithm 4 with initial messages indexed by time 0. Messages are indexed by a time parameter that is shifted by one unit compared to Algorithm 1.

over $x \in \mathbb{R}^{\partial w \setminus e}$, with $\tilde{R} \in \mathbb{R}^{(\partial w \setminus e) \times (\partial w \setminus e)}$ diagonal with $\tilde{R}_{ff} := R^{s-1}_{f \to w}$, $\tilde{r} \in \mathbb{R}^{\partial w \setminus e}$ with $\tilde{r}_f := r^{s-1}_{f \to w}$, and $\tilde{c} \in \mathbb{R}^{\partial w \setminus e}$ with $\tilde{c}_f := c^{s-1}_{f \to w}$. By Proposition 31 in Appendix B the solution of (7) reads $x^\star = h - A_{we} z y$, with $h := b_w y - (I - y\tilde{A})\tilde{R}^{-1}\tilde{r}$, $y := \tilde{R}^{-1}\tilde{A}^T\tilde{L}^{-1}$, and $\tilde{L} := \tilde{A}\tilde{R}^{-1}\tilde{A}^T \in \mathbb{R}$. Substituting $x^\star$ in (7) we get $\mu^s_{e \to v}(z) = \frac{1}{2}R^s_{e \to v} z^2 + r^s_{e \to v} z + c^s_{e \to v}$, with $R^s_{e \to v} := R_{ee} + y^T \tilde{R} y$, and $r^s_{e \to v} := -A_{we}(y^T \tilde{R} h + \tilde{r}^T y)$. The proof follows as $\tilde{L} = \sum_{f \in \partial w \setminus e} 1/R^{s-1}_{f \to w}$, $y^T \tilde{R} y = \tilde{L}^{-1}$, $y^T \tilde{R} h = \tilde{L}^{-1} b_w$, $\tilde{r}^T y = \tilde{L}^{-1} \sum_{f \in \partial w \setminus e} A_{wf} r^{s-1}_{f \to w} / R^{s-1}_{f \to w}$. ■

From Proposition 5 it follows that if we initialize Algorithm 4 with quadratic messages $\mu^0_{e \to v}(z) = \frac{1}{2}R^0_{e \to v} z^2 + r^0_{e \to v} z$, then, modulo the constant terms, we can track the evolution of the quadratic messages computed by the algorithm by evaluating recursively the set of parameters $\{R^s_{e \to v}\}, \{r^s_{e \to v}\}$ as prescribed by Proposition 5. The belief function associated to edge e at time t then reads $\mu^t_e(z) = \frac{1}{2}(R^t_{e \to v} + R^t_{e \to w} - R_{ee}) z^2 + (r^t_{e \to v} + r^t_{e \to w}) z + c^t_e$, for a certain constant c^t_e, so that $\hat{x}^t_e = \arg\min_{z \in \mathbb{R}} \mu^t_e(z) = -(r^t_{e \to v} + r^t_{e \to w})/(R^t_{e \to v} + R^t_{e \to w} - R_{ee})$, independent of c^t_e. The final procedure so obtained, in the case of quadratic initial messages, for a graph G with no leaves, is given in Algorithm 5.

Algorithm 5: Min-Sum, flow problem, quadratic messages, no leaves

Input: Initial messages $\{R^0_{e \to v}\}, \{r^0_{e \to v}\}$, $e \in \vec{E}$,$v \in \partial e$;
for $s \in \{1, \dots, t\}$ **do**
  Compute, for each $e \in \vec{E}$, $v \in \partial e$, with $w = \partial e \setminus v$

$$R^s_{e \to v} = R_{ee} + \frac{1}{\sum_{f \in \partial w \setminus e} 1/R^{s-1}_{f \to w}}, \qquad r^s_{e \to v} = -A_{we} \frac{\sum_{f \in \partial w \setminus e} A_{wf} r^{s-1}_{f \to w} / R^{s-1}_{f \to w} + b_w}{\sum_{f \in \partial w \setminus e} 1/R^{s-1}_{f \to w}};$$

Output: $\hat{x}^t_e = -\frac{r^t_{e \to v} + r^t_{e \to w}}{R^t_{e \to v} + R^t_{e \to w} - R_{ee}}$, for $e = (v, w) \in \vec{E}$.

Algorithm 5, with the choice of initial conditions $R^0_{e \to v} = R_{ee}$, $r^0_{e \to v} = 0$, for each $e \in \vec{E}, v \in \partial e$, will be the focus of our investigation for the flow problem. The convergence analysis will rely on the general formulation of Algorithm 4.

4. Results for d-Regular Graphs

A d-regular graph is a graph where each node has d neighbors. In this section, we first characterize the error committed by the Min-Sum algorithm to solve the voltage and flow problems on connected d-regular graphs in terms of the solution $\nu^\star = L^+ b$ of the voltage problem. Then, we develop convergence results with respect to various norms.

The proofs of the main results here presented—Lemma 6 and Lemma 9—are given in Section 5, where we develop the general machinery to analyze the behavior of the Min-Sum scheme to solve constrained optimization problems, and we present the characterization of the error committed by the voltage and flow algorithms on general weighted graphs in terms of hitting times of random walks on the computation tree (for the sake of simplicity, we postpone the results on general graphs to Section 5, as these results require the notion of computation tree). The proofs of the technical results used here are given in Appendix B.

Henceforth, let $\hat{\nu}^t$ be the outcome of Algorithm 3 at time t with the choice of initial conditions $W^0_{e \to v} = W_{vw}$ and $w^0_{e \to v} = 0$, for each $e = \{v, w\} \in E, v \in \partial e$. Analogously, let $\hat{x}^t$

be the outcome of Algorithm 5 at time t with the choice of initial conditions $R^0_{e\to v} = R_{ee}$, $r^0_{e\to v} = 0$, for each $e \in \vec{E}, v \in \partial e$. Throughout, given a weighted directed graph $\vec{G} = (V, \vec{E}, W)$ with no bi-directed edges (i.e., either $(v,w) \in \vec{E}$ or $(w,v) \in \vec{E}$), let $G = (V, E, W)$ be the weighted undirected graph naturally associated to $\vec{G}$, i.e., edges in E are obtained by removing the orientation of the edges in $\vec{E}$. Let $n := |V|$ and $m := |E|$ be the number of nodes and edges, respectively.

4.1. Error Characterization

We characterize the error committed by the Min-Sum algorithm as a function of time and as a function of the voltage solution $\nu^\star = L^+ b$.

The case $d = 2$, i.e., a cycle graph, allows a characterization for general weighted graphs.

Lemma 6 (Regular graphs, $d = 2$) *Let $\vec{G} = (V = \{0, \dots, n-1\}, \vec{E}, W)$ be a weighted directed cycle, and let $G = (V, E, W)$ be the corresponding undirected graph. Define the function $i \in \mathbb{N} \to \rho(i) := i \mod n$. For $v \in V$, $e = \{\rho(v), \rho(v+1)\} \in E$, $t \geq 2$, we have*

$$\nu^\star_v - \hat{\nu}^t_v = \alpha_{v,t}\, \nu^\star_{\rho(v-t-1)} + (1 - \alpha_{v,t})\, \nu^\star_{\rho(v+t+1)}, \qquad \alpha_{v,t} := \frac{\sum_{k=v}^{v+t} 1/W_{\rho(k)\rho(k+1)}}{\sum_{k=v-t-1}^{v+t} 1/W_{\rho(k)\rho(k+1)}},$$

$$x^\star_e - \hat{x}^t_e = A_{ve}\, \beta_{e,t}\, (\nu^\star_{\rho(v-t)} - \nu^\star_{\rho(v+t+1)}), \qquad \beta_{e,t} := \frac{1}{\sum_{k=v-t}^{v+t} 1/W_{\rho(k)\rho(k+1)}}.$$

If each edge has the same weight ω, then $\alpha_{v,t} = 1/2$ and $\beta_{e,t} = \omega/(2t+1)$.

In the case of constant weights, this result shows that Min-Sum displays a non-convergent (oscillatory) behavior in the voltage setting (as $\alpha_{v,t}$ does not decay with time), while Min-Sum displays a convergent behavior in the flow setting (as $\beta_{v,t}$ decays with time). We will discuss this behavior more extensively in the following sections.

Remark 7 *The fact that the Min-Sum algorithm does not converge is not surprising. Gaussian BP, see Remark 4, is known to suffer from divergent behaviors even in the absence of degeneracy, i.e., even when the quadratic form of interest is defined by a positive definite matrix, as observed numerically in Rusmevichientong and Roy (2001).*

For $d \geq 3$ we can provide a characterization of the error in the case of graphs with equal weights. We begin by defining and bounding some quantities of interest that we will need to state our results. The proof of the following proposition is given in Appendix B.

Proposition 8 *Fix $d \geq 3$. For $s \geq 1$ let $h_s := \frac{1}{(d-1)^s - 1}$, and let $\delta_0 := \frac{1}{d}$, $\delta_1 := \frac{1}{d-1} + d - 1$, and $\delta_s := \frac{1}{d-1}(2 + \frac{(d-2)^2}{d-1}(1 + h_{s+2}))\delta_{s-1} - \frac{1}{(d-1)^2}\delta_{s-2}$ for $s \geq 2$. For any $t \geq 3$ define*

$$b_{d,t} := \frac{1}{(d-1)^2}\left(1 + (d-2)(1+h_{t+1})\right)\delta_{t-2} - \frac{(d-2)(1+h_{t+1})}{(d-1)^3}\delta_{t-3},$$

$$c_{d,t} := \frac{1}{d-1}\left(1 + \frac{1}{(d-2)(1+h_t)}\right)\delta_{t-2} - \frac{1}{(d-1)^2}\delta_{t-3}.$$

Then we have $\frac{1}{2} \leq \frac{d-2}{d-1} \leq b_{d,t} \leq c_{d,t} \leq 1 + \varepsilon_d < 4$ and $c_{d,t} \geq 1$, where ε_d is a positive decreasing function of $d \geq 3$ so that $\varepsilon_d \to 0$ as $d \to \infty$ and $\varepsilon_3 < 3$.

For each $v \in V$, let $\vec{\mathbf{P}}_v$ be the law of a time-homogeneous *non-backtracking* random walk $Y_0, Y_1, Y_2, \dots$ on the nodes of G with initial state $Y_0 = v$, which is defined by $\vec{\mathbf{P}}_v(Y_0 = w) := 1$ if $w = v$ and equal to 0 otherwise, by $\vec{\mathbf{P}}_v(Y_1 = w) := 1/d$ if $\{w, v\} \in E$ and equal to 0, and, for any $\{z, z'\} \in E$, by $\vec{\mathbf{P}}_v(Y_{t+1} = w | Y_t = z, Y_{t-1} = z') := \frac{1}{d-1}$ if $\{w, z\} \in E, w \neq z'$ and equal to 0 otherwise. For ease of notation, for $t \geq 0$, $w \in V$, define the row-stochastic matrices $P^{(t)}$ and $P^{(t,w)}$ as

$$P^{(t)}_{vz} := \vec{\mathbf{P}}_v(Y_t = z), \qquad P^{(t,w)}_{vz} := \vec{\mathbf{P}}_v(Y_t = z | Y_1 \neq w), \qquad v, z \in V. \tag{8}$$

We can now state the main result for d-regular graphs with $d \geq 3$ and equal weights, namely, $W_{vw} = \omega$ for any $\{v, w\} \in E$, for some $\omega > 0$.

Lemma 9 (Regular graphs, $d \geq 3$, equal weights) *Let $\vec{G} = (V, \vec{E}, W)$ be a connected directed d-regular graph, with $d \geq 3$, where each edge has the same weight $\omega > 0$, and let $G = (V, E, W)$ be the corresponding undirected graph. For $v \in V$, $e = (v, w) \in \vec{E}$, $t \geq 3$,*

$$\nu^\star_v - \hat{\nu}^t_v = \frac{1}{b_{d,t}} (P^{(t+1)} \nu^\star)_v, \qquad x^\star_e - \hat{x}^t_e = \frac{\omega}{c_{d,t}} \frac{d-1}{d} ((P^{(t,w)} \nu^\star)_v - (P^{(t,v)} \nu^\star)_w),$$

where $b_{d,t}$ and $c_{d,t}$ are defined and bounded as in Proposition 8.

This result characterizes the error committed by the Min-Sum algorithm in terms of the distribution of properly-defined non-backtracking random walks. Several papers have previously related the convergence properties of message-passing algorithms to non-backtracking random walks. For instance, we refer to Moallemi and Roy (2006) for an application to consensus optimization (this paper also analyzes d-regular graphs), and to Abbe (2018) for a more recent application to the Stochastic Block Model.

Given the error characterizations provided by Lemma 6 and Lemma 9, we now investigate and discuss the convergence properties of the Min-Sum algorithm.

4.2. Convergence Results for the Voltage Problem

We first analyze the convergence behavior of the Min-Sum algorithm when applied to the voltage problem: Algorithm 3. Following the literature on Laplacian solvers, as already noticed in Section 2, we perform the analysis in the L-norm, which is defined as $\|\nu\|_L := \sqrt{\nu^T L \nu}$ for any $\nu \in \mathbb{R}^V$, where $\nu^T L \nu = \sum_{\{v,w\} \in E} W_{vw} (\nu_v - \nu_w)^2$.

In the case $d = 2$, as previously discussed, the results of Lemma 6 attest that the output $\hat{\nu}^t$ of Algorithm 3 does *not* converge, as it keeps oscillating as a function of t. This is particularly evident in the case of equal weights, as the following corollary attests.

Corollary 10 (Regular graphs, $d = 2$) *Consider the setting of Lemma 6 with equal weights $W_{vw} = \omega$. For $t \geq 2$, $\|\nu^\star - \hat{\nu}^t\|^2_L = \frac{1}{2}\|\nu^\star\|^2_L + \frac{\omega}{2} \sum_{v=0}^{n-1} \nu^\star_v (2\nu^\star_{\rho(v+2t+2)} - \nu^\star_{\rho(v+2t+3)} - \nu^\star_{\rho(v+2t+1)})$.*

Proof The proof follows immediately from Lemma 6 noticing that $\|\nu^\star - \hat{\nu}^t\|^2_L$ equals $\omega \sum_{v=0}^{n-1} (\nu^\star_{v+1} - \hat{\nu}^t_{v+1} - (\nu^\star_v - \hat{\nu}^t_v))^2 = \frac{\omega}{4} \sum_{v=0}^{n-1} (\nu^\star_{\rho(v-t)} - \nu^\star_{\rho(v-t-1)} + \nu^\star_{\rho(v+t+2)} - \nu^\star_{\rho(v+t+1)})^2$, which reads $\frac{1}{2}\|\nu^\star\|^2_L + \frac{\omega}{2} \sum_{v=0}^{n-1} (\nu^\star_{\rho(v-t)} - \nu^\star_{\rho(v-t-1)})(\nu^\star_{\rho(v+t+2)} - \nu^\star_{\rho(v+t+1)})$. The result follows by grouping and relabeling the terms in the sum. ■

The oscillatory behavior of the Min-Sum algorithm on cycle graphs in the voltage problem marks a clear difference with what happens in the flow problem where, as we will see in Corollary 16 below, the algorithm *does* converge to the problem solution on cycle graphs.

Let us now consider the case $d \geq 3$. In this case, as Corollary 13 below attests, the convergence of the Min-Sum algorithm in the L-norm can be controlled by the total variation (TV) distance between the probability distributions of non-backtracking random walks that originate from neighbor nodes. In this framework, it is natural to also consider a modification of the L-norm normalized by the probability mass that is left out by the overlap of the above-mentioned random walks, which coincides with the TV distance between the walks. We give a general definition of this norm, and then present some of its properties.

Definition 11 (TV-normalized L-norm) *Given a row-stochastic matrix $M \in \mathbb{R}^{V \times V}$, define the pseudo-norm $\|\cdot\|_{L,M}$ as $\|\nu\|_{L,M}^2 := \sum_{\{v,w\} \in E} W_{vw} \big(\frac{(M\nu)_v - (M\nu)_w}{\|M_v - M_w\|_{TV}} \big)^2$, for any $\nu \in \mathbb{R}^V$, where $\|M_v - M_w\|_{TV} := \frac{1}{2} \sum_{z \in V} |M_{vz} - M_{wz}|$ is the TV distance between the probability mass functions defined by the v-th and w-th rows of the matrix M, respectively.*

The fact that $\|\cdot\|_{L,M}$ is a pseudo-norm follows from the fact that $\|\cdot\|_L$ itself is a pseudo-norm. Indeed, note that $\|\nu\|_{L,M} = \|M\nu\|_{L'}$, where L' is the Laplacian of the weighted graph (V, E, W'), with $W'_{vw} := W_{vw} / \|M_v - M_w\|_{TV}^2$. By a well-known result (see Proposition 4.3 in Levin et al. (2009), for instance), we have $\|M_v - M_w\|_{TV} = \sum_{z \in V : M_{vz} > M_{wz}} (M_{vz} - M_{wz}) = \sum_{z \in V : M_{vz} < M_{wz}} (M_{vz} - M_{wz})$, which shows how the TV-normalized L-norm takes into account the overlap of the distributions defined by the rows of the matrix M.

Proposition 12 *For any row-stochastic matrix $M \in \mathbb{R}^{V \times V}$ and $\nu \in \mathbb{R}^V$ we have:*

(i) $\|\nu\|_{L,I} = \|\nu\|_L$, *where I is the identity matrix;*

(ii) $\|M\nu\|_L \leq \max_{\{v,w\} \in E} \|M_v - M_w\|_{TV} \, \|\nu\|_{L,M}$;

(iii) $\|\nu\|_L \leq (\sum_{\{v,w\} \in E} W_{vw})^{1/2} \max_{\{v,w\} \in E} |\nu_v - \nu_w| \leq (\sum_{\{v,w\} \in E} W_{vw})^{1/2} \operatorname{osc}(\nu)$;

(iv) $\|\nu\|_{L,M} \leq (\sum_{\{v,w\} \in E} W_{vw})^{1/2} \operatorname{osc}(\nu)$;

where the oscillation of ν is defined as $\operatorname{osc}(\nu) := \max_{v,w \in V} |\nu_v - \nu_w|$.

Proof Properties $(i), (ii)$ and (iii) are straightforward to verify. Property (iv) follows from $|(M\nu)_v - (M\nu)_w| \leq \|M_v - M_w\|_{TV} \operatorname{osc}(\nu)$. See Georgii (2011), Chapter 8, for instance. ■

We now state a convergence result for the Min-Sum algorithm $\hat{\nu}^t$ for the voltage problem, in the case of d-regular graphs, $d \geq 3$, with the same edge weights. As the L-norm involves a sum over neighbour nodes, and as Lemma 9 shows that the error of Min-Sum is characterized by the distribution of a (non-overlapping) random walk, we immediately see that Min-Sum for the voltage problem does not converge on bipartite graphs as in this case random walks that start at two neighbor locations never overlap. For this reason, we also consider an averaged version of the Min-Sum algorithm where the output of two consecutive iterations

is properly averaged: $\hat{\nu}^t_{ave} := \frac{b_{d,t-1}}{b_{d,t-1}+b_{d,t}}\hat{\nu}^{t-1} + \frac{b_{d,t}}{b_{d,t-1}+b_{d,t}}\hat{\nu}^t$. Let $Q^{(t)} := (P^{(t-1)} + P^{(t)})/2$ and define

$$\gamma(t) := \max_{\{v,w\}\in E} \|P_v^{(t)} - P_w^{(t)}\|_{TV}, \qquad \gamma_{\rm ave}(t) := \max_{\{v,w\}\in E} \|Q_v^{(t)} - Q_w^{(t)}\|_{TV}.$$

Corollary 13 (Regular graphs, $d \geq 3$, equal weights) *Setting of Lemma 9. For $t \geq 4$,*

$$\|\nu^\star - \hat{\nu}^t\|_L \leq 2\,\gamma(t+1)\,\|\nu^\star\|_{L,P^{(t+1)}} \leq 2\,\gamma(t+1)\,\sqrt{m}\,\mathrm{osc}(\nu^\star),$$
$$\|\nu^\star - \hat{\nu}^t_{\rm ave}\|_L \leq 2\,\gamma_{\rm ave}(t+1)\,\|\nu^\star\|_{L,Q^{(t+1)}} \leq 2\,\gamma_{\rm ave}(t+1)\,\sqrt{m}\,\mathrm{osc}(\nu^\star).$$

Proof From Lemma 9, $\nu^\star - \hat{\nu}^t = P^{(t+1)}\nu^\star/b_{d,t}$. Using property (ii) in Proposition 12 with $M = P^{(t+1)}$ we get $\|\nu^\star - \hat{\nu}^t\|_L^2 = \|P^{(t+1)}\nu^\star\|_L^2/b_{d,t}^2 \leq 4\,\gamma(t+1)^2\,\|\nu^\star\|^2_{L,P^{(t+1)}}$, as $b_{d,t} \geq 1/2$ by Proposition 8. As for $t \geq 4$ Lemma 9 yields $(b_{d,t-1} + b_{d,t})(\nu^\star - \hat{\nu}^t_{\rm ave}) = b_{d,t-1}(\nu^\star - \hat{\nu}^{t-1}) + b_{d,t}(\nu^\star - \hat{\nu}^t) = 2Q^{(t+1)}\nu^\star$, we get $\|\nu^\star - \hat{\nu}^t_{\rm ave}\|_L^2 = 4\|Q^{(t+1)}\nu^\star\|_L^2/(b_{d,t-1} + b_{d,t})^2 \leq 4\,\gamma_{\rm ave}(t+1)^2\,\|\nu^\star\|^2_{L,Q^{(t+1)}}$. The proof is concluded using Property (iv) in Proposition 12. ■

Corollary 13 shows that the convergence in the L-norm of the Min-Sum estimate $\hat{\nu}^t$ and its averaged version $\hat{\nu}^t_{\rm ave}$ is controlled by the behavior with time of the quantities $\gamma(t+1)$ and $\gamma_{\rm ave}(t+1)$, respectively, which represent uniform bounds on the TV distance between probability distributions of non-backtracking random walks that make $t+1$ steps originating from neighbor nodes. Proposition 32 in Appendix B yields a recursive formula to compute the matrices $P^{(t)}$'s in terms of the adjacency matrix of the undirected unweighted graph (V, E). This formula can then be used to compute the quantities $\gamma(t)$ and $\gamma_{\rm ave}(t)$. Here we present numerical results for a few classes of d-regular graphs (for convenience of presentation, we include also the numerical results related to the quantities $\widetilde{\gamma}(t)$ and $\widetilde{\gamma}_{\rm ave}(t)$ that control the convergence behavior of the Min-Sum algorithm in the flow problem and that will be defined later on in Section 4.3).

Example 1 (Complete graphs) *A n-complete graph is a graph with n nodes where every pair of nodes is connected by an edge. A n-complete graph is d-regular with degree $d = n-1$. Figure 1 (left). The quantity $\gamma(t)$ is seen to decay (albeit not monotonically) like an exponential function $\alpha(d)\exp(-\beta(d)t)$, where $\alpha(d) > 0$ is a decreasing function of d and $\beta(d) > 0$ is an increasing function of d. Figure 1 (center).*

Example 2 (Connected cycles) *A $d/2$-connected cycle, for $d \geq 4$ even, is a cycle graph where each node is connected to its d nearest nodes. Figure 2 (left). The quantity $\gamma(t)$ is seen to decay like a polynomial function $\alpha(d)/\sqrt{t}$, where $\alpha(d) > 0$ is an increasing function of d. The decay of $\gamma(t)$ is independent[3] of n and m. Figure 2 (center).*

Example 3 (Tori) *A $d/2$-dimensional torus, for $d \geq 4$ even, is a $d/2$-dimensional grid with periodic boundary conditions. Figure 3 (left). The quantity $\gamma_{\rm ave}(t)$ is seen to decay like a polynomial function $\alpha(d)/\sqrt{t}$, where $\alpha(d) > 0$ is an increasing function of d. The decay of $\gamma_{\rm ave}(t)$ is independent of n and m (the same considerations as in Example 2 hold). Figure 3 (center).*

3. It is implied that the decay with time is independent of n and m as long as time t satisfies $t \lesssim D/d$, where D is the diameter of the graph. Once the random walks reach the periodic boundary conditions the decay is faster (exponential instead of polynomial) due to the extra overlaps of the walks.

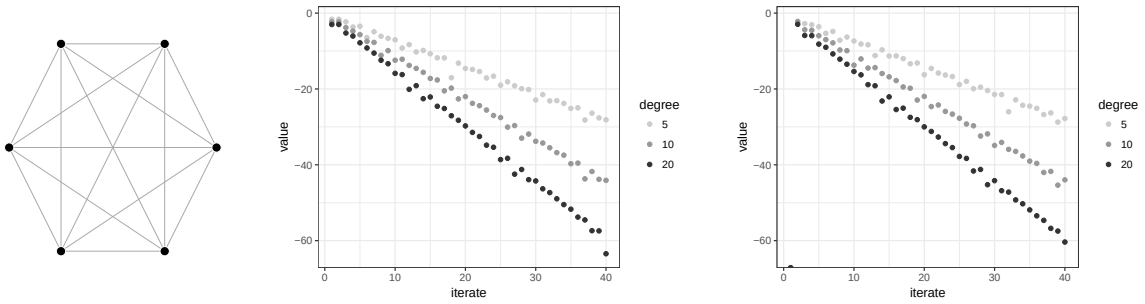

Figure 1: A 6-complete graph (left). Plot of $\log(\gamma(t))$ (center) and of $\log(\widetilde{\gamma}(t))$ (right) versus time t for various $(d+1)$-complete graphs.

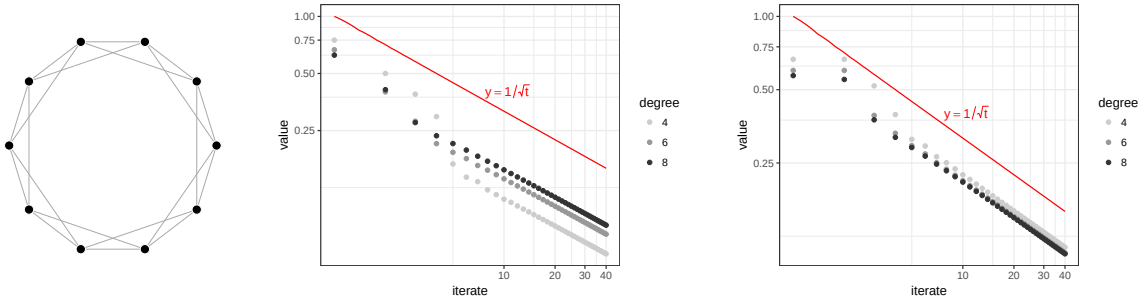

Figure 2: A 2-connected cycle (left). Plot in logarithmic scale of $\gamma(t)$ (center) and of $\widetilde{\gamma}(t)$ (right) versus time t for various $d/2$-connected cycles.

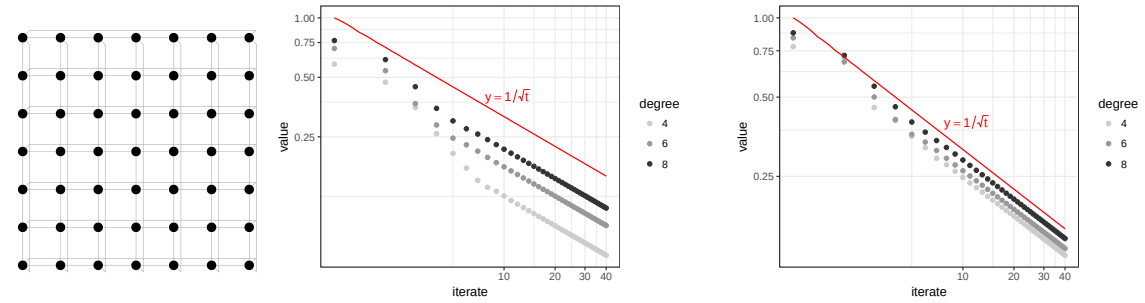

Figure 3: A 2-dimensional torus (left). Plot in logarithmic scale of $\gamma_{\text{ave}}(t)$ (center) and of $\widetilde{\gamma}_{\text{ave}}(t)$ (right) versus time t for various $d/2$-dimensional tori.

Remark 14 (Dobrushin's ergodicity coefficient) *For Markov chains with transition kernel P, there is a large literature investigating the contraction coefficient $\max_{v,w\in V} \|P_v^t - P_w^t\|_{TV}$ (a.k.a. Dobrushin's ergodicity coefficient), where t here represents the minimal time so that the coefficient is strictly less than 1. The Dobrushin's coefficient is fundamentally different from $\gamma(t)$. The former involves a worst-case bound over all nodes in the graph, while the latter involves a worst-case bound only over nearest-neighbors nodes. As a result,*

the Dobrushin's ergodicity coefficient typically depends on n and m, even when $\gamma(t)$ does not, such as in the examples above.

The fast solvers analyzed in the literature (see Introduction, Section 1, and Section 2) achieve a quasi-linear running time $\tilde{O}(m \log^c n \log 1/\varepsilon)$ to return an estimate $\hat{\nu}$ that satisfies $\|\nu^\star - \hat{\nu}\|_L \leq \varepsilon \|\nu^\star\|_L$. As per the results in Corollary 13, to compare the performance of the Min-Sum algorithm to these solvers we need to relate the TV-normalized L-norm to the ordinary L-norm. It is easy to construct examples where $\|\nu^\star\|_L < \|\nu^\star\|_{L,M}$ or where $\|\nu^\star\|_L > \|\nu^\star\|_{L,M}$, for a certain row-stochastic matrix M, while both norms satisfy $\|\nu^\star\|_L \leq \sqrt{m}\,\mathrm{osc}(\nu)$ (property (iii) in Proposition 12) and $\|\nu^\star\|_{L,M} \leq \sqrt{m}\,\mathrm{osc}(\nu)$ (property (iv) in Proposition 12). The following theorem summarizes the algorithmic complexity of the Min-Sum algorithm ν^t in the L-norm, under the assumption that the two norms of interest can be compared, and under the assumption that $\gamma(t)$ decays exponentially or polynomially, as attested numerically in the examples above (an analogous statement holds for the averaged Min-Sum algorithm ν^t_{ave}, replacing $P^{(t)}$ with $Q^{(t)}$, and $\gamma(t)$ with $\gamma_{\mathrm{ave}}(t)$).

Theorem 15 (Min-Sum voltage, running time, L-norm) *Setting of Lemma 9. Let $\alpha > 0, \beta > 0, \delta > 1$ independent of n, m. Assume that $\|\nu^\star\|_{L,P^{(t)}} \leq \delta \|\nu^\star\|_L$ for $t \geq 0$.*

1. *If $\gamma(t) \leq \alpha \exp(-\beta t)$ for $t \geq 0$, then the Min-Sum algorithm yields an estimate $\hat{\nu}$ such that $\|\nu^\star - \hat{\nu}\|_L \leq \varepsilon \|\nu^\star\|_L$ with running time $O(m \log 1/\varepsilon)$.*
2. *If $\gamma(t) \leq \alpha / t^\beta$ for $t \geq 0$, then the Min-Sum algorithm yields an estimate $\hat{\nu}$ such that $\|\nu^\star - \hat{\nu}\|_L \leq \varepsilon \|\nu^\star\|_L$ with running time $O(m/\varepsilon^{1/\beta})$.*

Proof The proof follows immediately from Corollary 13, recalling that each iteration of the Min-Sum algorithm requires a complexity that is linear in the number of edges, i.e., $O(m)$. If *1.* holds, then $\|\nu^\star - \hat{\nu}^t\|_L \leq \varepsilon \|\nu^\star\|_L$ is guaranteed by choosing t such that $2\alpha\delta \exp(-\beta(t+1)) \leq \varepsilon$; we can choose the smallest value $t = \lceil \log(2\alpha\delta/\varepsilon)/\beta - 1 \rceil \sim O(\log 1/\varepsilon)$. If *2.* holds, then $\|\nu^\star - \hat{\nu}^t\|_L \leq \varepsilon \|\nu^\star\|_L$ is guaranteed by choosing t such that $2\alpha\delta/(t+1)^\beta \leq \varepsilon$; we can choose the smallest value $t = \lceil (2\alpha\delta/\varepsilon)^{1/\beta} - 1 \rceil \sim O(1/\varepsilon^{1/\beta})$. ■

Theorem 15 shows that for certain classes of d-regular graphs, as characterized by the decaying properties of $\gamma(t)$, under the assumption that the TV-normalized L-norm of $\nu^\star$ is bounded by the L-norm of $\nu^\star$, the Min-Sum algorithm yields linear time solvers for the voltage problem, with respect to the L-norm.

4.3. Convergence Results for the Flow Problem

We now analyze the convergence behavior of the Min-Sum algorithm when applied to the flow problem: Algorithm 5. We present the results for the ℓ_∞ norm, defined as $\|x\|_\infty := \max_{e \in \vec{E}} x_e$ for any $x \in \mathbb{R}^{\vec{E}}$. The results for the ℓ_2 norm, defined as $\|x\|_2 := \sqrt{x^T x}$, are analogous to the results in the L-norm for the voltage case, as we describe in Remark 19.

For the case $d = 2$, Lemma 6 immediately yields the following result.

Corollary 16 (Regular graphs, $d = 2$) *Consider the setting of Lemma 6. Let $\omega_M := \max_{\{v,w\} \in E} W_{vw}$. Then, $\|x^\star - \hat{x}^t\|_\infty \leq \frac{2\omega_M}{2t+1} \|\nu^\star\|_\infty$. If ω_M is independent of n and m, then the Min-Sum algorithm yields $\hat{x}$ such that $\|x^\star - \hat{x}\|_\infty \leq \varepsilon \|\nu^\star\|_\infty$ with running time $O(m/\varepsilon)$.*

Proof From Lemma 6, by the triangle inequality, we have $|x^\star_e - \hat{x}^t_e| \leq \frac{\omega_M}{2t+1}(|\nu^\star_{\rho(v-t)}| + |\nu^\star_{\rho(v+t+1)}|)$, and the first bound follows by the triangle inequality for norms. The running time is immediate, as each iteration of the algorithm requires a complexity $O(m)$. ■

Corollary 16 shows that the flow Min-Sum algorithm on cycles converges in linear time to the problem solution, as opposed to the divergent behavior observed in the voltage case.

Let us now consider the case $d \geq 3$. As in the voltage problem, also in this case the convergence of the Min-Sum algorithm can be controlled by the TV distance between the probability distributions of properly-defined non-backtracking random walks that originate from neighbor nodes. As before, we also consider an averaged version of the Min-Sum algorithm where the output of two consecutive iterations is properly averaged: $\hat{x}^t_{\text{ave}} := \frac{c_{d,t-1}}{c_{d,t-1}+c_{d,t}}\hat{x}^{t-1} + \frac{c_{d,t}}{c_{d,t-1}+c_{d,t}}\hat{x}^t$. Analogously to the voltage case, let $Q^{(t,w)} := (P^{(t-1,w)} + P^{(t,w)})/2$ and

$$\widetilde{\gamma}(t) := \max_{\{v,w\}\in E} \|P^{(t,w)}_v - P^{(t,v)}_w\|_{TV}, \qquad \widetilde{\gamma}_{\text{ave}}(t) := \max_{\{v,w\}\in E} \|Q^{(t,w)}_v - Q^{(t,v)}_w\|_{TV}.$$

Corollary 17 (Regular graphs, $d \geq 3$, equal weights) *Setting of Lemma 9. For $t \geq 4$,*

$$\|x^\star - \hat{x}^t\|_\infty \leq 2\,\omega\,\frac{d-1}{d}\,\widetilde{\gamma}(t)\,\|\nu^\star\|_\infty, \qquad \|x^\star - \hat{x}^t_{\text{ave}}\|_\infty \leq 2\,\omega\,\frac{d-1}{d}\,\widetilde{\gamma}_{\text{ave}}(t)\,\|\nu^\star\|_\infty.$$

Proof For any $t \geq 1$, define the matrix $\widetilde{\Delta}^{(t)} \in \mathbb{R}^{\vec{E}\times V}$ as $\widetilde{\Delta}^{(t)}_{ez} := P^{(t,w)}_{vz} - P^{(t,v)}_{wz}, e = (v,w) \in \vec{E}$, $z \in V$. From Lemma 9 we have $x^\star - \hat{x}^t = \frac{\omega}{c_{d,t}}\frac{d-1}{d}\widetilde{\Delta}^{(t)}\nu^\star$, and the first bound follows from the sub-multiplicative property of the induced ℓ_∞ matrix norm, noticing that $\widetilde{\gamma}(t) = \|\widetilde{\Delta}^{(t)}\|_\infty/2$, and using that $c_{d,t} \geq 1$ from Proposition 8. The second bound follows analogously, as for any $t \geq 4$ we have $(c_{d,t-1}+c_{d,t})(x^\star - \hat{x}^t_{\text{ave}}) = c_{d,t-1}(x^\star - \hat{x}^{t-1}) + c_{d,t}(x^\star - \hat{x}^t) = \omega\,\frac{d-1}{d}(\widetilde{\Delta}^{(t-1)} + \widetilde{\Delta}^{(t)})\nu^\star$, and $\widetilde{\gamma}_{\text{ave}}(t) = \|(\widetilde{\Delta}^{(t-1)} + \widetilde{\Delta}^{(t)})/2\|_\infty/2$. ■

Corollary 17 shows that the convergence in the ℓ_∞ norm of the Min-Sum estimate $\hat{x}^t$ and its averaged version $\hat{x}^t_{\text{ave}}$ is controlled by the behavior with time of the quantities $\widetilde{\gamma}(t)$ and $\widetilde{\gamma}_{\text{ave}}(t)$, respectively, which represent uniform bounds on the TV distance between probability distributions of non-backtracking random walks that make t steps originating from neighbor nodes, conditioned on the event that the two walks avoid each others at the first step. Proposition 32 in Appendix B yields a recursive formula to compute the matrices $P^{(t,w)}$'s in terms of the adjacency matrix of the undirected unweighted graph (V, E). This formula can then be used to compute the quantities $\widetilde{\gamma}(t)$ and $\widetilde{\gamma}_{\text{ave}}(t)$. Numerical results for a few classes of d-regular graphs have been presented in the previous section. See Figure 1 (right)[4], Figure 2 (right), and Figure 3 (right). The following theorem summarizes the algorithmic complexity of the Min-Sum algorithm x^t in the ℓ_∞ norm, under the assumption that $\widetilde{\gamma}(t)$ decays exponentially or polynomially, as attested numerically in the examples above (an analogous statement holds for the output x^t_{ave}, replacing $\widetilde{\gamma}(t)$ with $\widetilde{\gamma}_{\text{ave}}(t)$).

Theorem 18 (Min-Sum flow, running time, ℓ_∞ norm) *Setting of Lemma 9. Let $\alpha > 0, \beta > 0$, and ω be independent of n and m.*

4. It is easy to see that $\widetilde{\gamma}(1) = 0$ for complete graphs, which explains the points at $t = 1$ in Figure 1 (right).

1. *If $\widetilde{\gamma}(t) \leq \alpha \exp(-\beta t)$ for $t \geq 0$, then the Min-Sum algorithm yields an estimate $\hat{x}$ such that $\|x^\star - \hat{x}\|_\infty \leq \varepsilon \|\nu^\star\|_\infty$ with running time $O(m \log 1/\varepsilon)$.*

2. *If $\widetilde{\gamma}(t) \leq \alpha/t^\beta$ for $t \geq 0$, then the Min-Sum algorithm yields an estimate $\hat{x}$ such that $\|x^\star - \hat{x}\|_\infty \leq \varepsilon \|\nu^\star\|_\infty$ with running time $O(m/\varepsilon^{1/\beta})$.*

Proof It follows from Corollary 17, analogously to the proof of Theorem 15. ■

Theorem 18 shows that for certain classes of d-regular graphs, as characterized by the decaying properties of $\widetilde{\gamma}(t)$, the Min-Sum algorithm yields linear time solvers for the flow problem, with respect to the ℓ_∞ norm. The following remark shows that an analogous result to Theorem 18 can be derived with respect to the ℓ_2 norm, with an additional assumption in the spirit of the one used in Theorem 15 for the voltage case.

Remark 19 (Analysis in ℓ_2 norm) *For d-regular graphs with $d \geq 3$, the analysis in the ℓ_2 norm for the flow problem follows the same steps as the analysis in the L-norm for the voltage problem. In fact, if one defines the matrices $\Delta^{(t)}$ and $\widetilde{\Delta}^{(t)} \in \mathbb{R}^{\vec{E} \times V}$ respectively as $\Delta^{(t)}_{ez} := P^{(t)}_{vz} - P^{(t)}_{wz}$ and $\widetilde{\Delta}^{(t)}_{ez} := P^{(t,w)}_{vz} - P^{(t,v)}_{wz}$, for any $e = (v,w) \in \vec{E}, z \in V$, then $\|x^\star - \hat{x}^t\|_2 = \frac{\omega}{c_{d,t}} \frac{d-1}{d} \|\widetilde{\Delta}^{(t)} \nu^\star\|_2$, while $\|\nu^\star - \hat{\nu}^t\|_L = \frac{1}{b_{d,t}} \|\Delta^{(t)} \nu^\star\|_2$. In analogy with Theorem 15, the results of Theorem 18 hold when replacing $\|x^\star - \hat{x}\|_\infty \leq \varepsilon \|\nu^\star\|_\infty$ with $\|x^\star - \hat{x}\|_2 \leq \varepsilon \|\nu^\star\|_L$, upon assuming that there exists $\delta > 1$ independent of n and m such that $\sum_{\{v,w\} \in E} W_{vw} \big(\frac{(P^{(t,w)}\nu)_v - (P^{(t,v)}\nu)_w}{\|P^{(t,w)}_v - P^{(t,v)}_w\|_{TV}} \big)^2 \leq \delta^2 \|\nu^\star\|^2_L$, for $t \geq 0$, which corresponds to the assumption $\|\nu^\star\|_{L,P^{(t)}} \leq \delta \|\nu^\star\|_L$ in Theorem 15.*

5. Results for General Weighted Graphs

This section develops the characterization of the error committed by the voltage and flow algorithms on general weighted graphs in terms of hitting times of ordinary diffusion random walks defined on the computation trees that are obtained by unraveling the operations of the algorithms with time (Theorem 23 and Lemma 24 for the flow case, and Theorem 28 for the voltage case). As applications, we specialize the general results here developed to prove the error characterization lemmas given in Section 4, i.e., Lemma 6 and Lemma 9.

While the main architecture of the results for the voltage problem (Algorithm 3) and the flow problem (Algorithm 5) is similar, there are some key differences that need to be taken into account. First, as the voltage algorithm updates functions (i.e., messages) on the vertices while the flow algorithm updates functions on the edges, the two algorithms give rise to different topologies for the computation tree. Second, as the voltage problem is an unconstrained optimization problem while the flow problem is an optimization problem with constraints, a different mechanism to find the fix point(s) of the algorithm is needed. As in the literature, to the best of our knowledge, previous analyses of the Min-Sum algorithm for quadratic and, more generally, convex problems only focused on the unconstrained case, we first present the full details of the results for the flow problem. Then, we give an overview of the results for the voltage problem, outlining in details only the parts where the analysis differs significantly from the one done for the flow problem.

5.1. Flow Problem, Algorithm 3

Our framework builds on the general scheme presented in Moallemi and Van Roy (2010) for unconstrained optimization problems, as we investigate the evolution of the Min-Sum algorithm under a set of linear perturbations. Henceforth, we consider Algorithm 5 when the initial messages are parametrized by $\{R^0_{e\to v} = R_{ee}\}, \{r^0_{e\to v} = p_{e\to v}\}$, for a certain set of real numbers $p = \{p_{e\to v}\}$. As far as the analysis is concerned, it is convenient (and more amenable to generalizations) to consider the form of Algorithm 4. For a given choice of perturbation parameters $p = \{p_{e\to v}\}$, define the initial messages as $\mu^0_{e\to v}(\,\cdot\,, p) : z \in \mathbb{R} \longrightarrow \mu^0_{e\to v}(z,p) = \frac{1}{2}R_{ee}z^2 + p_{e\to v}z$, and denote by $\{\mu^s_{e\to v}(\,\cdot\,,p)\}$, $s \geq 1$, the corresponding sequence of messages generated by the Min-Sum algorithm. Denote the estimates at time t as $\hat{x}^t(p)_e := \arg\min_{z\in\mathbb{R}} \mu^t_e(z,p)$, where $\mu^t_e(\,\cdot\,,p)$ is the corresponding belief function. We use the notation $\mathcal{N}(v) := \{w \in V : \{v,w\} \in E\}$ to denote the neighborhood of vertex v in G.

5.1.1. Computation Tree (Flow Problem)

To investigate the convergence behavior of the Min-Sum algorithm, we recall the concept of the computation tree, which is the graph obtained by unraveling the operations of the algorithm with time (Gallager, 1963; Weiss and Freeman, 2001; Tatikonda and Jordan, 2002). Given an edge $\tilde{e} = (\tilde{v}, \tilde{w}) \in \vec{E}$, the computation tree rooted at $\tilde{e}$ of depth t supports the optimization problem that is obtained by unfolding the computations involved in the Min-Sum estimate $\hat{x}^t(p)_{\tilde{e}}$. Formally, the computation tree is a directed tree $\vec{\mathbb{T}} = (\mathbb{V}, \vec{\mathbb{E}})$ (throughout, we use the double-struck notation to refer to quantities related to the computation tree) where each vertex in $\mathbb{V}$ is mapped to a vertex in V through a map $\sigma : \mathbb{V} \to V$ that preserves the edge structure, namely, if $\mathbb{e} = (\mathbb{v}, \mathbb{w}) \in \vec{\mathbb{E}}$ then $(\sigma(\mathbb{v}), \sigma(\mathbb{w})) \in \vec{E}$. Henceforth, we use the notation $\sigma(\mathbb{e}) := (\sigma(\mathbb{v}), \sigma(\mathbb{w}))$. The tree $\vec{\mathbb{T}}$ is defined iteratively. Initially, at level 0, the tree corresponds to a single root edge $\tilde{\mathbb{e}} = (\tilde{\mathbb{v}}, \tilde{\mathbb{w}}) \in \vec{\mathbb{E}}$ corresponding to $\tilde{e}$, i.e., $\sigma(\tilde{\mathbb{e}}) = \tilde{e}$. At this stage, $\tilde{\mathbb{v}}$ and $\tilde{\mathbb{w}}$ are the leaves of the tree. For the remaining t levels, the following procedure is repeated. The leaves in the tree are examined. Given a leaf $\mathbb{v}$ with $\sigma(\mathbb{v}) = v$ that is connected to a vertex $\mathbb{w}$ with $\sigma(\mathbb{w}) = w$, for any $z \in \mathcal{N}(v) \setminus w$, a vertex $\mathbb{z}$ with $\sigma(\mathbb{z}) = z$ and a directed edge $(\mathbb{v}, \mathbb{z})$ (resp. $(\mathbb{z}, \mathbb{v})$) are added to the next level of the tree if $(v,z) \in \vec{E}$ (resp. $(z,v) \in \vec{E}$). Figure 4 gives an example. We denote the set of vertices and edges in the k-th level of the tree respectively by $\mathbb{V}^k \subset \mathbb{V}$ and $\mathbb{E}^k \subset \mathbb{E}$. In what follows we also extend the usual neighborhood notation to vertices and edges in the graph $\mathbb{T}$, namely, $\partial\mathbb{v}$ is the set of edges in $\mathbb{T}$ that are connected to node $\mathbb{v}$, and $\partial\mathbb{e}$ is the set of vertices in $\mathbb{T}$ that are connected to edge $\mathbb{e}$.

The t-th (last) level of the tree $\vec{\mathbb{T}}$ is of particular relevance, as it supports the initial conditions of the Min-Sum algorithm, as we are now about to describe. Let $\mathbb{R} \in \mathbb{R}^{\vec{\mathbb{E}}\times\vec{\mathbb{E}}}$ be the diagonal matrix defined by $\mathbb{R}_{\mathbb{e}\mathbb{e}} := R_{\sigma(\mathbb{e})\sigma(\mathbb{e})}$. For a given choice of the perturbation parameters $p = \{p_{e\to v}\}$, define $\mathbb{p}(p) \in \mathbb{R}^{\vec{\mathbb{E}}}$ by $\mathbb{p}(p)_{\mathbb{e}} := p_{\sigma(\mathbb{e})\to\sigma(\mathbb{v})}$ if $\mathbb{e} \in \vec{\mathbb{E}}^t, \mathbb{v} = \partial\mathbb{e} \cap \mathbb{V}^{t-1}$, and equals to 0 otherwise. Let $\bar{\mathbb{V}} := \mathbb{V} \setminus \mathbb{V}^t$ (throughout, we use the bar notation to denote quantities related to a graph in which a node has been removed). Define the matrix $\bar{\mathbb{A}} \in \mathbb{R}^{\bar{\mathbb{V}}\times\vec{\mathbb{E}}}$ as $\bar{\mathbb{A}}_{\mathbb{v}\mathbb{e}} := A_{\sigma(\mathbb{v})\sigma(\mathbb{e})}$ if $\mathbb{e} \in \partial\mathbb{v}$ and $\bar{\mathbb{A}}_{\mathbb{v}\mathbb{e}} := 0$ otherwise. Define the vector $\bar{\mathbb{b}} \in \mathbb{R}^{\bar{\mathbb{V}}}$ as $\bar{\mathbb{b}}_{\mathbb{v}} := b_{\sigma(\mathbb{v})}$.

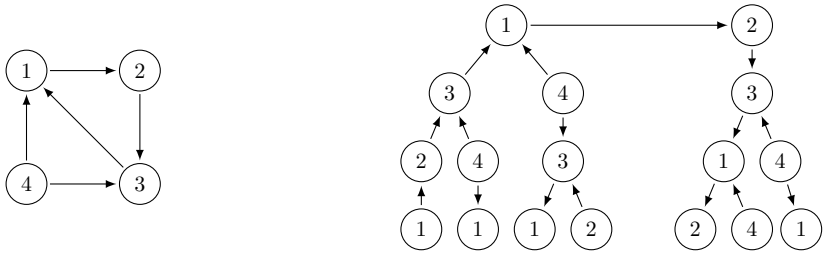

Figure 4: Graph $\vec{G} = (V = \{1, 2, 3, 4\}, \vec{E})$ (left) and computation tree $\vec{\mathbb{T}} = (\mathbb{V}, \vec{\mathbb{E}})$ of depth $t = 3$ with root edge corresponding to $(1, 2) \in \vec{E}$ (right). Vertices in $\vec{\mathbb{T}}$ are labeled by the corresponding vertices in $\vec{G}$.

Consider the following problem supported on the tree $\vec{\mathbb{T}}$, over $\mathbb{x} \in \mathbb{R}^{\vec{\mathbb{E}}}$:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\mathbb{x}^T \mathbb{R}\mathbb{x} + \mathbb{p}(p)^T \mathbb{x} \\ \text{subject to} \quad & \bar{\mathbb{A}}\mathbb{x} = \bar{\mathbb{b}}. \end{aligned} \tag{9}$$

Let $\mathbb{x}^\star(p)$ denote the unique optimal solution of this problem, as a function of the perturbation p. The relationship between the Min-Sum algorithm and the computation tree is made explicit by the following lemma, which can be easily established by inductively examining the operations of Algorithm 4.

Lemma 20 *Given the setting of this section,* $\hat{x}^t(p)_{\tilde{e}} = \mathbb{x}^\star(p)_{\tilde{\mathbb{e}}}$.

We now introduce a few quantities related to the computation tree $\vec{\mathbb{T}}$ and the problem (9) that will play a key role in our analysis. The signed vertex-edge incidence matrix $\mathbb{A} \in \mathbb{R}^{\mathbb{V} \times \vec{\mathbb{E}}}$ associated to the directed graph $\vec{\mathbb{T}}$ is defined for each $\mathbb{v} \in \mathbb{V}$, $\mathbb{e} \in \vec{\mathbb{E}}$ as $\mathbb{A}_{\mathbb{v}\mathbb{e}} := 1$ if edge $\mathbb{e}$ leaves node $\mathbb{v}$, $\mathbb{A}_{\mathbb{v}\mathbb{e}} := -1$ if edge $\mathbb{e}$ enters node $\mathbb{v}$, and $\mathbb{A}_{\mathbb{v}\mathbb{e}} := 0$ otherwise. It is easy to verify that $\bar{\mathbb{A}}$ is the submatrix of $\mathbb{A}$ that corresponds to the rows associated to $\bar{\mathbb{V}}$. Let $\mathbb{W} \in \mathbb{R}^{\mathbb{V} \times \mathbb{V}}$ be a symmetric matrix that assigns a positive weight to every edge in $\vec{\mathbb{E}}$, defined as $\mathbb{W}_{\mathbb{v}\mathbb{w}} := W_{\sigma(\mathbb{v})\sigma(\mathbb{w})}$ if either $(\mathbb{v}, \mathbb{w}) \in \vec{\mathbb{E}}$ or $(\mathbb{w}, \mathbb{v}) \in \vec{\mathbb{E}}$, and $\mathbb{W}_{\mathbb{v}\mathbb{w}} := 0$ otherwise. Henceforth, we think of $\vec{\mathbb{T}} = (\mathbb{V}, \vec{\mathbb{E}}, \mathbb{W})$ as the corresponding weighted graph, and $\mathbb{T} = (\mathbb{V}, \mathbb{E}, \mathbb{W})$ as the undirected graph naturally associated to $\vec{\mathbb{T}}$, where the edge set $\mathbb{E}$ is obtained by removing the orientation from the edges in $\vec{\mathbb{E}}$. As $\mathbb{T}$ is connected by assumption, it is easy to show that the range of $\mathbb{A}$ is spanned by the vectors orthogonal to the all-ones vector. It follows that the matrix $\bar{\mathbb{A}}$ has full row rank. Define the weighted degree of a vertex $\mathbb{v}$ by $\mathbb{d}_{\mathbb{v}} := \sum_{\mathbb{w} \in \mathbb{V}} \mathbb{W}_{\mathbb{v}\mathbb{w}}$, and let $\mathbb{D} \in \mathbb{R}^{\mathbb{V} \times \mathbb{V}}$ be the diagonal matrix defined by $\mathbb{D}_{\mathbb{v}\mathbb{v}} := \mathbb{d}_{\mathbb{v}}$. The Laplacian $\mathbb{L}$ of the graph $\vec{\mathbb{T}}$ is the matrix defined as $\mathbb{L} := \mathbb{D} - \mathbb{W}$. Let $\bar{\mathbb{L}} \in \mathbb{R}^{\bar{\mathbb{V}} \times \bar{\mathbb{V}}}$ denote the submatrix of $\mathbb{L}$ that corresponds to the rows and columns associated to $\bar{\mathbb{V}}$. Note that $\mathbb{R}_{\mathbb{e}\mathbb{e}} := 1/\mathbb{W}_{\mathbb{v}\mathbb{w}}$. It is easy to verify that the following equalities holds: $\mathbb{L} = \mathbb{A}\mathbb{R}^{-1}\mathbb{A}^T$ and $\bar{\mathbb{L}} = \bar{\mathbb{A}}\mathbb{R}^{-1}\bar{\mathbb{A}}^T$. We also define the transition matrix of the ordinary diffusion random walk on $\mathbb{T}$ as the matrix $\mathbb{P} := \mathbb{D}^{-1}\mathbb{W}$.

5.1.2. Fix Point (Flow Problem)

Clearly, the choice $p = 0$ for the linear perturbation parameters yields the original (unperturbed) algorithm. The next theorem below shows that for a particular choice of p the

Min-Sum algorithm yields the optimal solution at any time $t \geq 1$. The optimal choice of p is a function of the optimal Lagrangian multipliers for problem (3). Recall from (4) that the Lagrangian of this optimization problem is given by $\mathcal{L}(x,\nu) := \frac{1}{2}x^T Rx + \nu^T(b - Ax)$, where $\nu = (\nu_v)_{v\in V}$ is the vector formed by the Lagrangian multipliers. Define the function Ψ from $\mathbb{R}^{\vec{E}} \times \mathbb{R}^V$ to $\mathbb{R}^{\vec{E}} \times \mathbb{R}^V$ as $\Psi(x,\nu) := (\nabla_x \mathcal{L}(x,\nu), Ax - b) = (Rx - A^T\nu, Ax - b)$. Recall that $x^\star$ denotes the unique optimal solution of problem (3). As the constraints are linear, by the Lagrange multiplier theorem there exists $\nu^\star \in \mathbb{R}^V$ so that $\Psi(x^\star, \nu^\star) = 0$, i.e.,

$$\begin{cases} R_{ee}x^\star_e - (A^T\nu^\star)_e = 0 \text{ for } e \in \vec{E}, \\ \sum_{e\in\vec{E}} A_{ve}x^\star_e = b_v \text{ for } v \in V. \end{cases} \tag{10}$$

Note that the first line in (10) represents Ohm's law, and the second line represents Kirchhoff's law. Given the pair $(x^\star, \nu^\star)$, we are now ready to define the set of parameters $p^\star = \{p^\star_{e\to v}\}$ so that the Min-Sum algorithm yields the optimal solution at any time step.

Theorem 21 *For each $e \in \vec{E}$, $v \in \partial e$, let $w = \partial e \setminus v$ and define $p^\star_{e\to v} := -A_{we}\nu^\star_w$. Then, for any $\tilde{e} \in \vec{E}$, $t \geq 1$, we have $\hat{x}^t(p^\star)_{\tilde{e}} = x^\star_{\tilde{e}}$.*

Proof Fix $\tilde{e} \in \vec{E}$, $t \geq 1$. Consider the setting in Section 5.1.1, and let $\vec{\mathbb{T}}$ be the computation tree of depth t rooted at $\tilde{e}$. The Lagrangian of the problem (9) corresponding to the choice $p = p^\star$ is the function $\mathbb{L}$ from $\mathbb{R}^{\vec{\mathbb{E}}} \times \mathbb{R}^{\bar{\mathbb{V}}}$ to $\mathbb{R}$ defined by $\mathbb{L}(\mathbb{x}, \mathbb{v}) := \frac{1}{2}\mathbb{x}^T\mathbb{R}\mathbb{x} + \mathbb{p}(p^\star)^T\mathbb{x} + \mathbb{v}^T(\bar{\mathbb{b}} - \bar{\mathbb{A}}\mathbb{x})$, where $\mathbb{v} = (\mathbb{v}_\mathbb{v})_{\mathbb{v}\in\bar{\mathbb{V}}}$. Let Ψ from $\mathbb{R}^{\vec{\mathbb{E}}} \times \mathbb{R}^{\bar{\mathbb{V}}}$ to $\mathbb{R}^{\vec{\mathbb{E}}} \times \mathbb{R}^{\bar{\mathbb{V}}}$ be defined as $\Psi(\mathbb{x}, \mathbb{v}) := (\nabla_\mathbb{x}\mathbb{L}(\mathbb{x}, \mathbb{v}), \bar{\mathbb{A}}\mathbb{x} - \bar{\mathbb{b}}) = (\mathbb{R}\mathbb{x} + \mathbb{p}(p^\star) - \bar{\mathbb{A}}^T\mathbb{v}, \bar{\mathbb{A}}\mathbb{x} - \bar{\mathbb{b}})$. The Lagrangian multiplier theorem says for the unique minimizer $\mathbb{x}^\star \equiv \mathbb{x}^\star(p^\star) \in \mathbb{R}^{\vec{\mathbb{E}}}$ of (9) there exists $\mathbb{v}^\star \in \mathbb{R}^{\bar{\mathbb{V}}}$ so that $\Psi(\mathbb{x}^\star, \mathbb{v}^\star) = 0$, namely,

$$\begin{cases} \mathbb{R}_{\mathbb{e}\mathbb{e}}\mathbb{x}^\star_\mathbb{e} - (\bar{\mathbb{A}}^T\mathbb{v}^\star)_\mathbb{e} = 0, \mathbb{e} \in \vec{\mathbb{E}} \setminus \vec{\mathbb{E}}^t, \\ \mathbb{R}_{\mathbb{e}\mathbb{e}}\mathbb{x}^\star_\mathbb{e} + p^\star_{\sigma(\mathbb{e})\to\sigma(\mathbb{v})} - (\bar{\mathbb{A}}^T\mathbb{v}^\star)_\mathbb{e} = 0, \mathbb{e} \in \vec{\mathbb{E}}^t, \mathbb{v} = \partial\mathbb{e} \cap \mathbb{V}^{t-1}, \\ \sum_{\mathbb{e}\in\vec{\mathbb{E}}} \bar{\mathbb{A}}_{\mathbb{v}\mathbb{e}}\mathbb{x}^\star_\mathbb{e} = \bar{\mathbb{b}}_\mathbb{v}, \mathbb{v} \in \bar{\mathbb{V}}. \end{cases}$$

Using (10) it is easy to check that the choice $\mathbb{x}^\star_\mathbb{e} = x^\star_{\sigma(\mathbb{e})}$ and $\mathbb{v}^\star_\mathbb{v} = \nu^\star_{\sigma(\mathbb{v})}$ satisfies the above system of equations. The proof is concluded by Lemma 20. ■

5.1.3. Error Characterization (Flow Problem)

The next lemma characterizes the sensitivity of the $\tilde{\mathbb{e}}$-th component (root edge of $\vec{\mathbb{T}}$) of the optimal solution of the optimization problem (9) with respect to perturbations of the parameters $p_{e\to v}$'s, which are supported on $\vec{\mathbb{E}}^t$.

Lemma 22 *Consider the setting in Section 5.1.1. Fix $\tilde{e} = (\tilde{v}, \tilde{w}) \in \vec{E}$, $t \geq 1$, and let $\vec{\mathbb{T}}$ be the computation tree of depth t rooted at $\tilde{e}$, with root edge $\tilde{\mathbb{e}} = (\tilde{\mathbb{v}}, \tilde{\mathbb{w}}) \in \vec{\mathbb{E}}$. For any $p = \{p_{e\to v}\}$, $e \in \vec{E}$ and $v \in \partial e$, we have*

$$\frac{\partial}{\partial p_{e\to v}}\mathbb{x}^\star(p)_{\tilde{\mathbb{e}}} = \frac{1}{R_{\tilde{e}\tilde{e}}} \sum_{\mathbb{v}\in\mathbb{V}^{t-1}} (\bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{v}}\mathbb{v}} - \bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{w}}\mathbb{v}})(\mathbb{y}_{e\to v})_\mathbb{v},$$

where $\mathbb{y}_{e\to v} \in \mathbb{R}^{\mathbb{V}^{t-1}}$ is defined, for each $\mathbb{v} \in \mathbb{V}^{t-1}$, as $(\mathbb{y}_{e\to v})_\mathbb{v} := \frac{A_{ve}}{R_{ee}}\mathbf{1}_{\sigma(\mathbb{v})=v}\mathbf{1}_{\sigma(\partial\mathbb{v}\cap\vec{\mathbb{E}}^t)=e}$, where $\mathbf{1}_S$ is the indicator function defined as $\mathbf{1}_S = 1$ if statement S is true, $\mathbf{1}_S = 0$ otherwise.

Proof By Proposition 31 in Appendix B, the solution of problem (9) reads $\mathbb{x}^\star(p) = \mathbb{R}^{-1}\bar{\mathbb{A}}^T\bar{\mathbb{L}}^{-1}\bar{\mathbb{b}} + (\mathbb{R}^{-1}\bar{\mathbb{A}}^T\bar{\mathbb{L}}^{-1}\bar{\mathbb{A}} - I)\mathbb{R}^{-1}\mathbb{p}(p)$, where $I \in \mathbb{R}^{\vec{\mathbb{E}}\times\vec{\mathbb{E}}}$ is the identity matrix. As $\mathbb{p}(p)$ is supported on the edges of $\mathbb{T}$ that are connected to its leaves and $\tilde{\mathfrak{e}}$ is the root edge, we have $\frac{\partial}{\partial p_{e\to v}}\mathbb{x}^\star(p)_{\tilde{\mathfrak{e}}} = \big(\mathbb{R}^{-1}\bar{\mathbb{A}}^T\bar{\mathbb{L}}^{-1}\bar{\mathbb{A}}\mathbb{R}^{-1}\frac{\partial}{\partial p_{e\to v}}\mathbb{p}(p)\big)_{\tilde{\mathfrak{e}}} = \frac{1}{R_{\tilde{e}\tilde{e}}}\sum_{\mathbb{v}\in\mathbb{V}^{t-1}}(\bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{v}}\mathbb{v}} - \bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{w}}\mathbb{v}})(\mathbb{y}_{e\to v})_{\mathbb{v}}$, where $(\mathbb{y}_{e\to v})_{\mathbb{v}} = (\bar{\mathbb{A}}\mathbb{R}^{-1}\frac{\partial}{\partial p_{e\to v}}\mathbb{p}(p))_{\mathbb{v}} = \frac{A_{ve}}{R_{ee}}\mathbf{1}_{\sigma(\mathbb{v})=v}\mathbf{1}_{\sigma(\partial\mathbb{v}\cap\vec{\mathbb{E}}^t)=e}$ for each $\mathbb{v}\in\mathbb{V}^{t-1}$. ■

As a consequence of Lemma 22, we immediately have the following characterization of the error committed by Algorithm 5 with initial conditions $\{R^0_{e\to v} = R_{ee}\}$, $\{r^0_{e\to v} = 0\}$.

Theorem 23 *Consider the setting in Section 5.1.1. Fix $\tilde{e} = (\tilde{v}, \tilde{w}) \in \vec{E}$, $t \geq 1$, and let $\vec{\mathbb{T}}$ be the computation tree of depth t rooted at $\tilde{e}$, with root edge $\tilde{\mathfrak{e}} = (\tilde{\mathbb{v}}, \tilde{\mathbb{w}}) \in \vec{\mathbb{E}}$. We have*

$$x^\star_{\tilde{e}} - \hat{x}^t_{\tilde{e}} = \mathbb{W}_{\tilde{\mathbb{v}}\tilde{\mathbb{w}}}\sum_{\mathbb{v}\in\mathbb{V}^{t-1}}(\bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{v}}\mathbb{v}} - \bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{w}}\mathbb{v}})\sum_{\mathbb{w}\in\mathbb{V}^t}\mathbb{W}_{\mathbb{v}\mathbb{w}}\nu^\star_{\sigma(\mathbb{w})}.$$

Proof From Theorem 21 and Lemma 22, by the fundamental theorem of calculus and the chain rule of differentiation, we get $x^\star_{\tilde{e}} - \hat{x}^t_{\tilde{e}} = \hat{x}^t(p^\star)_{\tilde{e}} - \hat{x}^t(0)_{\tilde{e}} = \int_0^1 d\theta\, \frac{d}{d\theta}\mathbb{x}^\star(\theta p^\star)_{\tilde{\mathfrak{e}}} = \int_0^1 d\theta \sum_{e\in\vec{E}, v\in\partial e}\frac{\partial}{\partial p_{e\to v}}\mathbb{x}^\star(\theta p^\star)_{\tilde{\mathfrak{e}}}\, p^\star_{e\to v} = \mathbb{W}_{\tilde{\mathbb{v}}\tilde{\mathbb{w}}}\sum_{\mathbb{v}\in\mathbb{V}^{t-1}}(\bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{v}}\mathbb{v}} - \bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{w}}\mathbb{v}})\sum_{e\in\vec{E}, v\in\partial e}(\mathbb{y}_{e\to v})_{\mathbb{v}}\, p^\star_{e\to v}$. By Theorem 21 $p^\star_{e\to v} = -A_{we}\nu^\star_w$, $w = \partial e \setminus v$, and $\sum_{\substack{e\in\vec{E}\\ v\in\partial e}}(\mathbb{y}_{e\to v})_{\mathbb{v}}\, p^\star_{e\to v} = \sum_{\mathbb{w}\in\mathbb{V}^t}\mathbb{W}_{\mathbb{v}\mathbb{w}}\nu^\star_{\sigma(\mathbb{w})}$. ■

We now use the error characterization provided by Theorem 23 and the general connection between restricted Laplacians and random walks developed in Appendix A to prove the results in Section 4. Henceforth, for each $\mathbb{v}\in\mathbb{V}$, let $\mathbf{P}_{\mathbb{v}}$ be the law of a time homogeneous Markov chain $X_0, X_1, X_2, \ldots$ on $\mathbb{V}$ with transition matrix $\mathbb{P} := \mathbb{D}^{-1}\mathbb{W}$ and initial condition $X_0 = \mathbb{v}$. Analogously, denote by $\mathbf{E}_{\mathbb{v}}$ the expectation with respect to this law. The hitting time to a set $\mathbb{Z}\subseteq\mathbb{V}$ is defined as $T_{\mathbb{Z}} := \inf\{k\geq 0 : X_k\in\mathbb{Z}\}$. We also extend the notation $\mathcal{N}(\mathbb{v}) := \{\mathbb{w}\in\mathbb{V} : \{\mathbb{v},\mathbb{w}\}\in\mathbb{E}\}$ to denote the vertex-neighborhood of vertex $\mathbb{v}$ in $\mathbb{T}$.

The next lemma, which builds on the results developed in Appendix A, gives an expression for the restricted Laplacian in terms of hitting probabilities of random walks. This result represents the computational device that we will use repeatedly in what follows.

Lemma 24 *Consider the setting in Section 5.1.1. Fix $\tilde{e} = (\tilde{v}, \tilde{w}) \in \vec{E}$, $t\geq 1$, and let $\vec{\mathbb{T}}$ be the computation tree of depth t rooted at $\tilde{e}$, with root edge $\tilde{\mathfrak{e}} = (\tilde{\mathbb{v}}, \tilde{\mathbb{w}}) \in \vec{\mathbb{E}}$. For each $\mathbb{v}\in\bar{\mathbb{V}}$, $\mathbb{w}\in\mathbb{V}^{t-1}$, with $\mathbb{z} = \mathcal{N}(\mathbb{w})\setminus\mathbb{V}^t$, we have*

$$\bar{\mathbb{L}}^{-1}_{\mathbb{v}\mathbb{w}} = \frac{\mathbf{P}_{\mathbb{v}}(T_{\mathbb{w}} < T_{\mathbb{V}^t})}{[1 - \mathbf{P}_{\mathbb{z}}(T_{\mathbb{w}} < T_{\mathbb{V}^t})\,\mathbf{P}_{\mathbb{w}}(X_1 = \mathbb{z})]\,\mathbb{d}_{\mathbb{w}}}.$$

Proof Fix $\mathbb{v}\in\bar{\mathbb{V}}$, $\mathbb{w}\in\mathbb{V}^{t-1}$, and let $\mathbb{z} = \mathcal{N}(\mathbb{w})\setminus\mathbb{V}^t$. Define the random variable $A := \sum_{k=0}^{T_{\mathbb{V}^t}}\mathbf{1}_{X_k=\mathbb{w}}$. From Proposition 30 we have $\bar{\mathbb{L}}^{-1}_{\mathbb{v}\mathbb{w}} = \frac{1}{\mathbb{d}_{\mathbb{w}}}\mathbf{E}_{\mathbb{w}}[A]\,\mathbf{P}_{\mathbb{v}}(T_{\mathbb{w}} < T_{\mathbb{V}^t})$. By the Markov property, performing a first step analysis for the random walk, we get $\mathbf{E}_{\mathbb{w}}[A] = 1 + \sum_{\tilde{\mathbb{z}}\in\mathcal{N}(\mathbb{w})}\mathbf{E}_{\tilde{\mathbb{z}}}[A]\mathbf{P}_{\mathbb{w}}(X_1 = \tilde{\mathbb{z}}) = 1 + \mathbf{E}_{\mathbb{z}}[A]\mathbf{P}_{\mathbb{w}}(X_1 = \mathbb{z})$, where we used that $\mathbf{E}_{\tilde{\mathbb{z}}}[A] = 0$ for any $\tilde{\mathbb{z}}\in\mathbb{V}^t$. Conditioning on the event $\{T_{\mathbb{w}} < T_{\mathbb{V}^t}\}$ and its complementary, using that $\mathbf{E}_{\mathbb{z}}[A|T_{\mathbb{w}} > T_{\mathbb{V}^t}] = 0$, we also get $\mathbf{E}_{\mathbb{z}}[A] = \mathbf{E}_{\mathbb{z}}[A|T_{\mathbb{w}} < T_{\mathbb{V}^t}]\,\mathbf{P}_{\mathbb{z}}(T_{\mathbb{w}} < T_{\mathbb{V}^t})$. By the strong Markov property $\mathbf{E}_{\mathbb{z}}[A|T_{\mathbb{w}} < T_{\mathbb{V}^t}] = \mathbf{E}_{\mathbb{w}}[A]$, so that $\mathbf{E}_{\mathbb{w}}[A] = 1 + \mathbf{E}_{\mathbb{w}}[A]\,\mathbf{P}_{\mathbb{z}}(T_{\mathbb{w}} <$

$T_{\mathbb{V}^t})\,\mathbf{P}_{\mathbb{w}}(X_1 = \mathbb{z})$ and $\mathbf{E}_{\mathbb{w}}[A] = 1/(1 - \mathbf{P}_{\mathbb{z}}(T_{\mathbb{w}} < T_{\mathbb{V}^t})\,\mathbf{P}_{\mathbb{w}}(X_1 = \mathbb{z}))$. ■

5.1.4. Proofs of Results for d-Regular Graphs (Flow Problem)

We can now present the proofs of the results concerning the flow problem contained in Lemma 6 and Lemma 9 on the characterization of the error committed by the Min-Sum algorithm as a function of time and as a function of the voltage solution $\nu^\star = L^+ b$.

Proof [Proof of Lemma 6, flow problem] Let $\vec{G} = (V = \{0, \dots, n-1\}, \vec{E}, W)$ be a weighted directed cycle. Consider the setting in Section 5.1.1. Fix $v \in V$ and assume that $\tilde{e} = (\tilde{v} = \rho(v), \tilde{w} = \rho(v+1)) \in \vec{E}$ (the case $e = (\tilde{w} = \rho(v+1), \tilde{v} = \rho(v))$ yields to the same result with a minus sign in front). Fix $t \geq 2$, and let $\vec{\mathbb{T}}$ be the computation tree of depth t rooted at $\tilde{e}$, with root edge $\tilde{\mathbb{e}} = (\tilde{\mathbb{v}}, \tilde{\mathbb{w}}) \in \vec{\mathbb{E}}$. Label the tree vertices as $\mathbb{V} = \{0, 1, \dots, 2t+1\}$, as in Figure 5. Clearly, $\tilde{\mathbb{v}} = t, \tilde{\mathbb{w}} = t+1$, $\mathbb{V}^{t-1} = \{1, 2t\}$, and $\mathbb{V}^t = \{0, 2t+1\}$.

Figure 5: Cycle $G = (V, E)$ (left) and computation tree $\mathbb{T} = (\mathbb{V}, \mathbb{E})$ of depth t with root edge $\{\tilde{\mathbb{v}} = t, \tilde{\mathbb{w}} = t+1\} \in \mathbb{E}$ (right).

We will apply Lemma 24 to compute the quantities $\bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{v}}\mathbb{v}} - \bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{w}}\mathbb{v}}$, for each $\mathbb{v} \in \mathbb{V}^{t-1}$. We write the computations for $\mathbb{v} = 1$, as the case $\mathbb{v} = 2t$ follows immediately by symmetry. For convenience of notation, we write $\mathbb{W}_s$ to indicate $\mathbb{W}_{s,s+1}$. For each $s \in \mathbb{V}$, define $f_s := \mathbf{P}_s(T_1 < T_{\mathbb{V}^t})$, $p_s := \mathbf{P}_s(X_1 = s+1) = \mathbb{W}_s/\mathbb{d}_s$, $\mathbb{d}_s := \mathbb{W}_{s-1} + \mathbb{W}_s$, and $q_s := \mathbf{P}_s(X_1 = s-1) = 1 - p_s$. By a first step analysis of the random walk and the Markov property, we have

$$f_s = f_{s-1} q_s + f_{s+1} p_s, \quad s \in \{2, \dots, 2t\}, \tag{11}$$

with boundary conditions $f_1 = 1, f_{2t+1} = 0$. The above system of equations corresponds to the classical Gambler Ruin's problem, with final states 1 and $2t+1$, see Figure 6.

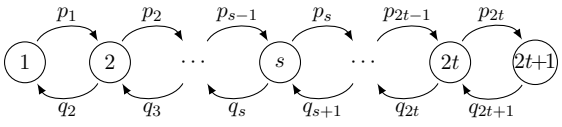

Figure 6: Gambler's Ruin.

To solve these equations, using that $1 = p_s + q_s$, we first rewrite them as $f_{s+1} - f_s = \frac{q_s}{p_s}(f_s - f_{s-1})$, for $s \in \{2, \dots, 2t\}$. Using that $q_s/p_s = \mathbb{W}_{s-1}/\mathbb{W}_s$ we get $f_{s+1} - f_s = \frac{\mathbb{W}_1}{\mathbb{W}_s}(f_2 - f_1)$. Summing these equations side by side, using that $f_1 = 1$, we get $f_s =$

$1+(f_2-1)\mathbb{W}_1(1/\mathbb{W}_1+1/\mathbb{W}_2+\cdots+1/\mathbb{W}_{s-1})$. Using that $f_{2t+1}=0$ we can solve for f_2 and finally obtain $f_s=\frac{1/\mathbb{W}_s+1/\mathbb{W}_{s+1}+\cdots+1/\mathbb{W}_{2t}}{1/\mathbb{W}_1+1/\mathbb{W}_2+\cdots+1/\mathbb{W}_{2t}}$ for any $s\in\{1,\dots,2t\}$. From Lemma 24, noticing that $\mathbb{z}=2$ in this case, we have $\bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{v}}1}-\bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{w}}1}=(f_t-f_{t+1})/((1-f_2p_1)\mathbb{d}_1)$, which equals $\bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{v}}1}-\bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{w}}1}=\frac{1/\mathbb{W}_t}{\mathbb{W}_0(1/\mathbb{W}_0+\cdots+1/\mathbb{W}_{2t})}$. By symmetry, we have $\bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{v}},2t}-\bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{w}},2t}=\frac{-1/\mathbb{W}_t}{\mathbb{W}_{2t}(1/\mathbb{W}_0+\cdots+1/\mathbb{W}_{2t})}$. By Theorem 23, as $\mathbb{W}_s=W_{\sigma(s),\sigma(s+1)}$, we finally get $x^\star_{\tilde{e}}-\hat{x}^t_{\tilde{e}}=\frac{\nu^\star_{\sigma(0)}-\nu^\star_{\sigma(2t+1)}}{1/W_{\sigma(0)\sigma(1)}+\cdots+1/W_{\sigma(2t)\sigma(2t+1)}}$. The proof follows as the map $\sigma:\mathbb{V}\to V$ reads $\sigma(s)=\rho(s-t+v)$. ■

Proof [Proof of Lemma 9, flow problem] Let $\vec{G}=(V,\vec{E},W)$ be a connected d-regular graph, with $d\geq 3$, where each edge has the same weight ω. Consider the setting in Section 5.1.1. Fix $\tilde{e}=(\tilde{v},\tilde{w})\in\vec{E}$, $t\geq 3$, and let $\vec{\mathbb{T}}$ be the computation tree of depth t rooted at $\tilde{e}$, with root edge $\tilde{\mathbb{e}}=(\tilde{\mathbb{v}},\tilde{\mathbb{w}})\in\vec{\mathbb{E}}$. Fix $\mathbb{v}\in\mathbb{V}^{t-1}$, and assume that $\mathbb{v}$ is contained in the subtree with root vertex $\tilde{\mathbb{v}}$. For what follows, it is convenient to label some of the vertices in $\mathbb{V}$. Let $\mathbb{v}=0,1=\mathcal{N}(\mathbb{v})\setminus\mathbb{V}^t,2=\mathcal{N}(1)\setminus\mathbb{V}^{t-1},\dots,k+1=\mathcal{N}(k)\setminus\mathbb{V}^{t-k}$, until we reach $\tilde{\mathbb{v}}=t-1$. Label $\tilde{\mathbb{w}}=t$. See Figure 7. We will apply Lemma 24 to compute the quantity $\bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{v}}\mathbb{v}}-\bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{w}}\mathbb{v}}$. To this

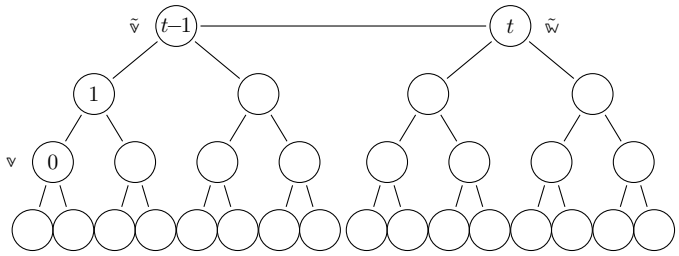

Figure 7: Computation tree $\mathbb{T}=(\mathbb{V},\mathbb{E})$ of depth $t=3$ associated to any 3-regular graph.

end, we need to compute the hitting probabilities $f_{\mathbb{w}}:=\mathbf{P}_{\mathbb{w}}(T_{\mathbb{v}}<T_{\mathbb{V}^t})$ for $\mathbb{w}\in\{1,t-1,t\}$. To do this, it is convenient to first reduce the network topology by invoking the equivalence between random walks and electrical circuits. See Peter G. Doyle (1984). The electrical circuit associated to the random walk X on $\mathbb{T}$ is constructed by assigning to each edge $\mathbb{e}=\{\mathbb{v},\mathbb{w}\}\in\mathbb{T}$ a conductor ω, which corresponds to a resistor $1/\omega$. Then, the quantities $f_{\mathbb{w}}$, $\mathbb{w}\in\mathbb{V}$, correspond to the voltages induced on $\mathbb{V}\setminus\{\mathbb{v}\cup\mathbb{V}^t\}$ when a voltage of intensity 1 is applied to $\mathbb{v}$ ($f_{\mathbb{v}}=1$), and a voltage of intensity 0 is applied to the sites in $\mathbb{V}^t$ ($f_{\mathbb{w}}=0$, $\mathbb{w}\in\mathbb{V}^t$). By repeatedly applying the standard network reduction operations for conductors in parallel and in series, it is easy to verify that, as long as the quantities $f_1,f_2,\dots,f_t$ are concerned, the original network is equivalent to the one represented in Figure 8.

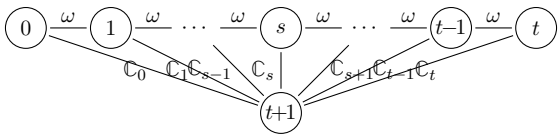

Figure 8: Reduced network.

In the reduced network, the conductance between the pairs of nodes s and $s+1$, for $s\in\{0,\dots,t-1\}$, remain ω as in the unreduced network, i.e., the computation tree. The

effect of all the nodes different from $0,\dots,t$ in the original graph is summarized in the reduced network by node $t+1$. For any $s \in \{0,\dots,t\}$, between node s and node $t+1$ there is a new conductor $\mathbb{C}_s$, which is obtained by reducing the subtree having root vertex s in the original graph. It is easy to verify that

$$\mathbb{C}_0 = \omega(d-1), \qquad \mathbb{C}_s = \omega\frac{(d-2)^2}{d-1}(1+h_{s+1}), s \in \{1,\dots,t-1\}, \qquad \mathbb{C}_t = \omega(d-2)(1+h_t),$$

with $h_s := \frac{1}{(d-1)^s-1}$. Let $p_0 := \frac{\omega}{\omega+\mathbb{C}_0}, q_0 := 1-p_0$, $p_s = q_s := \frac{\omega}{2\omega+\mathbb{C}_s}$ for $s \in \{1,\dots,t-1\}$, $p_t := \frac{\mathbb{C}_t}{\omega+\mathbb{C}_t}, q_t := 1-p_t$. By the Markov property we have

$$f_s = f_{s-1}q_s + f_{s+1}p_s, \quad s \in \{1,\dots,t\}, \tag{12}$$

with $f_0 = 1, f_{t+1} = 0$. This system of equations corresponds to a modification of the classical Gambler Ruin's problem, where states 0 and $t+1$ are the final states, and each intermediate state s has a probability $r_s := 1-p_s-q_s$ to reach state $t+1$. See Figure 9.

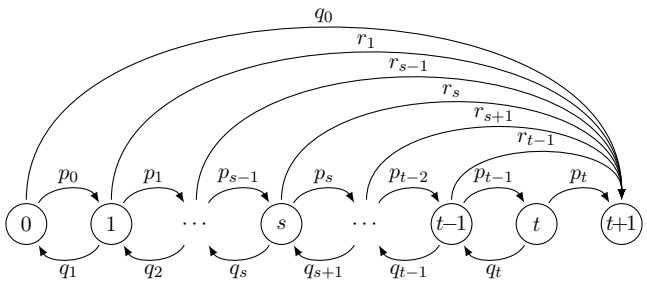

Figure 9: Gambler's Ruin on the reduced network.

Without loss of generality, we set $r_t = 0$. We can easily write the solution of this system of equations in a compact form. Define $\alpha_1 = \alpha_2 := 1, \alpha_s := \alpha_{s-1} - p_{s-2}q_{s-1}\alpha_{s-2}$ for $s \geq 3$. Define $\beta_1 = \beta_2 := 1$, $\beta_s := \beta_{s-1} - p_{s-1}q_s\beta_{s-2}$ for $s \geq 3$. Then, as a function of f_1, we can write the solution of (12) as $f_s = \frac{1}{p_1\cdots p_{s-1}}(\alpha_s f_1 - \beta_{s-1}q_1)$, for $s \in \{2,\dots,t+1\}$. In fact, it is easy to verify that this expression satisfies system (12). Using $f_{t+1} = 0$ we find $f_1 = q_1\beta_t/\alpha_{t+1}$ and $f_s = \frac{p_0q_1}{p_0\cdots p_{s-1}}(\frac{\alpha_s\beta_t}{\alpha_{t+1}} - \beta_{s-1})$ for $s \in \{1,\dots,t+1\}$, with $\beta_0 := 0$. Lemma 24 yields the following, using that $\mathbb{d}_{\mathrm{v}} = \omega d$,

$$\bar{\mathbb{L}}^{-1}_{\tilde{\mathrm{v}}\mathrm{v}} = \frac{f_{t-1}}{(1-f_1p_0)\mathbb{d}_{\mathrm{v}}} = \frac{1}{\omega d}\frac{q_1}{p_1\cdots p_{t-2}(\alpha_{t+1}-p_0q_1\beta_t)}(\alpha_{t-1}\beta_t - \alpha_{t+1}\beta_{t-2}),$$
$$\bar{\mathbb{L}}^{-1}_{\tilde{\mathrm{v}}\mathrm{v}} = \frac{f_t}{(1-f_1p_0)\mathbb{d}_{\mathrm{v}}} = \frac{1}{\omega d}\frac{q_1}{p_1\cdots p_{t-2}(\alpha_{t+1}-p_0q_1\beta_t)}\frac{1}{p_{t-1}}(\alpha_t\beta_t - \alpha_{t+1}\beta_{t-1}).$$

Using the recursive formulas for α_{t+1} and β_t, and the fact that $p_t + q_t = 1$, we get $\bar{\mathbb{L}}^{-1}_{\tilde{\mathrm{v}}\mathrm{v}} - \bar{\mathbb{L}}^{-1}_{\tilde{\mathrm{w}}\mathrm{v}} = \frac{1}{\omega d}\frac{q_1}{p_1\cdots p_{t-2}(\alpha_{t+1}-p_0q_1\beta_t)}p_t(\alpha_{t-1}\beta_{t-1} - \alpha_t\beta_{t-2})$, which, as $\alpha_s\beta_s - \alpha_{s+1}\beta_{s-1} = p_{s-1}q_s(\alpha_{s-1}\beta_{s-1} - \alpha_s\beta_{s-2})$, becomes

$$\bar{\mathbb{L}}^{-1}_{\tilde{\mathrm{v}}\mathrm{v}} - \bar{\mathbb{L}}^{-1}_{\tilde{\mathrm{w}}\mathrm{v}} = \frac{p_t}{q_t}\bar{\mathbb{L}}^{-1}_{\tilde{\mathrm{w}}\mathrm{v}} = \frac{1}{\omega d(d-1)^{t-1}}\frac{1}{\xi_{t-1}}, \tag{13}$$

where $\xi_{t-1} := \frac{1}{(d-1)^{t-1}} \frac{\alpha_{t+1} - p_0 q_1 \beta_t}{q_1 \cdots q_{t-1} p_t}$ is defined by $\xi_s = \frac{p_s}{q_s p_{s+1}(d-1)} \xi_{s-1} - \frac{p_{s-1} p_s q_{s+1}}{q_{s-1} q_s p_{s+1} (d-1)^2} \xi_{s-2}$ for $s \in \{2, \dots, t-1\}$, with $\xi_0 := \frac{p_0 q_1}{p_1}$ and $\xi_1 = \frac{1}{d-1} \frac{1 - p_0 q_1 - p_1 q_2}{q_1 p_2}$. Plugging in the definitions of the transition probabilities, recalling the quantities defined in Proposition 8, it is immediate to verify that $\delta_s \equiv \xi_s$ for $s \in \{0, \dots, t-2\}$ and $c_{d,t} \equiv \xi_{t-1}$. By symmetry, it is clear that $\bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{v}}\mathbb{v}} - \bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{w}}\mathbb{v}} = \frac{1}{\omega d (d-1)^{t-1}} \frac{1}{c_{d,t}}$ holds for *any* $\mathbb{v} \in \mathbb{V}^{t-1}$ that is contained in the subtree of $\mathbb{T}$ with root vertex $\tilde{\mathbb{v}}$. On the other hand, again by symmetry, for any $\mathbb{v} \in \mathbb{V}^{t-1}$ contained in the subtree of $\mathbb{T}$ with root vertex $\tilde{\mathbb{w}}$ we have $\bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{v}}\mathbb{v}} - \bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{w}}\mathbb{v}} = -\frac{1}{\omega d (d-1)^{t-1}} \frac{1}{c_{d,t}}$. Theorem 23 yields

$$x^\star_{\tilde{e}} - \hat{x}^t_{\tilde{e}} = \omega^2 \sum_{\mathbb{v} \in \mathbb{V}^{t-1}} (\bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{v}}\mathbb{v}} - \bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{w}}\mathbb{v}}) \sum_{\mathbb{w} \in \mathbb{V}^t} \nu^\star_{\sigma(\mathbb{w})} = \frac{\omega}{d(d-1)^{t-1}} \frac{1}{c_{d,t}} \sum_{v \in V} \nu^\star_v (\#^t_{\tilde{v} \to v, \tilde{w}} - \#^t_{\tilde{w} \to v, \tilde{v}}),$$

where $\#^t_{\tilde{v} \to v, \tilde{w}}$ is the number of non-backtracking paths in the original undirected graph $G = (V, E)$ that connect node $\tilde{v}$ to node v in t time steps, where the first step is different from $\tilde{w}$. As the total number of non-backtracking paths after t steps is $d(d-1)^{t-1}$, we have $\vec{\mathbf{P}}_{\tilde{v}}(Y_1 \neq \tilde{w}, Y_t = v) = \#^t_{\tilde{v} \to v, \tilde{w}} / (d(d-1)^{t-1})$. The proof follows as $\vec{\mathbf{P}}_{\tilde{v}}(Y_1 \neq \tilde{w}, Y_t = v) = \vec{\mathbf{P}}_{\tilde{v}}(Y_t = v | Y_1 \neq \tilde{w}) \vec{\mathbf{P}}_{\tilde{v}}(Y_1 \neq \tilde{w})$, and $\vec{\mathbf{P}}_{\tilde{v}}(Y_1 \neq \tilde{w}) = (d-1)/d$. ∎

5.2. Voltage Problem, Algorithm 5

We now give an overview of the results in the voltage case. The main framework for these results is the same as the one used in the previous section to investigate the behavior of the Min-Sum algorithm applied to the flow problem. For this reason, as outlined at the beginning of Section 5, here we only discuss in details the parts of the proofs that present major differences compared to the flow problem.

Henceforth we consider Algorithm 3 when the initial messages are parametrized by $\{W^0_{e \to v} = W_{vw}, e = \{v, w\}\}, \{h^0_{e \to v} = p_{e \to v}\}$, for a certain set of real numbers $p = \{p_{e \to v}\}$. Also in this case, as far as the analysis is concerned, it is convenient to consider the form of Algorithm 2. For a given choice of perturbation parameters $p = \{p_{e \to v}\}$, define the initial messages as $\mu^0_{e \to v}(\,\cdot\,, p) : z \in \mathbb{R} \longrightarrow \mu^0_{e \to v}(z, p) = \frac{1}{2} W_{vw} z^2 + p_{e \to v} z$, where $e = \{v, w\}$, and denote by $\{\mu^s_{e \to v}(\,\cdot\,, p)\}$, $s \geq 1$, the corresponding sequence of messages generated by the Min-Sum algorithm. Denote the estimates at time t as $\hat{\nu}^t_v = \arg\min_{z \in \mathbb{R}} \mu^t_v(z, p)$, where $\mu^t_v(\,\cdot\,, p)$ is the corresponding belief function.

5.2.1. Computation Tree (Voltage Problem)

The computation tree for the voltage problem differs from the one in the flow problem due to the fact that the Min-Sum algorithms for the voltage problem propagates messages that are supported on vertices, not on edges. For this reason, the computation tree that we will now define is rooted at a vertex, not at an edge as the one defined in Section 5.1.1.

Given a vertex $\tilde{v} \in V$, the computation tree rooted at $\tilde{v}$ of depth t supports the optimization problem that is obtained by unfolding the computations involved in the Min-Sum estimate $\hat{\nu}^t(p)_{\tilde{v}}$. Formally, the computation tree is an undirected tree $\mathbb{T} = (\mathbb{V}, \mathbb{E})$ where each vertex in $\mathbb{V}$ is mapped to a vertex in V through a map $\sigma : \mathbb{V} \to V$ that preserves the edge structure, namely, if $\mathbb{e} = \{\mathbb{v}, \mathbb{w}\} \in \mathbb{E}$ then $\sigma(\mathbb{e}) := \{\sigma(\mathbb{v}), \sigma(\mathbb{w})\} \in E$. The tree $\mathbb{T}$

is defined iteratively. Initially, at level -1, the tree is made by a single root vertex $\tilde{\mathbb{v}} \in \mathbb{V}$ corresponding to $\tilde{v}$, i.e., $\sigma(\tilde{\mathbb{v}}) = \tilde{v}$. Consecutively, for any $z \in \mathcal{N}(\tilde{v})$, a vertex $\mathbb{z}$ with $\sigma(\mathbb{z}) = z$ and an edge $\{\tilde{\mathbb{v}}, \mathbb{z}\}$ are added to the level 0 of the tree. For the remaining $t-1$ levels, the following procedure is repeated. The leaves in the tree are examined. Given a leaf $\mathbb{v}$ with $\sigma(\mathbb{v}) = v$ that is connected to a vertex $\mathbb{w}$ with $\sigma(\mathbb{w}) = w$, for any $z \in \mathcal{N}(v) \setminus w$, a vertex $\mathbb{z}$ with $\sigma(\mathbb{z}) = z$ and an edge $\{\mathbb{v}, \mathbb{z}\}$ are added to the next level of the tree. Figure 10 gives an example. We denote the set of vertices and edges in the k-th level of the tree respectively by $\mathbb{V}^k \subset \mathbb{V}$ and $\mathbb{E}^k \subset \mathbb{E}$. In what follows we also extend the usual neighborhood notation to vertices and edges in the graph $\mathbb{T}$, namely, $\partial\mathbb{v}$ is the set of edges in $\mathbb{T}$ that are connected to node $\mathbb{v}$, and $\partial\mathbb{e}$ is the set of vertices in $\mathbb{T}$ that are connected to edge $\mathbb{e}$.

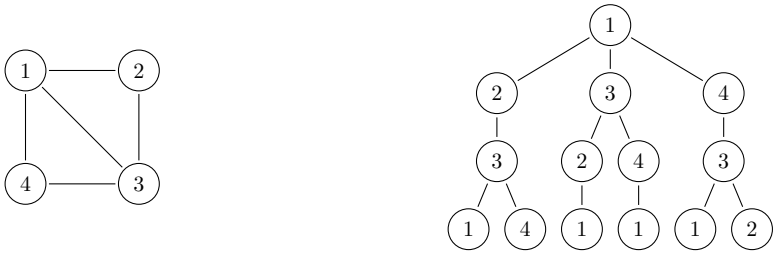

Figure 10: Graph $G = (V = \{1, 2, 3, 4\}, E)$ (left) and computation tree $\mathbb{T} = (\mathbb{V}, \mathbb{E})$ of depth $t = 2$ with root edge corresponding to $1 \in V$ (right). Vertices in $\mathbb{T}$ are labeled by the corresponding vertices in G.

The t-th (last) level of the tree is of particular relevance as it supports the initial conditions of the Min-Sum algorithm, as we are now about to describe. Let $\mathbb{W} \in \mathbb{R}^{\mathbb{V}\times\mathbb{V}}$ be the diagonal matrix defined by $\mathbb{W}_{\mathbb{v}\mathbb{w}} := W_{\sigma(\mathbb{v})\sigma(\mathbb{w})}$ if $\{\mathbb{v}, \mathbb{w}\} \in \mathbb{E}$ and $\mathbb{W}_{\mathbb{v}\mathbb{w}} := 0$ otherwise. Let $\bar{\mathbb{V}} := \mathbb{V} \setminus \mathbb{V}^t$. Define the weighted degree of a vertex $\mathbb{v}$ by $\mathbb{d}_{\mathbb{v}} := \sum_{\mathbb{w}\in\mathbb{V}} \mathbb{W}_{\mathbb{v}\mathbb{w}}$, and let $\mathbb{D} \in \mathbb{R}^{\mathbb{V}\times\mathbb{V}}$ be the diagonal matrix defined by $\mathbb{D}_{\mathbb{v}\mathbb{v}} := \mathbb{d}_{\mathbb{v}}$. The Laplacian $\mathbb{L}$ of the graph $\mathbb{T}$ is the matrix defined as $\mathbb{L} := \mathbb{D} - \mathbb{W}$. Let $\bar{\mathbb{L}} \in \mathbb{R}^{\bar{\mathbb{V}}\times\bar{\mathbb{V}}}$ denote the submatrix of $\mathbb{L}$ that corresponds to the rows and columns associated to $\bar{\mathbb{V}}$. For a given choice of the perturbation parameters $p = \{p_{e\to v}\}$, define $\bar{\mathbb{p}}(p) \in \mathbb{R}^{\bar{\mathbb{V}}}$ by

$$\bar{\mathbb{p}}(p)_{\mathbb{v}} := \begin{cases} \sum_{\mathbb{f}\in\partial\mathbb{v}\setminus\mathbb{e}} p_{\sigma(\mathbb{f})\to\sigma(\mathbb{v})} & \text{if } \mathbb{v} \in \mathbb{V}^{t-1}, \mathbb{e} = \partial\mathbb{v} \setminus \mathbb{E}^t, \\ 0 & \text{otherwise,} \end{cases} \tag{14}$$

and let $\bar{\mathbb{b}} \in \mathbb{R}^{\bar{\mathbb{V}}}$ defined by $\bar{\mathbb{b}}_{\mathbb{v}} := b_{\sigma(\mathbb{v})}$ for each $\mathbb{v} \in \bar{\mathbb{V}}$.

Consider the following problem supported on the computation tree $\mathbb{T}$, over $\bar{\mathbb{v}} \in \mathbb{R}^{\bar{\mathbb{V}}}$:

$$\text{minimize} \quad \frac{1}{2}\bar{\mathbb{v}}^T\bar{\mathbb{L}}\bar{\mathbb{v}} + (\bar{\mathbb{p}}(p) - \bar{\mathbb{b}})^T\bar{\mathbb{v}}. \tag{15}$$

While the Laplacian $\mathbb{L}$ is positive semi-definite, the restricted Laplacian $\bar{\mathbb{L}}$ is positive definite (Spielman, 2010). Hence, for any choice of the perturbation p the objective function in problem (15) is strictly convex and it admits a unique solution $\bar{\mathbb{v}}^\star(p)$. The relationship between the Min-Sum algorithm and the tree is made explicit by the following lemma, which can be established by inductively examining the operations of Algorithm 2.

Lemma 25 *Given the setting of this section,* $\hat{\nu}^t(p)_{\tilde{v}} = \bar{\mathbb{v}}^\star(p)_{\tilde{\mathbb{v}}}$.

5.2.2. Fix Point (Voltage Problem)

The choice $p = 0$ for the linear perturbation parameters yields the original (unperturbed) algorithm. The next theorem below shows that for a particular choice of p the Min-Sum algorithm yields the optimal solution at any time $t \geq 1$. The optimal choice of p is a function of the optimal solution $\nu^\star$ for problem (2). Recall that, by definition, $\nu^\star$ is the unique vector that satisfy $L\nu^\star = b$ and $1^T\nu^\star = 0$. Hence, component-wise, for each $v \in V$,

$$d_v\nu_v^\star - \sum_{w\in\partial v} W_{vw}\nu_w^\star - b_v = 0, \tag{16}$$

where recall that $d_v := \sum_{w\in\partial v} W_{vw}$. Given $\nu^\star$, we are now ready to define the set of parameters $p^\star = \{p^\star_{e\to v}\}$ so that the Min-Sum algorithm yields the optimal solution at any time step.

Theorem 26 *For each $e \in E$, $v \in \partial e$, let $w = \partial e \setminus v$ and define $p^\star_{e\to v} := -W_{vw}\nu^\star_w$. Then, for any $\tilde{v} \in V$, $t \geq 1$, we have $\hat{\nu}^t(p^\star)_{\tilde{v}} = \nu^\star_{\tilde{v}}$.*

Proof Fix $\tilde{v} \in V$, $t \geq 1$. Consider the setting in Section 5.2.1, and let $\mathbb{T}$ be the computation tree of depth t rooted at $\tilde{v}$. From the first order optimality conditions for problem (15) corresponding to the choice $p = p^\star$, we know that $\bar{\nu}^\star \equiv \bar{\nu}^\star(p^\star)$ is the unique vector so that

$$\begin{cases} \mathbb{d}_{\mathbb{v}}\bar{\nu}^\star_{\mathbb{v}} - \sum_{\mathbb{w}\in\partial\mathbb{v}} \mathbb{W}_{\mathbb{v}\mathbb{w}}\bar{\nu}^\star_{\mathbb{w}} - \bar{\mathbb{b}}_{\mathbb{v}} = 0, \mathbb{v} \in \bar{\mathbb{V}} \setminus \mathbb{V}^{t-1}, \\ \mathbb{d}_{\mathbb{v}}\bar{\nu}^\star_{\mathbb{v}} - \mathbb{W}_{\mathbb{v}\mathbb{w}}\bar{\nu}^\star_{\mathbb{w}} + \bar{\mathbb{p}}(p^\star)_{\mathbb{v}} - \bar{\mathbb{b}}_{\mathbb{v}} = 0, \mathbb{v} \in \mathbb{V}^{t-1}, \{\mathbb{v},\mathbb{w}\} = \partial\mathbb{v} \setminus \mathbb{E}^t. \end{cases}$$

Using (16) it is easy to check that the choice $\bar{\nu}^\star_{\mathbb{v}} = \nu^\star_{\sigma(\mathbb{v})}$ satisfies the above system of equations. The proof is concluded by Lemma 25. ■

5.2.3. Error Characterization (Voltage Problem)

The next lemma characterizes the sensitivity of the $\tilde{\mathbb{v}}$-th component (root vertex of $\mathbb{T}$) of the optimal solution of the optimization problem (15) with respect to perturbations of the parameters $p_{e\to v}$'s, which are supported on $\mathbb{V}^{t-1}$.

Lemma 27 *Consider the setting in Section 5.2.1. Fix $\tilde{v} \in V$, $t \geq 1$, and let $\mathbb{T}$ be the computation tree of depth t rooted at $\tilde{v}$, with root edge $\tilde{\mathbb{v}}$. For any $p = \{p_{e\to v}\}$, $e \in E$ and $v \in \partial e$, we have*

$$\frac{\partial}{\partial p_{e\to v}}\bar{\nu}^\star(p)_{\tilde{\mathbb{v}}} = -\sum_{\mathbb{v}\in\mathbb{V}^{t-1}} \bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{v}}\mathbb{v}}(\mathbb{y}_{e\to v})_{\mathbb{v}},$$

where $\mathbb{y}_{e\to v} \in \mathbb{R}^{\mathbb{V}^{t-1}}$ is defined, for each $\mathbb{v} \in \mathbb{V}^{t-1}$, as $(\mathbb{y}_{e\to v})_{\mathbb{v}} := \mathbf{1}_{\sigma(\mathbb{v})=v}\mathbf{1}_{\sigma(\partial\mathbb{v}\cap\vec{\mathbb{E}}^t)=e}$.

Proof From the first order optimality conditions, we know that the unique solution of problem (15) reads $\bar{\nu}^\star(p) = \bar{\mathbb{L}}^{-1}(\bar{\mathbb{b}} - \bar{\mathbb{p}}(p))$ so that $\frac{\partial}{\partial p_{e\to v}}\bar{\nu}^\star(p) = -\bar{\mathbb{L}}^{-1}\frac{\partial}{\partial p_{e\to v}}\bar{\mathbb{p}}(p)$. Using (14) it is easy to check that $(\mathbb{y}_{e\to v})_{\mathbb{v}} = (\frac{\partial}{\partial p_{e\to v}}\bar{\mathbb{p}}(p))_{\mathbb{v}}$. ■

As a consequence of Lemma 27, we immediately have the following characterization of the error committed by Algorithm 3 with $\{W^0_{e\to v} = W_{vw}, e = \{v,w\}\}, \{h^0_{e\to v} = 0\}$.

Theorem 28 *Consider the setting in Section 5.2.1. Fix $\tilde{v} \in V$, $t \geq 1$, and let $\mathbb{T}$ be the computation tree of depth t rooted at $\tilde{v}$, with root vertex $\tilde{\mathbb{v}} \in \mathbb{V}$. We have*

$$\nu^\star_{\tilde{v}} - \hat{\nu}^t_{\tilde{v}} = \sum_{\mathbb{v} \in \mathbb{V}^{t-1}} \bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{v}}\mathbb{v}} \sum_{\mathbb{w} \in \mathbb{V}^t} \mathbb{W}_{\mathbb{v}\mathbb{w}} \nu^\star_{\sigma(\mathbb{w})}.$$

Proof From Theorem 26 and Lemma 27, $\nu^\star_{\tilde{v}} - \hat{\nu}^t_{\tilde{v}} = \hat{\nu}^t(p^\star)_{\tilde{v}} - \hat{\nu}^t(0)_{\tilde{v}} = \int_0^1 d\theta \, \frac{d}{d\theta} \bar{\mathbb{v}}^\star(\theta p^\star)_{\tilde{\mathbb{v}}} = \int_0^1 d\theta \sum_{e \in E, v \in \partial e} \frac{\partial}{\partial p_{e \to v}} \bar{\mathbb{v}}^\star(\theta p^\star)_{\tilde{\mathbb{v}}} \, p^\star_{e \to v} = -\sum_{\mathbb{v} \in \mathbb{V}^{t-1}} \bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{v}}\mathbb{v}} \sum_{e \in E, v \in \partial e} (\mathbb{y}_{e \to v})_{\mathbb{v}} \, p^\star_{e \to v}$, by the fundamental theorem of calculus and the chain rule of differentiation. From Theorem 26 we get $p^\star_{e \to v} := -W_{vw} \nu^\star_w$ for $e = \{v, w\}$, and $\sum_{e \in E, v \in \partial e} (\mathbb{y}_{e \to v})_{\mathbb{v}} \, p^\star_{e \to v} = -\sum_{\mathbb{w} \in \mathbb{V}^t} \mathbb{W}_{\mathbb{v}\mathbb{w}} \nu^\star_{\sigma(\mathbb{w})}$. ■

Lemma 24 for the flow case holds also in the present case, upon referring the notation $\mathbf{P}_{\mathbb{v}}$, $\mathbf{E}_{\mathbb{v}}$, etc. to the Markov chain defined on the computation tree for the voltage case.

5.2.4. Proofs of Results for d-Regular Graphs (Voltage Problem)

We can now present the outline of the proofs for the results concerting the voltage problem in Lemma 6 and Lemma 9 in Section 4. Here we assume that the reader is already familiar with the proofs for the flow case.

Proof [Proof of Lemma 6, voltage problem] Let $G = (V = \{0, \dots, n-1\}, E, W)$ be a weighted cycle. Consider the setting in Section 5.2.1. Fix $v \in V$, $t \geq 2$, and let $\mathbb{T}$ be the computation tree of depth t rooted at $\tilde{v}$, with root vertex $\tilde{\mathbb{v}}$, i.e., $\sigma(\tilde{\mathbb{v}}) = \tilde{v}$. Label the vertices of the computation tree as $\mathbb{V} = \{0, 1, \dots, 2t+2\}$, as in Figure 11. Clearly, $\tilde{\mathbb{v}} = t+1$, $\mathbb{V}^{t-1} = \{1, 2t+1\}$, and $\mathbb{V}^t = \{0, 2t+2\}$.

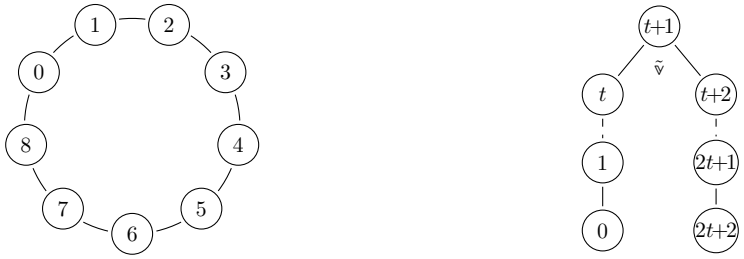

Figure 11: Ring $G = (V, E)$ (left) and computation tree $\mathbb{T} = (\mathbb{V}, \mathbb{E})$ of depth t with root vertex $\tilde{\mathbb{v}} = t + 1 \in \mathbb{V}$ (right).

Given this set up for the computation tree, the remaining of the proof follows exactly the steps of the proof in the flow case. Using the analogue of Lemma 24 for the voltage case, we can compute the quantities $\bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{v}}\mathbb{v}}$, for each $\mathbb{v} \in \mathbb{V}^{t-1}$, by solving a system of equations like (11). If we write $\mathbb{W}_s$ to indicate $\mathbb{W}_{s,s+1}$, we find that $\bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{v}}1} = \frac{1/\mathbb{W}_{t+1} + \cdots + 1/\mathbb{W}_{2t+1}}{\mathbb{W}_0 (1/\mathbb{W}_0 + \cdots + 1/\mathbb{W}_{2t+1})}$, and, by symmetry, $\bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{v}},2t+1} = \frac{1/\mathbb{W}_0 + \cdots + 1/\mathbb{W}_t}{\mathbb{W}_{2t+1} (1/\mathbb{W}_0 + \cdots + 1/\mathbb{W}_{2t+1})}$. By Theorem 28 we get $\nu^\star_{\tilde{v}} - \hat{\nu}^t_{\tilde{v}} = \sum_{\mathbb{v} \in \mathbb{V}^{t-1}} \bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{v}}\mathbb{v}} \sum_{\mathbb{w} \in \mathbb{V}^t} \mathbb{W}_{\mathbb{v}\mathbb{w}} \nu^\star_{\sigma(\mathbb{w})} = \frac{1/\mathbb{W}_{t+1} + \cdots + 1/\mathbb{W}_{2t+1}}{1/\mathbb{W}_0 + \cdots + 1/\mathbb{W}_{2t+1}} \nu^\star_{\sigma(0)} + \frac{1/\mathbb{W}_0 + \cdots + 1/\mathbb{W}_t}{1/\mathbb{W}_0 + \cdots + 1/\mathbb{W}_{2t+1}} \nu^\star_{\sigma(2t+2)}$, and the proof follows using $\mathbb{W}_s = W_{\sigma(s), \sigma(s+1)}$ and that $\sigma : \mathbb{V} \to V$ reads $\sigma(v) = \rho(s - t - 1 + \tilde{v})$. ■

Proof [Proof of Lemma 9, voltage problem] Let $G = (V, E, W)$ be a connected d-regular graph, with $d \geq 3$, where each edge has the same weight ω. Consider the setting in Section 5.2.1. Fix $\tilde{w} \in V$, $t \geq 3$, and let $\mathbb{T}$ be the computation tree of depth t rooted at $\tilde{w}$, with root vertex $\tilde{\mathbb{w}} \in \mathbb{V}$. Fix $\mathbb{v} \in \mathbb{V}^{t-1}$. For what follows, it is convenient to label some of the vertices in $\mathbb{V}$. Let $\mathbb{v} = 0, 1 = \mathcal{N}(\mathbb{v}) \setminus \mathbb{V}^t, 2 = \mathcal{N}(1) \setminus \mathbb{V}^{t-1}, \ldots, k+1 = \mathcal{N}(k) \setminus \mathbb{V}^{t-k}$, until we reach $\tilde{\mathbb{w}} = t$. See Figure 12. Given this set up for the computation tree, the remaining

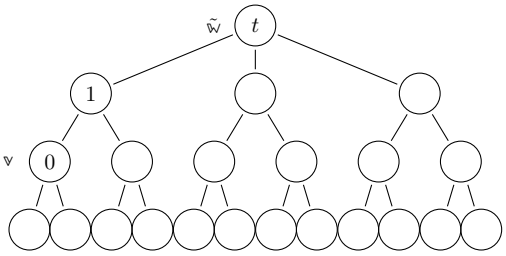

Figure 12: Computation tree $\mathbb{T} = (\mathbb{V}, \mathbb{E})$ of depth $t = 2$ associated to any given d-regular graph, with $d = 3$.

of the proof follows exactly the steps of the proof in the flow case. Using the analogue of Lemma 24 for the voltage case, we can compute $\bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{w}}\mathbb{v}}$ by solving a system of equations of the form (12) supported on the a reduced network that is exactly like the one in Figure 8. The only difference is that now $\mathbb{C}_t = \omega(d-2)(1+h_{t+1})$, while all other effective conductances $\mathbb{C}_s$, for $s \in \{0, \ldots, t-1\}$, read exactly as for the flow problem. Rearranging (13) we get $\bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{w}}\mathbb{v}} = \frac{1}{\omega d(d-1)^{t-1}} \frac{q_t}{p_t} \frac{1}{\xi_{t-1}}$, where $\xi_{t-1} := \frac{1}{(d-1)^{t-1}} \frac{\alpha_{t+1} - p_0 q_1 \beta_t}{q_1 \cdots q_{t-1} p_t}$ is defined as in the flow case by $\xi_s = \frac{p_s}{q_s p_{s+1}(d-1)} \xi_{s-1} - \frac{p_{s-1} p_s q_{s+1}}{q_{s-1} q_s p_{s+1} (d-1)^2} \xi_{s-2}$ for $s \in \{2, \ldots, t-1\}$, with $\xi_0 := \frac{p_0 q_1}{p_1}$ and $\xi_1 = \frac{1}{d-1} \frac{1 - p_0 q_1 - p_1 q_2}{q_1 p_2}$. Plugging in the definitions of the transition probabilities, noticing that p_t and q_t differ from the analogous quantities defined in the flow case due to the new definition of $\mathbb{C}_t$, recalling Proposition 8, it is immediate to verify that $\delta_s \equiv \xi_s$ for $s \in \{0, \ldots, t-2\}$ and $\frac{d-1}{(d-2)(1+h_{t+1})} b_{d,t} \equiv \xi_{t-1}$, so that $\bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{w}}\mathbb{v}} = \frac{1}{\omega d(d-1)^t} \frac{1}{b_{d,t}}$. Theorem 28 yields

$$\nu^\star_{\tilde{w}} - \hat{\nu}^t_{\tilde{w}} = \sum_{\mathbb{v} \in \mathbb{V}^{t-1}} \bar{\mathbb{L}}^{-1}_{\tilde{\mathbb{w}}\mathbb{v}} \sum_{\mathbb{w} \in \mathbb{V}^t} \mathbb{W}_{\mathbb{v}\mathbb{w}} \nu^\star_{\sigma(\mathbb{w})} = \frac{1}{d(d-1)^t} \frac{1}{b_{d,t}} \sum_{v \in V} \nu^\star_v \, \#^{t+1}_{\tilde{w} \to v}.$$

where $\#^{t+1}_{\tilde{w} \to v}$ is the number of non-backtracking paths in the original undirected graph $G = (V, E)$ that connect node $\tilde{w}$ to node v in $t+1$ time steps. The proof is easily concluded noticing that the total number of non-backtracking paths after $t+1$ steps is $d(d-1)^t$, so that $\vec{\mathbf{P}}_{\tilde{w}}(Y_{t+1} = v) = \#^{t+1}_{\tilde{w} \to v} / (d(d-1)^t)$. ■

Acknowledgments

We would like to acknowledge support for this project from the National Science Foundation (NSF grant ECCS-1609484).

Appendix A. Inverse of Restricted Laplacians and Hitting Times

In this appendix we establish a standalone general connection between the inverse of restricted Laplacian matrices—which are obtained by removing certain rows and columns from Laplacians matrices—and hitting times of random walks on graphs. This connection (Proposition 30 below) is used repeatedly in Section 5 to prove the error characterization of the min-sum algorithm.

Let $G = (V, E, W)$ be a simple (i.e., no self-loops, and no multiple edges), connected, undirected, weighted graph, where to each edge $\{v, w\} \in E$ is associated a positive weight $W_{vw} = W_{wv} > 0$, and $W_{vw} = 0$ if $\{v, w\} \notin E$. Let D be a diagonal matrix with entries $d_v = D_{vv} = \sum_{w \in V} W_{vw}$ for each $v \in V$. Let $L := D - W$ be the Laplacian matrix for G. Henceforth, for each $v \in V$, let $\mathbf{P}_v$ be the law of a time homogeneous Markov chain $X_0, X_1, X_2, \dots$ on V with transition matrix $P = D^{-1}W$ and initial condition $X_0 = v$. Analogously, denote by $\mathbf{E}_v$ the expectation with respect to this law. The hitting time to a set $Z \subseteq V$ is defined as $T_Z := \inf\{k \geq 0 : X_k \in Z\}$. Let $Z \subseteq V$ be fixed, and define $\bar{W}$ and $\bar{D}$ as the matrix obtained by removing the rows and columns associated to Z from W and D, respectively. Let $\bar{V} := V \setminus Z$, $\bar{E} := E \setminus \{\{u, v\} \in E : u \in Z \text{ or } v \in Z\}$ and consider the graph $\bar{G} := (\bar{V}, \bar{E})$. The quantity $\bar{L} := \bar{D} - \bar{W}$ represents a restricted Laplacian—sometimes also called grounded Laplacian. Let $\bar{P} := \bar{D}^{-1}\bar{W}$ be the transition matrix of the transient part of the killed random walk that is obtained from X by adding cemeteries at the sites in Z. Creating cemeteries at the sites in Z means modifying the walk X so that each site in Z becomes a recurrent state, i.e., once the walk is in state $\bar{z} \in Z$ it will go back to $\bar{z}$ with probably 1. This is clearly done by replacing the $\bar{z}$-th row of P by a row with zeros everywhere but in the $\bar{z}$-th coordinate, where the entry is equal to 1.

The relation between the transition matrix $\bar{P}$ of the killed random walk and the law of the random walk X itself is made explicit in the next proposition.

Proposition 29 *For any $v, w \in \bar{V}$, $k \geq 0$, we have $\bar{P}^k_{vw} = \mathbf{P}_v(X_k = w, T_Z > k)$.*

Proof Proof is by induction. Clearly, for any $v, w \in \bar{V}$, we have $\mathbf{P}_v(X_0 = w, T_Z > 0) = \mathbf{P}_v(X_0 = w) = \mathbf{1}_{v=w} = \bar{P}^0_{vw}$, which proves the statement for $k = 0$ ($\mathbf{1}_S$ is the indicator function defined as $\mathbf{1}_S = 1$ if statement S is true, $\mathbf{1}_S = 0$ otherwise). Assume that the statement holds for any time $i \geq 0$ up to $k > 0$. By the properties of conditional expectation, as $\{T_Z > k+1\} = \{X_0 \notin Z, \dots, X_{k+1} \notin Z\}$, we have $\mathbf{P}_v(X_{k+1} = w, T_Z > k+1) = \mathbf{E}_v[\mathbf{P}_v(X_{k+1} = w, T_Z > k+1 | X_0, \dots, X_k)] = \mathbf{E}_v[\mathbf{1}_{\{X_0 \notin Z, \dots, X_k \notin Z\}} \mathbf{P}_v(X_{k+1} = w, X_{k+1} \notin Z | X_0, \dots, X_k)]$ for any $v, w \in \bar{V}$. By the Markov property, on the event $\{X_k \notin Z\}$, we have $\mathbf{P}_v(X_{k+1} = w, X_{k+1} \notin Z | X_0, \dots, X_k) = \mathbf{P}_{X_k}(X_1 = w, X_1 \notin Z) = \bar{P}_{X_k w}$, so that by the induction hypothesis we have $\mathbf{P}_v(X_{k+1} = w, T_Z > k+1) = \mathbf{E}_v[\mathbf{1}_{\{T_Z > k\}} \bar{P}_{X_k w}] = \sum_{u \in V \setminus Z} \mathbf{P}_v(X_k = u, T_Z > k) \bar{P}_{uw} = \bar{P}^{k+1}_{vw}$, which proves the statement for $k+1$. ∎

We can now relate the inverse of the reduced Laplacian $\bar{L}$ with the hitting times of the original random walk X. The following result represents the main computational tool used in Section 5 to establish decay of correlation in the computation tree that supports the operations of the min-sum algorithm over time.

Proposition 30 *For $v, w \in \bar{V}$, $\bar{L}^{-1}_{vw} = \bar{L}^{-1}_{ww} \mathbf{P}_v(T_w < T_Z)$, and $\bar{L}^{-1}_{ww} = \frac{1}{d_w} \mathbf{E}_w \left[\sum_{k=0}^{T_Z} \mathbf{1}_{X_k = w} \right]$.*

Proof Let us first assume that $\bar{G}$ is connected. The matrix $\bar{P}$ is clearly sub-stochastic. Then $\bar{P}$ is irreducible (in the sense of Markov chains, i.e., for each $v, w \in \bar{V}$ there exists t to that $\bar{P}^k_{vw} \neq 0$) and the spectral radius of $\bar{P}$ is strictly less than 1 (see Corollary 6.2.28 in Horn and Johnson (1986), for instance), so that the Neumann series $\sum_{k=0}^{\infty} \bar{P}^k$ converges. The Neumann series expansion for $\bar{L}^{-1}$ yields $\bar{L}^{-1} = \sum_{k=0}^{\infty} (I - \bar{D}^{-1}\bar{L})^k \bar{D}^{-1} = \sum_{k=0}^{\infty} \bar{P}^k \bar{D}^{-1}$, or, $\bar{L}^{-1}_{vw} = \frac{1}{d_w} \sum_{k=0}^{\infty} \bar{P}^k_{vw}$. As $\bar{P}^k_{vw} = \mathbf{P}_v(X_k = w, T_Z > k)$ by Proposition 29, by the Monotone convergence theorem we can take the summation inside the expectation and get $\sum_{k=0}^{\infty} \bar{P}^k_{vw} = \sum_{k=0}^{\infty} \mathbf{E}_v[\mathbf{1}_{X_k=w} \mathbf{1}_{T_Z>k}] = \mathbf{E}_v[\sum_{k=0}^{T_Z-1} \mathbf{1}_{X_k=w}]$, which equals $\mathbf{E}_v[\sum_{k=0}^{T_Z} \mathbf{1}_{X_k=w}]$ as $X_{T_Z} \in Z$, and $w \notin Z$. Recall that if S is a stopping time for the Markov chain $X := X_0, X_1, X_2, \ldots$, then by the strong Markov property we know that, conditionally on $\{S < \infty\}$ and $\{X_S = w\}$, the chain $X_S, X_{S+1}, X_{S+2}, \ldots$ has the same law as the Markov chain $Y := Y_0, Y_1, Y_2, \ldots$ with transition matrix P and initial condition $Y_0 = w$, and Y is independent of $X_0, \ldots, X_S$. The hitting times T_w and T_Z are two stopping times for X, and so is their minimum $S := T_w \wedge T_Z$. As either $X_S = w$ or $X_S \in Z$, we have $\mathbf{E}_v[\sum_{k=0}^{T_Z} \mathbf{1}_{X_k=w}] = \mathbf{E}_v[\sum_{k=0}^{T_Z} \mathbf{1}_{X_k=w} | X_S = w] \mathbf{P}_v(X_S = w)$, where we used that, conditionally on $\{X_S \in Z\} = \{T_w > k_Z\}$, clearly $\sum_{k=0}^{T_Z} \mathbf{1}_{X_k=w} = 0$. Conditionally on $\{X_S = w\} = \{T_w < T_Z\} = \{S = T_w\}$, we have $T_Z = S + \inf\{k \geq 0 : X_{S+k} \in Z\}$, and the strong Markov property yields (note that the event $\{S < \infty\}$ has probability one from any starting point, as the graph G is connected by assumption so that the Markov chain will almost surely eventually hit either w or a site in Z) $\mathbf{E}_v[\sum_{k=0}^{T_Z} \mathbf{1}_{X_k=w} | X_S = w] = \mathbf{E}_v[\sum_{k=0}^{\inf\{k\geq 0 : X_{S+k} \in Z\}} \mathbf{1}_{X_{S+k}=w} | X_S = w] = \mathbf{E}_w[\sum_{k=0}^{T_Z} \mathbf{1}_{X_k=w}]$. Putting everything together we have $\bar{L}^{-1}_{vw} = \frac{1}{d_w} \mathbf{E}_w[\sum_{k=0}^{T_Z} \mathbf{1}_{X_k=w}] \mathbf{P}_v(T_w < T_Z)$. As $\mathbf{P}_w(T_w < T_Z) = 1$, clearly $\bar{L}^{-1}_{ww} = \frac{1}{d_w} \mathbf{E}_w[\sum_{k=0}^{T_Z} \mathbf{1}_{X_k=w}]$ so that $\bar{L}^{-1}_{vw} = \bar{L}^{-1}_{ww} \mathbf{P}_v(T_w < T_Z)$. The argument just presented extends easily to the case when $\bar{G}$ is not connected. In this case the matrix $\bar{P}$ has a block structure, where each block corresponds to a connected component and to a sub-stochastic submatrix, so the argument above can be applied to each block. ■

Appendix B. Technical Results

This appendix contains technical results that are used in the paper. The following result on quadratic optimization has been used repeatedly in the main text (see Section 2 and the proofs of Proposition 5 and Lemma 22).

Proposition 31 *Given $R \in \mathbb{R}^{m \times m}$ symmetric positive definite, $r \in \mathbb{R}^m$, $A \in \mathbb{R}^{n \times m}$, $b \in \mathbb{R}^n$ in the range of A, the unique solution of the optimization problem*

$$\begin{aligned} \textit{minimize} \quad & \frac{1}{2} x^T R x + r^T x \\ \textit{subject to} \quad & Ax = b \end{aligned}$$

over $x \in \mathbb{R}^m$ reads $x^\star = R^{-1}A^T L^+ b + (R^{-1}A^T L^+ A - I) R^{-1} r$ with $L := AR^{-1}A^T$.

Proof The KKT optimality equations for this problem read

$$\begin{pmatrix} R & -A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x \\ \nu \end{pmatrix} = \begin{pmatrix} -r \\ b \end{pmatrix}.$$

Solving the first equation for x we have $x = R^{-1}(A^T\nu - r)$, which, upon substitution into the second equation $Ax = b$ yields a solution $\nu = L^+(AR^{-1}r + b)$, and the proof follows. ■

We now present the proof of Proposition 8 in Section 4.

Proof [Proof of Proposition 8] Fix $d \geq 3$, $t \geq 3$. As $s \to h_s$ monotonically converges to 0 exponentially fast, we expect that the function $s \to \delta_s$, hence the values of both $b_{d,t}$ and $c_{d,t}$, can be controlled by the system that is obtained by replacing h_s with 0 for any $s \geq 0$. Thus, we define $\hat{\delta}_0 := \delta_0$, $\hat{\delta}_1 := \delta_1$, and $\hat{\delta}_s := \frac{1}{d-1}(2 + \frac{(d-2)^2}{d-1})\hat{\delta}_{s-1} - \frac{1}{(d-1)^2}\hat{\delta}_{s-2}, s \geq 2$. The characteristic equation of the dynamical system $\hat{\delta}_s$, $s \geq 0$, reads $(d-1)^2\lambda^2 - (d^2-2d+2)\lambda + 1 = 0$. This equation has two distinct real solutions: $\lambda_+ = 1$ and $\lambda_- = 1/(d-1)^2 < 1$. Imposing the initial conditions for $\hat{\delta}_0$ and $\hat{\delta}_1$, we find $\hat{\delta}_s = c_+ + c_-\lambda_-^s \geq 0$ with $c_+ = \frac{d(d-1)((d-1)^2+1)-1}{d((d-1)^2-1)} > 0$ and $c_- = 1/d - c_+ < 0$.

We begin by analyzing $c_{d,t}$. To this end, define $\hat{c}_{d,t} := \frac{1}{d-1}(1 + \frac{1}{(d-2)(1+h_t)})\hat{\delta}_{t-2} - \frac{1}{(d-1)^2}\hat{\delta}_{t-3}$. First, we prove that $c_{d,t} \geq 1$. By induction it is easy to show that $(d-1)(\delta_s - \hat{\delta}_s) \geq \delta_{s-1} - \hat{\delta}_{s-1} \geq 0$ for any $s \geq 1$, so we have $c_{d,t} \geq \hat{c}_{d,t}$. As $\hat{\delta}_s \geq c_+ + c_-\lambda_-$ for any $s \geq 1$ and $\hat{\delta}_s \leq c_+$ for any $s \geq 0$, we get $\hat{c}_{d,t} \geq \frac{1}{d-1}(1 + \frac{1}{(d-2)(1+h_1)})(c_+ + c_-\lambda_-) - \frac{1}{(d-1)^2}c_+ \geq 1$. Second, we prove that $c_{d,t} \leq 1 + \varepsilon_d$. Given $\alpha := 1 + \frac{1}{(d-1)^2}$, we first show that $(d-1)(\alpha\hat{\delta}_s - \delta_s) \geq \alpha\hat{\delta}_{s-1} - \delta_{s-1} \geq 0$ for any $s \geq 1$, so that $c_{d,t} \leq \alpha\hat{c}_{d,t}$. Define $\xi_s := \alpha\hat{\delta}_s - \delta_s$. We proceed by induction. Note that $(d-1)\xi_1 \geq \xi_0 \geq 0$. For a given $s \geq 3$, assume that $(d-1)\xi_r \geq \xi_{r-1}$ for all $r \leq s-1$. We will prove that $(d-1)\xi_s \geq \xi_{s-1}$. We have $(d-1)\xi_s - \xi_{s-1} = \frac{(d-1)\xi_{s-1} - \xi_{s-2}}{d-1} + \frac{(d-2)^2}{d-1}((1+h_{s+2})\xi_{s-1} - \alpha h_{s+2}\hat{\delta}_{s-1})$. As $\hat{\delta}_{s-1} \leq c_+$ and $(d-1)^{s-1}h_{s+2} \leq h_3$, with $c_+h_3 < 1$, by the induction hypothesis we have $\xi_{s-1} \geq \frac{\xi_1}{(d-1)^{s-2}} = \frac{(\alpha-1)\delta_1}{(d-1)^{s-2}}$ and $(d-1)\xi_s - \xi_{s-1} \geq \frac{(d-2)^2}{(d-1)^s}((d-1)(\alpha-1)\delta_1 - \alpha) = 0$, which proves the induction step. As $\frac{1}{d} = c_+ + c_- \leq \hat{\delta}_s \leq c_+$, we get $c_{d,t} \leq \alpha\hat{c}_{d,t} \leq \bar{\alpha} := \alpha(\frac{c_+}{d-1}(1 + \frac{1}{d-2}) - \frac{1}{d(d-1)^2}))$. The function $d \geq 3 \to \varepsilon_d := \bar{\alpha} - 1$ is decreasing, with $\varepsilon_d \to 0$ as $d \to \infty$, and $\varepsilon_3 < 3$.

By induction it is immediate to show that $(d-1)\delta_s \geq \delta_{s-1} \geq 0$ for any $s \geq 1$, so that a direct comparison yields $b_{d,t} \leq c_{d,t}$. Finally, we prove that $b_{d,t} \geq \frac{d-2}{d-1}$. Define $\hat{b}_{d,t} := \frac{1}{(d-1)^2}(1 + (d-2)(1+h_{t+1}))\hat{\delta}_{t-2} - \frac{(d-2)(1+h_{t+1})}{(d-1)^3}\hat{\delta}_{t-3}$. As $(d-1)(\delta_s - \hat{\delta}_s) \geq \delta_{s-1} - \hat{\delta}_{s-1} \geq 0$ for $s \geq 1$, we have $b_{d,t} \geq \hat{b}_{d,t}$. As $\hat{\delta}_s \geq \hat{\delta}_3 = c_+ + c_-\lambda_-^3$ for $s \geq 3$ and $\hat{\delta}_s \leq c_+$ for $s \geq 0$, by the monotonicity of h_s we get $\hat{b}_{d,t} \geq \frac{1}{(d-1)^2}(1 + (d-2))(c_+ + c_-\lambda_-^3) - \frac{(d-2)(1+h_1)}{(d-1)^3}c_+ \geq \frac{d-2}{d-1}$. ■

The next proposition yields a recursive formula to compute, in the case of d-regular graphs, the probability distributions $P_v^{(t)}$ and $P_v^{(t,w)}$ (interpreted as row vectors) given by the v-th row of the matrices $P^{(t)}$ and $P^{(t,w)}$ defined in (8), respectively. For any $v \in V$, let $\mathbf{1}_v \in \mathbb{R}^V$ be defined as $(\mathbf{1}_v)_z := 1$ if $v = z$, $(\mathbf{1}_v)_z := 0$ otherwise.

Proposition 32 *Let (V, E) be a connected d-regular graph with $d \geq 2$, and let $B \in \mathbb{R}^{V \times V}$ be its adjacency matrix, that is, the symmetric matrix defined by $B_{vw} := 1$ if $\{v, w\} \in E$,*

$B_{vw} := 0$ *otherwise. For any* $v \in V$ *we have*

$$P_v^{(1)} = \frac{\mathbf{1}_v^T B}{d}, \quad P_v^{(2)} = \frac{P_v^{(1)} B - \mathbf{1}_v^T}{d-1}, \quad P_v^{(t)} = \frac{P_v^{(t-1)} B - P_v^{(t-2)}}{d-1} \; t \geq 3,$$

Given $w \in V$ *such that* $\{v, w\} \in E$*, we have that* $P_v^{(t,w)}$ *satisfies the same recursion as above with initial condition* $P_v^{(1,w)} = (\mathbf{1}_v^T B - \mathbf{1}_w^T)/(d-1)$*.*

Proof We first prove the recursion for $P_v^{(t,w)}$. Fix $\{v, w\} \in E$, and let $\widetilde{\#}_z^t$ be the number of non-backtracking paths that connect node v to node z in t steps, where the first step is different from w. By definition, B_{vz} equals the number of paths that connect node v to node z in a single step. Clearly, $\widetilde{\#}_z^1 = B_{vz} - (\mathbf{1}_w)_z$, i.e., in row-vector form, $\widetilde{\#}^1 = \mathbf{1}_v^T B - \mathbf{1}_w^T$. It is immediate to verify that $\widetilde{\#}^2 = \widetilde{\#}^1 B - (d-1)\mathbf{1}_v^T$, as there are $d-1$ paths that can backtrack to v at time $t = 2$. For $t \geq 3$ we have $\widetilde{\#}^t = \widetilde{\#}^{t-1} B - (d-1)\widetilde{\#}^{t-2}$. In fact, $(\widetilde{\#}^{t-1} B)_z$ equals the number of paths that go from v to z in t steps such that the first move is different from w and the first $t-1$ steps are non-backtracking. To count only the paths that are non-backtracking for all t steps, we need to subtract from $(\widetilde{\#}^{t-1} B)_z$ the number of paths that backtrack at the t-th step, which are the ones that are at z both at step $t-2$ ant t. There are $(d-1)\widetilde{\#}_z^{t-2}$ of these paths, as vertex z has d neighbors but only $d-1$ moves are allowed at the $(t-1)$-th step due to the non-backtracking nature of the paths up to step $t-1$. The proof follows noticing that the total number of non-backtracking after t steps is $d(d-1)^{t-1}$, so that $\vec{\mathbf{P}}_v(Y_1 \neq w, Y_t = z) = \widetilde{\#}_z^t/(d(d-1)^{t-1})$ and $P_{vz}^{(t,w)} = \vec{\mathbf{P}}_v(Y_t = z | Y_1 \neq w) = \vec{\mathbf{P}}_v(Y_1 \neq w, Y_t = z)/\vec{\mathbf{P}}_v(Y_1 \neq w) = \widetilde{\#}_z^t/(d-1)^t$, $\vec{\mathbf{P}}_v(Y_1 \neq w) = (d-1)/d$. The recursion for $P_v^{(t)}$ is proved analogously. Let $\#_z^t$ be the number of non-backtracking paths that connect node v to node z in t steps. Clearly, $\#^1 = \mathbf{1}_v^T B$, $\#^2 = \#^1 B - d\mathbf{1}_v^T$, and $\#^t = \#^{t-1} B - (d-1)\#^{t-2}$. The proof follows as $P_{vz}^{(t)} = \vec{\mathbf{P}}_v(Y_t = z) = \#_z^t/(d(d-1)^{t-1})$. ■

References

Emmanuel Abbe. Community detection and stochastic block models. *Foundations and Trends® in Communications and Information Theory*, 14(1-2):1–162, 2018.

Jean Barbier, Florent Krzakala, Nicolas Macris, Léo Miolane, and Lenka Zdeborová. Optimal errors and phase transitions in high-dimensional generalized linear models. In *COLT*, volume 75 of *Proceedings of Machine Learning Research*, pages 728–731. PMLR, 2018.

C. Berrou, A. Glavieux, and P. Thitimajshima. Near shannon limit error-correcting coding and decoding: Turbo-codes. 1. In *Communications, 1993. ICC '93 Geneva. Technical Program, Conference Record, IEEE International Conference on*, volume 2, pages 1064–1070 vol.2, 1993.

A. Braunstein, M. Mézard, and R. Zecchina. Survey propagation: An algorithm for satisfiability. *Random Struct. Algorithms*, 27(2):201–226, 2005.

Michael B. Cohen, Rasmus Kyng, Gary L. Miller, Jakub W. Pachocki, Richard Peng, Anup B. Rao, and Shen Chen Xu. Solving sdd linear systems in nearly mlog1/2n time. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 343–352. ACM, 2014.

David L. Donoho, Arian Maleki, and Andrea Montanari. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences*, 106(45):18914–18919, 2009.

R.G. Gallager. *Low-density Parity-check Codes*. M.I.T. Press research monographs. M.I.T. Press, 1963.

David Gamarnik, Devavrat Shah, and Yehua Wei. Belief propagation for min-cost network flow: Convergence and correctness. *Operations Research*, 60(2):410–428, 2012.

Hans-Otto Georgii. *Gibbs measures and phase transitions*, volume 9 of *de Gruyter Studies in Mathematics*. Walter de Gruyter & Co., Berlin, second edition, 2011.

Roger A. Horn and Charles R. Johnson, editors. *Matrix Analysis*. Cambridge University Press, New York, NY, USA, 1986.

Jonathan A. Kelner, Lorenzo Orecchia, Aaron Sidford, and Zeyuan Allen Zhu. A simple, combinatorial algorithm for solving sdd systems in nearly-linear time. *CoRR*, abs/1301.6628, 2013.

I. Koutis, G. L. Miller, and R. Peng. A nearly-m log n time solver for sdd linear systems. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 590–598, 2011.

Rasmus Kyng and Sushant Sachdeva. Approximate Gaussian elimination for Laplacians - fast, sparse, and simple. *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 573–582, 2016.

David Asher Levin, Yuval Peres, and Elizabeth Lee Wilmer. *Markov chains and mixing times*. Providence, R.I. American Mathematical Society, 2009.

Dmitry M. Malioutov, Jason K. Johnson, and Alan S. Willsky. Walk-sums and belief propagation in gaussian graphical models. *J. Mach. Learn. Res.*, 7:2031–2064, 2006.

M. Mezard, G. Parisi, and Riccardo Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297:812 – 815, 2002.

C. C. Moallemi and B. Van Roy. Consensus propagation. *IEEE Transactions on Information Theory*, 52(11):4753–4766, Nov 2006.

C. C. Moallemi and B. Van Roy. Convergence of min-sum message passing for quadratic optimization. *IEEE Transactions on Information Theory*, 55(5):2413–2423, 2009.

Ciamac C. Moallemi and Benjamin Van Roy. Convergence of min-sum message-passing for convex optimization. *Information Theory, IEEE Transactions on*, 56(4):2041–2050, 2010.

Andrea Montanari. *Graphical models concepts in compressed sensing*, pages 394–438. Cambridge University Press, 2012.

Richard Peng and Daniel A. Spielman. An efficient parallel solver for sdd linear systems. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 333–342, New York, NY, USA, 2014. ACM.

J. Laurie Snell Peter G. Doyle. *Random Walks and Electric Networks*, volume 22. Mathematical Association of America, 1 edition, 1984.

T. J. Richardson and R. L. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Transactions on Information Theory*, 47(2):599–618, 2001.

P. Rusmevichientong and B. Van Roy. An analysis of belief propagation on the turbo decoding graph with gaussian densities. *IEEE Transactions on Information Theory*, 47 (2):745–765, 2001.

Daniel A Spielman. Algorithms, graph theory, and linear equations in laplacian matrices. In *Proceedings of the International Congress of Mathematicians*, volume 4, pages 2698–2722, 2010.

Daniel A. Spielman. *Algorithms, graph theory, and linear equations in Laplacian matrices.*, pages 2698–2722. Hackensack, NJ: World Scientific; New Delhi: Hindustan Book Agency, 2011.

Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing*, STOC '04, pages 81–90. ACM, 2004.

Daniel A. Spielman and Shang-Hua Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM Journal on Matrix Analysis and Applications*, 35(3):835–885, 2014.

Sekhar C. Tatikonda and Michael I. Jordan. Loopy belief propagation and Gibbs measures. In *Proc. UAI*, volume 18, pages 493–500, 2002.

Nisheeth K. Vishnoi. Lx = b Laplacian Solvers and Their Algorithmic Applications. *Foundations and Trends® in Theoretical Computer Science*, 8(1–2):1–141, 2013.

Yair Weiss and William T. Freeman. Correctness of Belief Propagation in Gaussian Graphical Models of Arbitrary Topology. *Neural Computation*, 13(10):2173–2200, 2001.