Pooling or Sampling: Collective Dynamics for Electrical Flow Estimation

Luca Becchetti
Sapienza University of Rome
Rome, Italy
becchetti@diag.uniroma1.it

Vincenzo Bonifaci
Consiglio Nazionale delle Ricerche
Rome, Italy
vincenzo.bonifaci@iasi.cnr.it

Emanuele Natale
Max Planck Institute for Informatics
Saarbrücken, Germany
enatale@mpi-inf.mpg.de

ABSTRACT

The computation of electrical flows is a crucial primitive for many recently proposed optimization algorithms on weighted networks. While typically implemented as a centralized subroutine, the ability to perform this task in a fully decentralized way is implicit in a number of biological systems. Thus, a natural question is whether this task can provably be accomplished in an efficient way by a network of agents executing a simple protocol.

We provide a positive answer, proposing two distributed approaches to electrical flow computation on a weighted network: a deterministic process mimicking Jacobi's iterative method for solving linear systems, and a randomized token diffusion process, based on revisiting a classical random walk process on a graph with an absorbing node. We show that both processes converge to a solution of Kirchhoff's node potential equations, derive bounds on their convergence rates in terms of the weights of the network, and analyze their time and message complexity.

KEYWORDS

token diffusion; electrical flow; Laplacian system; Kirchhoff's equations; Jacobi's method

ACM Reference Format:
Luca Becchetti, Vincenzo Bonifaci, and Emanuele Natale. 2018. Pooling or Sampling: Collective Dynamics for Electrical Flow Estimation. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 9 pages.

1 INTRODUCTION

The computation of currents and voltages in a resistive electrical network, besides being an interesting problem on its own, is a crucial primitive in many recently proposed optimization algorithms on weighted networks. Examples include the fast computation of maximum flows [6, 14], network sparsification [24], and the generation of random spanning trees [12].

Solving the electrical flow problem requires solving a system of linear equations, whose variables are the electrical voltages, or "potentials", at the nodes of the network (equivalently, the currents traversing its edges). Performing this task can be computationally nontrivial, and is typically achieved in a centralized fashion.

At the same time, the ability to perform this task in a fully decentralized way is implicit in a number of biological systems by virtue of the electronic-hydraulic analogy [13], including the *P. polycephalum* slime mold [4, 25, 26] and ant colonies [17]. These organisms have been showed to implicitly solve the electrical flow problem in the process of forming food-transportation networks. Such capability of biological systems naturally raises the following questions, which motivate our paper:

(Q1) Can this task be collectively accomplished by the network itself, if every node is an agent that follows an elementary protocol, and each agent can only interact with its immediate neighbours, otherwise possessing no knowledge of the underlying topology?
(Q2) In case of a positive answer to *Q1*, what is the involved computational effort for the network, in terms of convergence time and communication overhead?

We address the two aforementioned questions by providing analytical bounds which are of interest for many bio-inspired multiagent systems in swarm robotics and sensor networks (e.g. [11, 21]).

1.1 Our Contribution

We propose two complementary, fully decentralized approaches to electrical flow computation on a weighted network. In particular, Towards question (Q1), we make the following contributions:

(1) We consider a deterministic distributed process, based on *Jacobi's iterative method* for solving linear systems. This process converges to a solution of Kirchhoff's node potential equations. We bound the convergence rate of this process in terms of a graph-theoretic parameter of the network – *graph conductance*.
(2) Driven by a natural probabilistic interpretation of the aforementioned process, we further consider a *randomized token diffusion* process, implementing Monte Carlo sampling via independent random walks. This process also converges to a solution of Kirchhoff's node potential equations, but differently from the deterministic algorithm, randomized token diffusion does not involve any arithmetics on real numbers: each agent/node simply maintains a counter of the number of random walks currently visiting it, from which a simple unbiased estimator of the node's potential can be derived. We derive a bound on the convergence rate of this process in terms of another graph-theoretic parameter – *edge expansion*.

With respect to question (Q2), while the strong connection between electrical flows and random walks has been known for a while and has been extensively investigated in the past [8, 27], any effective exploitation of electrical flow computation crucially requires explicit and plausible bounds on the efficiency and accuracy of the algorithm(s) under consideration. In this respect, besides establishing the correctness of the two algorithmic approaches

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden. © 2018 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

above, our core contribution is to derive detailed bounds on their time and communication complexities in terms of fundamental combinatorial properties of the network.

Finally, our results highlight the algorithmic potential of classical models of opinion formation, as discussed in more detail in the next section.

1.2 Related Work

We briefly review contributions that are most closely related to the spirit of this work.

Computing electrical flows. The problem of computing voltages and currents of a given resistive network, that is, the question of solving Kirchhoff's equations, is a classical example of solving a linear equation system with a *Laplacian* constraint matrix [30]. While Jacobi's method is a well-known approach to the solution of a class of linear systems that subsumes Laplacian systems, its complexity analysis in the literature (for example, in [22, 29]) is generic, and does not exploit the additional matrix structure that is inherent in Laplacian systems. In our setting, existing results on the convergence of Jacobi's method could be leveraged to prove convergence to a correct solution, but they would fail to provide explicit bounds on convergence rate.

Electrical flows and random walks. Relations between electrical quantities and statistical properties of random walks have been known for a long time, and are nicely discussed, for example, in Doyle and Snell [8], Lovasz [16], and Levin, Peres and Wilmer [15]. In particular, in their monograph, Doyle and Snell point out the interpretation of the electrical current along an edge as the *expected number of net traversals* of the edge by a random walker that is injected at the source node and absorbed at the sink node of the flow. While our randomized token diffusion algorithm refers to the same underlying process, it crucially differs from the former in the interpretation of electrical current, as it links the electric potential of a node to the expected number of random walks currently visting the node (see Section 4).

This connection has been also explored by Chandra et al. [5], who characterized the *cover time* of the random walk in terms of the maximal effective resistance of the network, as well as by Tetali [27], who characterized the *hitting time* of the random walk in terms of the effective resistance between source and sink. Tetali [27] also proved that the expected number of visits to a node by a random walker injected at the source and absorbed at the sink is related to the electrical potential of the node in a simple way. In principle, like the one discussed by Doyle and Snell, this characterization could be used as the basis for another random walk-based approach to the estimation of electric potentials, with essentially the same complexity as the method we propose. Still, this is a static characterization that, by itself, does not provide an iterative algorithm or error bound. On the contrary, our interpretation results in an estimator, which depends solely on the number of tokens at each node and is thus entirely *local*, as opposed to previous methods, which entail tracking an event that depends on *global* properties of the network (such as the hitting time of a specific node or the absorption at the sink). Hence, we believe our proposed randomized diffusion process is more suitable to accommodate dynamic changes in the weights of the network when coupled with other processes, such as the *Physarum* dynamics [4, 17].

Electrical flows and complex systems. Understanding how electrical flows are computed in a decentralized fashion can help explain the emergent behavior of certain social and biological systems. For example, foraging behaviors of the *P. polycephalum* slime mold [4, 25, 26] and of ant colonies [17] can both be formulated in terms of *current-reinforced* random walks, see Ma et al. [17]. In this respect, our results can be seen as a step towards a more thorough understanding of these complex biological processes, *at the microscopic scale*. Moreover, the simple processes we propose shed new light on the computational properties of models of opinion dynamics in social networks [1, 7, 20]. In particular, the classical model of opinion formation proposed by DeGroot [7] essentially corresponds to the decentralized version of Jacobi's iterative method presented in this paper.[1] This is a hint that opinion dynamics are extremely versatile processes, whose algorithmic potential is not completely understood.

Distributed optimization. Since the electrical flow is one of minimum energy, the decentralized computation of electrical flows can be seen as an instance of *distributed optimization*, akin to the problems considered within the multiagent framework introduced by Tsui and Liu [28] and of potential interest for the class of distributed constraint optimization problems considered, for example, in Modi et al. [19]. Although for other distributed problems it has been suggested that Laplacian-based approaches are not the most computationally effective [9], as we have mentioned in the previous paragraph current-reinforced random walks are considered a feasible model, at least for certain biological systems [10, 17]. In this paper, we leveraged the specific structure of electrical flows to prove the effectiveness of our decentralized solutions. In particular, the dependency of convergence rates on the size of the network is polynomial, which cannot be claimed for other more generic approaches to distributed optimization.

Finally, in the rather different context of social choice and agents with preferences (as opposed to our perspective motivated by natural processes for network optimization), [23] investigate mechanisms for social choice on social networks as a weighted form of classical preference aggregation. One of the update processes they consider is loosely related to our Jacobi process.

1.3 Outline

The rest of this paper is organized as follows. In Section 2 we discuss some preliminaries about electrical networks and flows and set up the necessary notation and terminology. In Section 3 we describe and analyze the deterministic distributed algorithm based on Jacobi's method for the solution of Kirchhoff's equations. In Section 4 we propose and analyze our randomized token-diffusion method for the estimation of the electric potentials. We conclude by summarizing our findings in Section 5.

[1] The only difference is the presence of two "special" agents (the source and the sink), whose behaviours slightly differ from the others, in that they exchange information with the exterior in the form of a current flow.

2 PRELIMINARIES ON ELECTRICAL NETWORKS AND NOTATION

We consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, with node set $\mathcal{V}$, edge set $\mathcal{E}$, and positive edge weights $(w_e)_{e\in\mathcal{E}}$ representing electrical conductances. We also denote the weight of an edge by w_{uv} if $u, v \in \mathcal{V}$ are the endpoints of the edge; if no edge corresponds to the pair (u, v), $w_{uv} = 0$. We use n and m to denote the number of nodes and edges, respectively, of the graph. Without loss of generality we assume that $\mathcal{V} = \{1, 2, \ldots, n\}$.

The (weighted) *adjacency matrix* of $\mathcal{G}$ is the matrix $A \in \mathbb{R}^{n\times n}$ whose (u, v)-entry is equal to w_{uv} if $\{u, v\} \in \mathcal{E}$, and to 0 otherwise. The *volume* (or generalized degree) of a node v is the total weight of the edges incident to it, and is denoted by $\mathrm{vol}(v)$. The generalized *degree matrix* $D \in \mathbb{R}^{n\times n}$ is the diagonal matrix with $D_{uu} = \mathrm{vol}(u)$. The matrix $P \stackrel{\text{def}}{=} D^{-1}A$ is the *transition matrix* of $\mathcal{G}$; we denote its eigenvalues (which are all real) by $\rho_1 \geq \ldots \geq \rho_n$. We use $\mathrm{vol}_{\min}$ and $\mathrm{vol}_{\max}$ to denote the smallest and largest volume, respectively, of the nodes of $\mathcal{G}$.

We adhere to standard linear algebra notation, and we reserve boldface type for vectors. We denote by $\boldsymbol{\chi}_i$ the i-th standard basis vector, that is, a vector whose entries are 0 except for the i-th entry which is 1. With $\mathbf{0}$ and $\mathbf{1}$ we denote vectors with entries all equal to 0 and 1, respectively.

In the next sections, we make use of the following fact.

Fact 2.1. The transition matrix $P \stackrel{\text{def}}{=} D^{-1}A$ is similar to the symmetric matrix $N \stackrel{\text{def}}{=} D^{-1/2}AD^{-1/2}$ via conjugation by the matrix $D^{-1/2}$. In particular, we have

$$P^t = (D^{-1}A)^t = (D^{-\frac{1}{2}}ND^{\frac{1}{2}})^t = D^{-\frac{1}{2}}N^tD^{\frac{1}{2}}.$$

Moreover, thanks to the fact that N is symmetric, N has n orthonormal eigenvectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$, which correspond to the eigenvectors $\mathbf{y}_1, \ldots, \mathbf{y}_n$ of P via the similarity transformation $\mathbf{x}_i = D^{1/2}\mathbf{y}_i$ for each i. Observe also that both $\mathbf{x}_i$ and $\mathbf{y}_i$, for each i, are associated to the same eigenvalue ρ_i of P.

The *Laplacian matrix* of $\mathcal{G}$ is the $n \times n$ matrix $L \stackrel{\text{def}}{=} D - A$. We denote by $\Lambda_1 \leq \ldots \leq \Lambda_n$ the eigenvalues of L. We often use the facts that $\Lambda_1 = 0$ and that $L \cdot \mathbf{1} = \mathbf{0}$.

In our setting, one node of the graph acts as the *source*, and one as the *sink* of the electrical flow. Kirchhoff's equations for a network $\mathcal{G}$ are then neatly expressed by the linear system

$$L\mathbf{p} = \mathbf{b},$$

where $\mathbf{p}$ is the unknown vector of electric *potentials*, and $\mathbf{b} \in \mathbb{R}^n$ is a vector such that $b_{\text{source}} = 1$, $b_{\text{sink}} = -1$, and $b_u = 0$ if $u \notin \{\text{source}, \text{sink}\}$. The *electrical flow* is easily obtained from the vector $\mathbf{p}$: the electrical flow along an edge $\{u, v\}$, in the direction from u to v, equals $w_{uv} \cdot (p_u - p_v)$.

For a given weighted graph with a source and a sink, the electrical flow is uniquely defined. We remark however that Kirchhoff's equations have infinite solutions, since electric potentials are defined up to any constant offset: if $L\mathbf{p} = \mathbf{b}$, then $L(\mathbf{p} + c\mathbf{1}) = \mathbf{b}$ for any constant c (since $L\mathbf{1} = \mathbf{0}$.) We call the (unique) solution $\mathbf{p}$ such that $p_{\text{sink}} = 0$ the *grounded* solution to Kirchhoff's equations.

The *graph conductance* (or *bottleneck ratio*) of graph $\mathcal{G}$ is the constant

$$\phi(\mathcal{G}) \stackrel{\text{def}}{=} \min_{S\subset\mathcal{V}\,:\,\mathrm{vol}(S)\leq\mathrm{vol}(\mathcal{V})/2} \frac{w(S, \mathcal{V}\setminus S)}{\mathrm{vol}(S)},$$

where $\mathrm{vol}(S)$ denotes the total volume of the nodes in S, and $w(S, \mathcal{V}\setminus S)$ denotes the total weight of the edges crossing the cut $(S, \mathcal{V}\setminus S)$. The *edge expansion* of $\mathcal{G}$ is the constant

$$\theta(\mathcal{G}) \stackrel{\text{def}}{=} \min_{S\subset\mathcal{V}\,:\,|S|\leq n/2} \frac{w(S, \mathcal{V}\setminus S)}{|S|}.$$

The graph conductance is a number between 0 and 1, while the edge expansion is a number between 0 and $\mathrm{vol}_{\min}$.

3 JACOBI'S METHOD

The potentials $\mathbf{p}$ are a solution of the linear system

$$L\mathbf{p} = \mathbf{b}. \tag{1}$$

A classic parallel iterative algorithm for solving such a system is Jacobi's method [22, 29], which goes as follows. System (1) can be rewritten as $D\mathbf{p} - A\mathbf{p} = \mathbf{b}$, which is equivalent to

$$\mathbf{p} = D^{-1}\left(A\mathbf{p} + \mathbf{b}\right). \tag{2}$$

The idea underlying Jacobi's method is to introduce the related linear recurrence system

$$\tilde{\mathbf{p}}\,(t+1) = D^{-1}\left(A\tilde{\mathbf{p}}\,(t) + \mathbf{b}\right). \tag{3}$$

For any node $u \in \mathcal{V}$, (3) becomes

$$\tilde{p}_u(t+1) = \frac{b_u + \sum_{v\sim u} w_{uv}\tilde{p}_v(t)}{\sum_{v\sim u} w_{uv}}, \tag{4}$$

where the sums range on all neighbors v of u. Note that the denominator in (4) equals $\mathrm{vol}(u)$.

Indeed, from (2) we know that $\tilde{\mathbf{p}}$ is a fixed point of (4) whenever $L\tilde{\mathbf{p}} = \mathbf{b}$. This suggests running the following message-passing algorithm at every node $u \in \mathcal{V}$ (Algorithm 1).

```
   Parameter: u ∈ V
1  while true do
       // Step 1: Send potentials
2      for every neighbor v of u do
3          send p̃_u to v
4      end
       // Step 2: Receive potentials
5      for every neighbor v of u do
6          receive p̃_v from v
7      end
       // Step 3: Update potentials
8      p̃_u = 1/vol(u) (b_u + Σ_{v~u} w_uv p̃_v)
9  end
```

Algorithm 1: Jacobi's method for solving $L\mathbf{p} = \mathbf{b}$.

Algorithm 1 does not specify an initial value for $\tilde{\mathbf{p}}$: any initial value can be used. There is also no explicit termination condition. Terminating the algorithm sooner or later has only an effect on the numerical error, as explained in the next subsection.

3.1 Correctness and rate of convergence

To study the convergence of Algorithm 1, fix any solution $\mathbf{p}$ of (1), and define the *error at step t* as

$$\mathbf{e}(t) \stackrel{\text{def}}{=} \mathbf{p} - \tilde{\mathbf{p}}(t) = \mathbf{e}_{\perp}(t) + \alpha(t) \cdot \mathbf{1}, \tag{5}$$

where $\mathbf{e}_{\perp}(t)$ is the projection of $\mathbf{p} - \tilde{\mathbf{p}}(t)$ on the subspace orthogonal to $\mathbf{1}$, i.e.,

$$\mathbf{e}_{\perp}(t) = \left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\top}\right)\mathbf{e}(t),$$

while $\alpha(t) \cdot \mathbf{1}$ is the component of $\mathbf{p} - \tilde{\mathbf{p}}(t)$ parallel to $\mathbf{1}$, i.e.,

$$\alpha(t) \cdot \mathbf{1} = \frac{1}{n}\mathbf{1}\mathbf{1}^{\top}\mathbf{e}(t).$$

The reason we decompose $\mathbf{e}(t)$ as in (5) is that $\mathbf{p}$ is defined in (1) *up to translation along* $\mathbf{1}$, since $\mathbf{1}$ is in the kernel of L: any vector $\mathbf{p} + \beta \cdot \mathbf{1}$ satisfies $L(\mathbf{p} + \beta \cdot \mathbf{1}) = \mathbf{b}$ as well. Therefore, one has converged to a solution of (1) as soon as $\tilde{\mathbf{p}} = \mathbf{p} + \beta \cdot \mathbf{1}$ for any β, which implies that in (5) we do not care about the value of α, we only care about $\mathbf{e}_{\perp}(t)$ having a small norm. This becomes clear by plugging the decomposition (5) in (3):

$$\begin{aligned}
\mathbf{p} - \mathbf{e}_{\perp}(t+1) - \alpha(t+1) \cdot \mathbf{1} &\stackrel{\text{def}}{=} \tilde{\mathbf{p}}(t+1) \\
&= D^{-1}\left(A\tilde{\mathbf{p}}(t) + \mathbf{b}\right) \\
&= D^{-1}\left(A\left(\mathbf{p} - \mathbf{e}_{\perp}(t) - \alpha(t) \cdot \mathbf{1}\right) + \mathbf{b}\right) \\
&\stackrel{(a)}{=} \mathbf{p} - P\mathbf{e}_{\perp}(t) - \alpha(t) \cdot \mathbf{1},
\end{aligned} \tag{6}$$

where in (a) we used that for any transition matrix $P \stackrel{\text{def}}{=} D^{-1}A$ it holds $P\mathbf{1} = \mathbf{1}$ and (2). From (6) it follows that $\mathbf{e}_{\perp}(t+1) = P\mathbf{e}_{\perp}(t) - (\alpha(t+1) - \alpha(t))\mathbf{1}$.

By projecting, it also follows that

$$\begin{aligned}
\mathbf{e}_{\perp}(t+1) &= \left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\top}\right)P\mathbf{e}_{\perp}(t) \\
&= \left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\top}\right)P\left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\top}\right)P\mathbf{e}_{\perp}(t-1) \\
&\stackrel{(a)}{=} \left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\top}\right)\left(P - \frac{1}{n}\mathbf{1}\mathbf{1}^{\top}\right)P\mathbf{e}_{\perp}(t-1) \\
&\stackrel{(b)}{=} \left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\top}\right)P^2\mathbf{e}_{\perp}(t-1),
\end{aligned}$$

where in (a) we used again that $P\mathbf{1} = \mathbf{1}$ and in (b) we used that $\left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\top}\right)\frac{1}{n}\mathbf{1}\mathbf{1}^{\top} = 0$. By repeating steps (a) and (b) above we can unroll the previous equation and get

$$\mathbf{e}_{\perp}(t) = \left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\top}\right)P^t\mathbf{e}_{\perp}(0). \tag{7}$$

Recall Fact 2.1, which implies that $\mathbf{x}_1 = \|D^{1/2}\mathbf{1}\|^{-1}D^{1/2}\mathbf{1}$, and that we can write

$$N = \|D^{\frac{1}{2}}\mathbf{1}\|^{-2}D^{\frac{1}{2}}\mathbf{1}(D^{\frac{1}{2}}\mathbf{1})^{\top} + \sum_{i=2}^{n}\rho_i\mathbf{x}_i\mathbf{x}_i^{\top}.$$

Combining the previous observations with (7), we get

$$\begin{aligned}
\|\mathbf{e}_{\perp}(t)\| &= \left\|\left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\top}\right)P^t\mathbf{e}_{\perp}(0)\right\| \\
&= \left\|\left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\top}\right)D^{-\frac{1}{2}}N^tD^{\frac{1}{2}}\mathbf{e}_{\perp}(0)\right\| \\
&\stackrel{(a)}{=} \left\|\left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\top}\right)D^{-\frac{1}{2}}\left(\sum_{i=2}^{n}\rho_i^t\mathbf{x}_i\mathbf{x}_i^{\top}\right)D^{\frac{1}{2}}\mathbf{e}_{\perp}(0)\right\| \\
&\stackrel{(b)}{\leq} \left\|\left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\top}\right)\right\|\left\|D^{-\frac{1}{2}}\right\|\left\|\sum_{i=2}^{n}\rho_i^t\mathbf{x}_i\mathbf{x}_i^{\top}\right\|\left\|D^{\frac{1}{2}}\right\|\|\mathbf{e}_{\perp}(0)\| \\
&\stackrel{(c)}{\leq} \sqrt{\frac{\text{vol}_{\max}}{\text{vol}_{\min}}}\,\rho_*^t\,\|\mathbf{e}_{\perp}(0)\|,
\end{aligned} \tag{8}$$

where in (a) we used the fact that $\left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\top}\right)D^{-1/2}\mathbf{x}_1 = 0$, in (b) we used the submultiplicativity of the norm and in (c) we used that $\|D^{-1/2}\| \leq (\text{vol}_{\min})^{-1/2}$, that $\|D^{1/2}\| \leq (\text{vol}_{\max})^{1/2}$ and where by definition $\rho_* = \max_{i \neq 1}|\rho_i| = \max(|\rho_2|, |\rho_n|)$.

3.2 Time and message complexity

In summary, the arguments in the previous subsection prove the following.

Theorem 3.1. *After t rounds, the orthogonal component of the error of the solution $\tilde{\mathbf{p}}(t)$ produced by Algorithm 1 is reduced by a factor*

$$\frac{\|\mathbf{e}_{\perp}(t)\|}{\|\mathbf{e}_{\perp}(0)\|} \leq \left(\frac{\text{vol}_{\max}}{\text{vol}_{\min}}\right)^{1/2}\rho_*^t, \tag{9}$$

where ρ_* is the second largest absolute value of an eigenvalue of P. The message complexity per round is $O(m)$.

Proof. The first claim follows by (8). For the second part of the claim, note that at any round of Algorithm 1, each node sends its estimated potential value to each of its neighbors. Therefore, the number of exchanged messages is $O(m)$ per round. □

Observe that in typical applications we have $\rho_* = \rho_2$. This is the case, for example, when one considers the *lazy* variant of a transition matrix in order to avoid pathological cases [15, Section 1.3]. The condition $\rho_* = \rho_2$ implies, in particular, that (9) in Theorem 3.1 can be bounded in terms of the *graph conductance* of the network,[2] since for any graph $\mathcal{G}$, $\rho_2(\mathcal{G}) \leq 1 - \phi(\mathcal{G})^2/2$ [15, Theorem 13.14].

4 A TOKEN DIFFUSION METHOD

Following a well-known analogy between electrical flows and random walks [5, 8, 15, 27], in this section we propose a random walk-based approach to approximate electric potentials. The process is described by Algorithm 2. In each round, the algorithm starts K new, mutually independent random walks at the source node. Each random walker (or *token*) moves one step during each round of the algorithm, until it reaches the sink node, where it is absorbed. The independent parameter K controls the accuracy of the process.

Let $Z_K^{(t)}(u)$ denote the number of tokens at vertex u at the end of round t, when K new independent random walks are started at the source. Our *estimator* of the potential at node u at time t will be

$$V_K^{(t)}(u) \stackrel{\text{def}}{=} \frac{Z_K^{(t)}(u)}{K \cdot \text{vol}(u)}.$$

We next show that *in expectation*, our estimator evolves following a recurrence that, though not identical, is very close to (4).

[2] We remark that here the term *conductance* refers to the graph-theoretic notion also known as *bottleneck ratio* [15], and shall not be confused with the notion of *electrical conductance* in the theory of electrical networks [8].

Parameters: $u \in \mathcal{V}, K \in \mathbb{N}$
// Step 1: Send tokens
for *every token* T *on* u ***and*** *every neighbour* v *of* u **do**
  with prob $\propto w_{uv}$ **do**
    send T to v
    $Z(u) = Z(u) - 1$
end
// Step 2: Receive tokens
for *every token* T *received* **do**
  $Z(u) = Z(u) + 1$
end
// Step 3: Replenish source, deplete sink
if $u = source$ **then**
  $Z(u) = Z(u) + K$ // inject K new tokens at u
end
if $u = sink$ **then**
  $Z(u) = 0$ // absorb all tokens at u
end

Algorithm 2: Random walk algoritm.

Input: $u \in \mathcal{V}, K \in \mathbb{N}$
Return $\frac{\mathbf{Z}(u)}{K\mathrm{vol}(u)}$

Algorithm 3: Estimator.

Lemma 4.1. *Consider Algorithm 2 with* $K = 1$. *Define inductively* $\mathbf{p}^{(t)} \in \mathbb{R}^{\mathcal{V}}$ *by*

$$p_u^{(0)} = 0, \qquad \text{for all } u \in \mathcal{V}, \tag{10}$$

$$p_u^{(t+1)} = \begin{cases} \frac{1}{\mathrm{vol}(u)} \left(\sum_{v \sim u} w_{uv} p_v^{(t)} + b_u \right) & \text{if } u \neq \text{sink}, \\ 0 & \text{if } u = \text{sink}. \end{cases} \tag{11}$$

Then, for every time $t = 0, 1, 2, \ldots$ *and for every* $u \in \mathcal{V}$ *we have:*

$$\mathbb{E}\left[V_1^{(t)}(u)\right] = p_u^{(t)}.$$

Proof. The claim is proved by induction. It clearly holds when $t = 0$, since at that time there are no tokens and thus $V_1^{(t)} = \mathbf{0} = p^{(0)}$. For $t \geq 1$, and for every $u \in \mathcal{V} \setminus \{sink\}$:

$$\begin{aligned} \mathbb{E}\left[Z_1^{(t+1)}(u) \mid Z_1^{(t)}\right] &= \sum_{v \sim u} \frac{Z_1^{(t)}(v) w_{vu}}{\mathrm{vol}(v)} + b_u \\ &= \sum_{v \sim u} w_{vu} V_1^{(t)}(v) + b_u, \end{aligned}$$

where we used $V_1^{(t)}(v) = Z_1^{(t)}(v)/\mathrm{vol}(v)$. Dividing both sides by $\mathrm{vol}(u)$, we obtain:

$$\frac{\mathbb{E}\left[Z_1^{(t+1)}(u) \mid Z_1^{(t)}\right]}{\mathrm{vol}(u)} = \frac{1}{\mathrm{vol}(u)} \left(\sum_{v \sim u} w_{vu} V_1^{(t)}(v) + b_u \right).$$

By recalling that $V_1^{(t+1)}(u) = Z_1^{(t+1)}(u)/\mathrm{vol}(u)$ and by the law of iterated expectations we obtain:

$$\mathbb{E}\left[V_1^{(t+1)}(u)\right] = \begin{cases} \frac{1}{\mathrm{vol}(u)} \left(\sum_{v \sim u} w_{vu} \mathbb{E}\left[V_1^{(t)}(v)\right] + b_u \right) & \text{if } u \neq \text{sink}, \\ 0 & \text{if } u = \text{sink}. \end{cases} \tag{12}$$

Recurrence (12) has the very same form as (11). In particular, from the inductive hypothesis $\mathbb{E}\left[V_1^{(t)}\right] = \mathbf{p}^{(t)}$ we obtain $\mathbb{E}\left[V_1^{(t+1)}\right] = \mathbf{p}^{(t+1)}$. This completes the proof. □

The following corollary justifies our estimator in Algorithm 3, when $K > 1$:

Corollary 4.2. *Let*

$$V_K^{(t)}(u) \stackrel{\text{def}}{=} \frac{Z_K^{(t)}(u)}{K\mathrm{vol}(u)},$$

for every $t = 0, 1, 2, \ldots$ *and* $u \in \mathcal{V}$. *Then:*

$$\mathbb{E}\left[V_K^{(t)}(u)\right] = p_u^{(t)}.$$

Proof. First of all observe that, obviously:

$$\mathbb{E}\left[Z_K^{(t)}(u)\right] = K\mathbb{E}\left[Z_1^{(t)}(u)\right].$$

As a consequence:

$$\frac{\mathbb{E}\left[Z_K^{(t)}(u)\right]}{\mathrm{vol}(u)} = K\frac{\mathbb{E}\left[Z_1^{(t)}(u)\right]}{\mathrm{vol}(u)} = Kp_u^{(t)},$$

from Lemma 4.1. This proves the claim. □

Note that the definition of $\mathbf{p}^{(t)}$ in Lemma 4.1 is akin to that of $\tilde{\mathbf{p}}(t)$ in Equation 4. One might thus reasonably expect that, like $\tilde{\mathbf{p}}(t)$, $\mathbf{p}^{(t)}$ also converges to a solution of Kirchhoff's equations. Nevertheless, the two definitions *are* different and establishing this requires a separate proof, which we give in Section 4.1.

That result will justify the interpretation of $\mathbf{p}^{(t)}$, and hence of the vector $\mathbf{V}_K^{(t)}$, as an iterative approximation of the correct Kirchhoff potentials. Note that there are two sources of inaccuracy in this estimation. One is intrinsic to the iterative process, i.e., the rate with which $\mathbf{p}^{(t)}$ converges to a solution of Kirchhoff's equations; this will be the subject of Section 4.2. The second source of error is stochastic and reflects the accuracy of the estimator itself; it will be discussed in Section 4.3, where we show that for a large enough K, the estimator yields an accurate approximation of the potential with high probability, and not only in expectation.

4.1 Correctness of the token diffusion method

We can reexpress the system (11) as

$$\begin{cases} \mathbf{p}^{(0)} &= \mathbf{0}, \\ \mathbf{p}^{(t+1)} &= \underline{P}\,\mathbf{p}^{(t)} + D^{-1}\underline{\mathbf{b}}, \end{cases} \tag{13}$$

where $\underline{P}$ is obtained from $P \stackrel{\text{def}}{=} D^{-1}A$ by zeroing out all entries on the row and column corresponding to the sink node. Likewise, $\underline{\mathbf{b}}$ is obtained from $\mathbf{b}$ by zeroing out the entry corresponding to the sink node. We next prove that the spectral radius of $\underline{P}$ is *strictly* between

0 and 1. Using this fact, we prove that the token diffusion method converges to a feasible potential vector.

Lemma 4.3. *The spectral radius of $\underline{P}$, $\underline{\rho}$, satisfies $0 < \underline{\rho} < 1$. More precisely, $\underline{\rho} = 1 - \sum_{i=1}^{n} v_i \cdot P_{i,\text{sink}} / \|\mathbf{v}\|_1$, where $\mathbf{v}$ is the left Perron eigenvector of $\underline{P}$.*

Proof. First of all, observe that $\underline{P}$ is diagonalizable: if we let $\underline{A}$ be the matrix obtained from A by zeroing out the entries on the row and column corresponding to the sink node, then $\underline{P} = D^{-1}\underline{A}$, and thus $\underline{P}$ is similar to the symmetric real matrix $\underline{N} = D^{-1/2}\underline{A}D^{-1/2}$.

Moreover, $\underline{P}$ is nonnegative and the Perron-Frobenius theorem for nonnegative matrices (for example, see [18, Section 8.3]) guarantees the existence of a nonnegative *row vector* $\mathbf{v}$ such that $\mathbf{v}\underline{P} = \underline{\rho}\mathbf{v}$. Without loss of generality, assume that $\|\mathbf{v}\|_1 = 1$.

Now observe that

$$\underline{\rho} = \left|\underline{\rho}\right| = \left\|\underline{\rho}\mathbf{v}\right\|_1 = \left\|\mathbf{v}\underline{P}\right\|_1 = \sum_{i=1}^{n}\sum_{j=1}^{n} v_i \underline{P}_{ij}.$$

Let $\epsilon_{ij} \stackrel{\text{def}}{=} P_{ij} - \underline{P}_{ij} \geq 0$ be the nonnegative "gaps" between P and $\underline{P}$. Then we can continue,

$$\begin{aligned}\underline{\rho} &= \sum_{i=1}^{n}\sum_{j=1}^{n} v_i \underline{P}_{ij} = \sum_i \sum_j v_i(\underline{P}_{ij} + \epsilon_{ij} - \epsilon_{ij}) \\ &= \sum_i \sum_j v_i P_{ij} - \sum_i \sum_j v_i \epsilon_{ij} = \|\mathbf{v}P\|_1 - \sum_i v_i \sum_j \epsilon_{ij} \\ &= 1 - \mathbf{v}\cdot\boldsymbol{\gamma}, \end{aligned} \tag{14}$$

where $\boldsymbol{\gamma}$ is the vector of row gaps, i.e., $\gamma_i \stackrel{\text{def}}{=} \sum_j \epsilon_{ij}$, and for the last equality we also used the fact that every row of P sums to 1. Note that, for $i \neq \text{sink}$, γ_i is the same as $P_{i,\text{sink}}$, that is, the probability that a token at node i reaches the sink in one step.

We claim that the spectral radius is strictly positive. To see that, observe that, similarly to Fact 2.1, $\underline{P}$ is similar to $\underline{N}$ and thus shares the same eigenvalues. If the spectral radius $\underline{\rho}$ were zero, $\underline{N}$ would be a null matrix, i.e., a matrix whose entries are all zeros (this follows by looking at the diagonalized form of $\underline{N}$). Since $\underline{P}$ is similar to $\underline{N}$, $\underline{P}$ would be a null matrix as well, which is clearly not the case for any nonempty graph.

Observe also that $0 = \mathbf{v}\cdot\mathbf{0} = \mathbf{v}\underline{P}\boldsymbol{\chi}_{\text{sink}} = \underline{\rho}\mathbf{v}\cdot\boldsymbol{\chi}_{\text{sink}} = \underline{\rho}v_{\text{sink}}$ and therefore $v_{\text{sink}} = 0$. To show that $\underline{\rho} < 1$, assume by contradiction that $\underline{\rho} = 1$. Then, $v_{\text{sink}} = 0$ and (14) implies $\sum_{i\neq\text{sink}} v_i\gamma_i = \sum_i v_i\gamma_i = 0$, i.e., $v_i = 0$ whenever $\gamma_i \neq 0$; in particular, $v_i = 0$ for each node i adjacent to the sink. Continuing this argument would yield that $v_i = 0$ for each i adjacent to a node adjacent to the sink, (since the ith entry of $\mathbf{v}\underline{P}$ equals $\underline{\rho}v_i$), and so on. Since the original graph is connected, this contradicts the fact that $\mathbf{v} \neq \mathbf{0}$. □

In the proof of next theorem, we make use of the following fact, which is analogous to Fact 2.1.

Fact 4.1. The matrix $\underline{P}$ is similar to the matrix $\underline{N}$ obtained from N by zeroing out its last column and row. In particular, $\underline{P}^t = D^{-\frac{1}{2}}\underline{N}^t D^{\frac{1}{2}}$, and $\underline{N}$ has n orthonormal eigenvectors $\underline{\mathbf{x}}_1, \ldots, \underline{\mathbf{x}}_n$ which correspond to the eigenvalues $\underline{\rho}_1 \geq \ldots \geq \underline{\rho}_n$ of $\underline{P}$.

Theorem 4.4. *The iterates of* (11) *converge to a feasible solution of the linear system* (1)*. The rate of convergence is proportional to $\underline{\rho}$.*

Proof. Since $\underline{\rho} < 1$, the matrix $I - \underline{P}$ is invertible, and its inverse can be expressed as

$$(I - \underline{P})^{-1} = \sum_{k=0}^{\infty} \underline{P}^k.$$

If we recursively expand the updates $\mathbf{p}^{(t+1)} = \underline{P}\mathbf{p}^{(t)} + D^{-1}\underline{\mathbf{b}}$, we get, for any $t \geq 1$,

$$\mathbf{p}^{(t)} = \underline{P}^t\mathbf{p}^{(0)} + \sum_{k=0}^{t-1} \underline{P}^k D^{-1}\underline{\mathbf{b}} = \left(\sum_{k=0}^{t-1} \underline{P}^k\right) D^{-1}\underline{\mathbf{b}}, \tag{15}$$

where we used that in (13) $\mathbf{p}^{(0)} = 0$. As $t \to \infty$, this yields

$$\mathbf{p}^{(\infty)} \stackrel{\text{def}}{=} \lim_{t\to\infty} \mathbf{p}^{(t)} = (I - \underline{P})^{-1} D^{-1}\underline{\mathbf{b}}, \tag{16}$$

which shows that in the limit, the iterates satisfy the linear system $(I - \underline{P})\mathbf{p}^{(\infty)} = D^{-1}\underline{\mathbf{b}}$, or, recalling that $\underline{P} \stackrel{\text{def}}{=} D^{-1}\underline{A}$,

$$(D - \underline{A})\mathbf{p}^{(\infty)} = \underline{\mathbf{b}}. \tag{17}$$

To conclude that $\mathbf{p}^{(\infty)}$ also satisfies the original system (1), notice the following. The two matrices $L = D - A$ and $D - \underline{A}$, as well as the two vectors $\mathbf{b}$ and $\underline{\mathbf{b}}$, differ only in the row corresponding to the sink node, and the difference of the *sink*-th rows of the two matrices is given by the row vector

$$\boldsymbol{\chi}_{\text{sink}}^\top((D - \underline{A}) - L) = \mathbf{1}^\top(A - \underline{A}) = \mathbf{1}^\top(D - \underline{A}),$$

where in the last equality we used $\mathbf{1}^\top(D - A) = \mathbf{0}^\top$. Therefore, using (17),

$$\begin{aligned}&\boldsymbol{\chi}_{\text{sink}}^\top(D - \underline{A})\mathbf{p}^{(\infty)} - \boldsymbol{\chi}_{\text{sink}}^\top(D - A)\mathbf{p}^{(\infty)} \\ &\qquad = \mathbf{1}^\top(D - \underline{A})\mathbf{p}^{(\infty)} = \mathbf{1}^\top\underline{\mathbf{b}} = 1 = \chi_{sink}^\top(\underline{\mathbf{b}} - \mathbf{b}),\end{aligned}$$

which, after rearranging terms, together with (17) implies that

$$0 = \boldsymbol{\chi}_{\text{sink}}^\top((D - \underline{A})\mathbf{p}^{(\infty)} - \underline{\mathbf{b}}) = \boldsymbol{\chi}_{\text{sink}}^\top((D - A)\mathbf{p}^{(\infty)} - \mathbf{b}).$$

This proves that $(D - A)\mathbf{p}^{(\infty)} = L\mathbf{p}^{(\infty)} = \mathbf{b}$.

The second part of the theorem follows from Fact 4.1, (15) and (16), which yield

$$\begin{aligned}\|\mathbf{p}^{(t)} - \mathbf{p}^{(\infty)}\| &= \left\|\left(\sum_{k=t}^{\infty} \underline{P}^k\right) D^{-1}\underline{\mathbf{b}}\right\| \leq \sum_{k=t}^{\infty} \left\|\underline{P}^k D^{-1}\underline{\mathbf{b}}\right\| \\ &\stackrel{(a)}{\leq} \sqrt{\frac{\text{vol}_{\max}}{\text{vol}_{\min}}} \frac{\sum_{k=t}^{\infty} \underline{\rho}^k}{\text{vol}(source)} = \sqrt{\frac{\text{vol}_{\max}}{\text{vol}_{\min}}} \frac{\underline{\rho}^t}{(1-\underline{\rho})} \frac{1}{\text{vol}(source)},\end{aligned}$$

where in (a) we performed a calculation analogous to (8). □

4.2 Convergence rate of token diffusion

In Section 4.1 we showed that the token diffusion system converges to one of the solutions of Kirchhoff's equations. Moreover, its rate of convergence is dictated by the spectral radius of the transition matrix $\underline{P} = D^{-1}\underline{A}$ (Theorem 4.4), which is similar to the original transition matrix P, except for the fact that all entries of the row and column corresponding to the sink are equal to 0 in $\underline{P}$.

For simplicity of exposition, in the remainder we simply remove the row and column of $\underline{P}$ corresponding to the sink. Assume without loss of generality that the sink corresponds to the n-th row/column index. Given an $n \times n$ matrix M, consider the $(n-1) \times (n-1)$ matrix $\mathrm{gr}(M)$, obtained from M by *grounding* the n-th index, that is, removing the n-th row and n-th column. The next fact shows that this operation does not affect the spectral radius of $\underline{P}$.

Proposition 4.5. *Assume M is an $n \times n$ matrix where each entry of the n-th row and of the n-th column is zero. Then:*

1. *for every eigenpair $((x_1, \ldots, x_{n-1}), \mu)$ of $\mathrm{gr}(M)$ there is an eigenpair $((x_1, \ldots, x_{n-1}, 0), \mu)$ of M;*
2. *$(\chi_n, 0)$ is an eigenpair of M;*
3. *the spectral radius of M and $\mathrm{gr}(M)$ is the same.*

Proof. Point (1) follows from the assumption that the n-th column of M is identically zero and thus the i-th entry of $M(x_1, \ldots, x_{n-1}, 0)$ is equal to the i-th entry of $\mathrm{gr}(M)(x_1, \ldots, x_{n-1})$ for any $i < n$. Point (2) follows from $M\chi_n = \mathbf{0}$. Point (3) is a direct consequence of the first two. □

Since $\underline{P}$ satisfies the hypothesis of Proposition 4.5, we can equivalently study the spectral radius of $\mathrm{gr}(\underline{P})$ (= $\mathrm{gr}(P)$). To simplify (and with a slight abuse of) notation, in the remainder of this section we write $\underline{M}$ for $\mathrm{gr}(M)$.

Additional notation. We denote by $\overline{\mathcal{G}}$ the graph obtained from $\mathcal{G}$ by removing the sink node and its incident edges. We denote by $\overline{L}$ the Laplacian matrix of $\overline{\mathcal{G}}$, so that $\overline{L} = \overline{D} - \overline{A}$, with $\overline{A}$ and $\overline{D}$ respectively the adjacency and degree matrices of $\overline{\mathcal{G}}$. We also define $\underline{L} \stackrel{\text{def}}{=} \underline{D} - \underline{A}$. Note that $(\overline{A})_{ij} = (\underline{A})_{ij}$ for each $i, j < n$. On the other hand $\underline{L}$ is not a proper graph Laplacian, since $(\overline{D})_{ii} \neq (\underline{D})_{ii}$ for some $i < n$.[3] However, $\underline{L}$ can be viewed as a *perturbed* Laplacian, since $\underline{D} = \overline{D} + \Delta$, where $\Delta = \mathrm{diag}(w_{1,n}, \ldots, w_{n-1,n})$.

The rate of convergence of the token diffusion process is dictated by $\underline{\rho}$, the dominant eigenvalue of the matrix $\underline{P} \stackrel{\text{def}}{=} D^{-1}\underline{A} = \underline{D}^{-1}\underline{A}$. Thanks to Fact 4.1, we can equivalently study the matrix $\underline{N} = \underline{D}^{-1/2}\,\underline{A}\,\underline{D}^{-1/2}$, which shares the same spectrum as $\underline{P}$, or, equivalently, the matrix $\underline{\mathcal{L}} = I - \underline{N} = \underline{D}^{-1/2}\underline{L}\underline{D}^{-1/2} = \underline{D}^{-1/2}(\overline{L} + \Delta)\underline{D}^{-1/2}$. Again, the matrix $\underline{\mathcal{L}}$ can be interpreted as a perturbed normalized Laplacian. The eigenvalue $\underline{\rho}$ of $\underline{N}$ corresponds to an eigenvalue $\underline{\lambda} = 1 - \underline{\rho}$ of $\underline{\mathcal{L}}$.

In this section we provide a lower bound on $\underline{\lambda}$. Let $\mathbf{x}$ denote the (unit norm) eigenvector of $\underline{\mathcal{L}}$ corresponding to $\underline{\lambda}$ and let $\mathbf{y} = \underline{D}^{-1/2}\mathbf{x}$. Since $\underline{\mathcal{L}}$ is symmetric, we have by definition:

$$\underline{\lambda} = \mathbf{x}^\top \underline{\mathcal{L}}\mathbf{x} = \mathbf{x}^\top \underline{D}^{-1/2}\underline{L}\underline{D}^{-1/2}\mathbf{x} = \mathbf{x}^\top \underline{D}^{-1/2}(\overline{L} + \Delta)\underline{D}^{-1/2}\mathbf{x} = \mathbf{y}^\top(\overline{L} + \Delta)\mathbf{y}, \tag{18}$$

Proposition 4.6. *The following holds:*

$$\frac{1}{\mathrm{vol}_{\max}} \leq \|\mathbf{y}\|^2 \leq \frac{1}{\mathrm{vol}_{\min}}. \tag{19}$$

[3] Precisely, this happens whenever $i \sim n$ in the original graph $\mathcal{G}$.

Proof. We have:

$$\|\mathbf{y}\|^2 = \|\underline{D}^{-1/2}\mathbf{x}\|^2 = \sum_{i=1}^{n-1} \frac{\mathbf{x}_i^2}{\mathrm{vol}(i)}.$$

The claim then follows immediately since $\|\mathbf{x}\|^2 = 1$. □

In the remainder, we decompose $\mathbf{y}$ as $\mathbf{y} = \mathbf{u} + \mathbf{z}$, with $\mathbf{u}$ and $\mathbf{z}$ the components of $\mathbf{y}$ respectively parallel and orthogonal to the vector $\mathbf{1}$. The next fact highlights a general property of the perturbed Laplacian matrix that affords a simplification of (18). The proof is deferred to the full version [2].

Proposition 4.7. *For any $\mathbf{y} \in \mathbb{R}^{n-1}$, it holds $\mathbf{y}^\top\overline{L}\mathbf{y} = \mathbf{z}^\top\overline{L}\mathbf{z}$, where $\mathbf{z} = \mathbf{y} - ((\mathbf{1}^\top\mathbf{y})/(\mathbf{1}^\top\mathbf{1})) \cdot \mathbf{1}$ is the component of $\mathbf{y}$ orthogonal to $\mathbf{1}$.*

We can now give a lower bound on $\underline{\lambda}$, in terms of $\mathbf{y}$ and $\mathbf{z}$.

Lemma 4.8. *The following holds:*

$$\underline{\lambda} \geq \mathbf{y}^\top\Delta\mathbf{y} + \overline{\lambda}_2\|\mathbf{z}\|^2, \tag{20}$$

where $\overline{\lambda}_2$ is the second smallest eigenvalue of $\overline{L}$, the Laplacian of the graph $\overline{\mathcal{G}}$.

Proof. From (18) and Proposition 4.7, we have

$$\underline{\lambda} = \mathbf{y}^\top(\overline{L} + \Delta)\mathbf{y} = \mathbf{y}^\top\Delta\mathbf{y} + \mathbf{z}^\top\overline{L}\mathbf{z} = \mathbf{y}^\top\Delta\mathbf{y} + \frac{\mathbf{z}^\top\overline{L}\mathbf{z}}{\|\mathbf{z}\|^2}\|\mathbf{z}\|^2$$
$$\geq \mathbf{y}^\top\Delta\mathbf{y} + \|\mathbf{z}\|^2 \min_{\substack{\mathbf{w}\perp\mathbf{1} \\ \|\mathbf{w}\|=1}} \mathbf{w}^\top\overline{L}\mathbf{w} = \mathbf{y}^\top\Delta\mathbf{y} + \overline{\lambda}_2\|\mathbf{z}\|^2,$$

where the inequality follows by recalling that $\mathbf{z} \perp \mathbf{1}$ by definition, and by observing that the second term of the sum is the Rayleigh quotient associated to $\overline{L}$, multiplied by $\|\mathbf{z}^2\|$. □

We can now prove the main result of this section.

Theorem 4.9. *The following holds:*

$$\underline{\lambda} \geq \frac{\overline{\lambda}_2}{2\mathrm{vol}_{\max}(n-1)} \sum_i \frac{w_{in}}{w_{in} + \overline{\lambda}_2}.$$

Proof. Since $\mathbf{u}$ is the component of $\mathbf{y}$ parallel to $\mathbf{1}$, we have $\mathbf{u}_i^2 = u^2/(n-1)$ for some $u \in \mathbb{R}$. Given $\mathbf{u}$, we compute the vector $\mathbf{z}$ that minimizes the right-hand side of (20). Let $f(\mathbf{z}) = (\mathbf{u}+\mathbf{z})^\top\Delta(\mathbf{u}+\mathbf{z}) + \overline{\lambda}_2\|\mathbf{z}\|^2$, where $\mathbf{u}$ is regarded as a constant. We have:

$$\frac{\partial f}{\partial \mathbf{z}_i} = 2w_{in}(\mathbf{u}_i + \mathbf{z}_i) + 2\overline{\lambda}_2\mathbf{z}_i,$$
$$\frac{\partial^2 f}{\partial \mathbf{z}_i \partial \mathbf{z}_j} = 0,\ i \neq j,$$
$$\frac{\partial^2 f}{\partial \mathbf{z}_i^2} = 2w_{in} + 2\overline{\lambda}_2.$$

Since $w_{in}, \overline{\lambda}_2 > 0$ (the latter following since $\overline{\lambda}_2$ is the second eigenvalue of a Laplacian matrix), the determinant of the Hessian matrix is positive, hence $f(\mathbf{z})$ has a global minimum that is the critical point. If we set the i-th first-order partial derivative to 0 we obtain $\mathbf{z}_i = -w_{in}\mathbf{u}_i/(w_{in} + \overline{\lambda}_2)$. Substituting back into $f(\mathbf{z})$ yields:

$$f(\mathbf{z}) = \sum_i w_{in}\left(\mathbf{u}_i - \frac{w_{in}\mathbf{u}_i}{w_{in} + \overline{\lambda}_2}\right)^2 + \overline{\lambda}_2 \sum_i \left(\frac{w_{in}\mathbf{u}_i}{w_{in} + \overline{\lambda}_2}\right)^2$$

$$= \overline{\lambda}_2^2 \sum_i \frac{w_{in}\mathbf{u}_i^2}{(w_{in}+\overline{\lambda}_2)^2} + \overline{\lambda}_2 \sum_i \left(\frac{w_{in}\mathbf{u}_i}{w_{in}+\overline{\lambda}_2}\right)^2$$
$$= \frac{\overline{\lambda}_2 u^2}{n-1} \sum_i \frac{w_{in}}{w_{in}+\overline{\lambda}_2} \tag{21}$$

Next, recall that $\|\mathbf{u}\|^2 + \|\mathbf{z}\|^2 \geq 1/\mathrm{vol}_{\max}$ from Proposition 4.6. We consider two cases. If $\|\mathbf{z}\|^2 \geq 1/(2\mathrm{vol}_{\max})$, then we have:

$$\underline{\lambda} \geq \frac{\overline{\lambda}_2}{2\mathrm{vol}_{\max}}. \tag{22}$$

Otherwise, $u^2 \geq 1/(2\mathrm{vol}_{\max})$, and (21) immediately gives

$$\underline{\lambda} \geq \frac{\overline{\lambda}_2}{2\mathrm{vol}_{\max}(n-1)} \sum_i \frac{w_{in}}{w_{in}+\overline{\lambda}_2}. \tag{23}$$

To conclude the proof, note that the latter bound is always the worse case, since $w_{in}/(w_{in}+\overline{\lambda}_2) \leq 1$. □

4.3 Stochastic accuracy of the estimator

In this subsection, we study how closely the estimator $V_K^{(t)}(u)$ approximates its expected value, the potential $p_u^{(t)}$. In a nutshell, we show the following: i) the larger K (the independent parameter controlling the number of tokens injected per round), the higher the accuracy; ii) all the rest being equal, the higher the potential, the higher the accuracy.

Notation. Starting at $t = 0$, we index tokens in increasing order of their release dates. In more detail, any token released in the $(i+1)$-th round has an index in the interval $\{iK+1, \ldots, (i+1)K\}$, with $i = 0, 1, \ldots$, while the relative order of tokens released in the same round is irrelevant and arbitrary. The main result of this subsection is the following.

Theorem 4.10. *For any given K, $0 < \epsilon, \delta < 1$, for every t and for every u, such that $p_u^{(t)} \geq \frac{3}{\epsilon^2 K \mathrm{vol}(u)} \ln \frac{2}{\delta}$, Algorithms 2 and 3 together provide an (ϵ, δ)-approximation of $p_u^{(t)}$.*[4]

Remark 1. Given the statistical and node-wise nature of the counter estimator, there is a "resolution" limit for the minimum value of a potential that can be estimated with desired accuracy and confidence levels for a specific value of K. This is a consequence of the law of large numbers (applied in the form of a Chernoff bound in our case). A similar issue would arise if we used a different estimator, e.g., one based on Tetali and Snell's approach. On the other hand, accuracy and confidence can be improved by increasing K. This leads to an equivalent way of expressing Theorem 4.10 in which, given the minimum potential value we want to estimate with given accuracy and confidence levels, we can compute the minimum K that achieves the desired performance. More formally, an (ϵ, δ)-approximation of the potentials $p_u^{(t)}$ greater than $p_\star^{(t)}$ can be achieved by setting $K \geq \frac{3}{\epsilon^2 p_\star^{(t)} \mathrm{vol}(u)} \ln \frac{2}{\delta}$.

Proof of Theorem 4.10. Let $X_j^{(t)}(u) = 1$ if the j-th token is at node u at time t, $X_j^{(t)}(u) = 0$ otherwise. From Corollary 4.2 we have:

$$K \mathrm{vol}(u)\, p_u^{(t)} = \mathbb{E}\left[Z_K^{(t)}(u)\right] = \mathbb{E}\left[\sum_{j=1}^{Kt} X_j^{(t)}(u)\right].$$

The $X^{(t)}(u)$'s are independent Bernoulli variables and the expectation of their sum is $K\mathrm{vol}(u)p_u^{(t)}$. Hence, a simple application of the multiplicative Chernoff bound yields

$$\Pr\left[\left|Z_K^{(t)} - K\mathrm{vol}(u)p^{(t)}(u)\right| > \epsilon K \mathrm{vol}(u) p^{(t)}(u)\right] \leq$$
$$\leq 2e^{-\frac{\epsilon^2}{3} K \mathrm{vol}(u) p^{(t)}(u)} \leq \delta,$$

whenever $p_u^{(t)} \geq \frac{3}{\epsilon^2 K \mathrm{vol}(u)} \ln \frac{2}{\delta}$. Finally, note that

$$\left|Z_K^{(t)} - K\mathrm{vol}(u)p_u^{(t)}\right| \leq \epsilon K \mathrm{vol}(u) p_u^{(t)} \iff \left|V_K^{(t)}(u) - p_u^{(t)}\right| \leq \epsilon p_u^{(t)}.$$

directly from definitions. This completes the proof. □

4.4 Time and message complexity

The arguments from previous sections lead to the following conclusions about the token diffusion process, whose proof is deferred to the full version [2].

Theorem 4.11. *The expected value of the estimator vector $\mathbf{V}_K^{(t)}$ constructed by Algorithms 2 and 3 converges to the grounded solution $\mathbf{p}$ of the Kirchhoff equations at a rate*

$$O((1-\underline{\lambda})^t) = O\left(\left(1 - \frac{\overline{\lambda}_2}{2n\,\mathrm{vol}_{\max}} \sum_{u \in \mathcal{V}, u \neq n} \frac{w_{un}}{w_{un}+\overline{\lambda}_2}\right)^t\right). \tag{24}$$

Note that the right hand side in (24) is *decreasing* with $\overline{\lambda}_2$. Thus, any lower bound on $\overline{\lambda}_2$ yields an upper bound on the right hand side in (24). By recalling that $\overline{\lambda}_2$ is the second smallest eigenvalue of the graph $\overline{\mathcal{G}}$, this allows to connect the error term in (24) to the *edge expansion* of $\overline{\mathcal{G}}$, since for any graph $\mathcal{G}$ it is known [3, Theorem 2.2] that

$$\lambda_2(\mathcal{G}) \geq \mathrm{vol}_{\max} - (\mathrm{vol}_{\max}^2 - \theta(\mathcal{G})^2)^{1/2}.$$

Thus, the higher the edge expansion of $\overline{\mathcal{G}}$, the higher $\overline{\lambda}_2$, and the faster the convergence of the token diffusion process.

We finally derive a bound on the expected message complexity of Algorithm 2. The proof is again deferred to the full version [2].

Proposition 4.12. *As $t \to \infty$, the expected message complexity per round of Algorithm 2 is $O(K\, n\, \mathrm{vol}_{\max} \cdot E)$, where $E \stackrel{\text{def}}{=} \mathbf{p}^\top L \mathbf{p}$.*

5 BEYOND POTENTIALS: AN OUTLOOK

Our results show that the effectiveness of decentralized, simple processes for electrical flow computation can be quantitatively analyzed, which is a step forward in the microscopic-level analysis of social, biological and artificial systems that can be described in terms of time-varying resistive networks or as current-reinforced random walks.

On the other hand, while opinion dynamics were originally proposed as elementary models of information exchange and manipulation in social networks, our results highlight their potential as versatile and powerful primitives for collective computing. We believe this is a perspective that deserves further investigation.

[4] A random variable X gives an (ϵ, δ)-approximation of a non-negative quantity Y if $\mathbf{P}(|X - Y| > \epsilon Y) \leq \delta$.

REFERENCES

[1] Daron Acemoglu and Asuman Ozdaglar. 2011. Opinion dynamics and learning in social networks. *Dynamic Games and Applications* 1, 1 (2011), 3–49.

[2] Luca Becchetti, Vincenzo Bonifaci, and Emanuele Natale. 2018. Pooling or Sampling: Collective Dynamics for Electrical Flow Estimation. arXiv:1804.06127. (2018). http://arxiv.org/abs/1804.06127

[3] Abraham Berman and Xiao-Dong Zhang. 2000. Lower bounds for the eigenvalues of Laplacian matrices. *Linear Algebra Appl.* 316 (2000), 13–20.

[4] Vincenzo Bonifaci, Kurt Mehlhorn, and Girish Varma. 2012. Physarum can compute shortest paths. *Journal of Theoretical Biology* 309 (2012), 121–133.

[5] Ashok K. Chandra, Prabhakar Raghavan, Walter L. Ruzzo, Roman Smolensky, and Prasoon Tiwari. 1997. The electrical resistance of a graph captures its commute and cover times. *Computational Complexity* 6, 4 (1997), 312–340.

[6] Paul Christiano, Jonathan A. Kelner, Aleksander Madry, Daniel A. Spielman, and Shang-Hua Teng. 2011. Electrical flows, Laplacian systems, and faster approximation of maximum flow in undirected graphs. In *Proceedings of the 43rd ACM Symposium on Theory of Computing*. ACM, 273–282.

[7] Morris H. DeGroot. 1974. Reaching a consensus. *J. Amer. Statist. Assoc.* 69, 345 (1974), 118–121.

[8] Peter G. Doyle and J. Laurie Snell. 1984. *Random Walks and Electric Networks*. Mathematical Association of America.

[9] Nelson Elhage and Jacob Beal. 2010. Laplacian-based consensus on spatial computers. In *9th International Conference on Autonomous Agents and Multiagent Systems*. ACM, 907–914.

[10] Daniel R. Figueiredo and Michele Garetto. 2017. On the Emergence of Shortest Paths by Reinforced Random Walks. *IEEE Trans. Network Science and Engineering* 4, 1 (2017), 55–69.

[11] Ryusuke Fujisawa, Shigeto Dobata, Ken Sugawara, and Fumitoshi Matsuno. 2014. Designing pheromone communication in swarm robotics: Group foraging behavior mediated by chemical substance. *Swarm Intelligence* 8, 3 (Sept. 2014), 227–246. https://link.springer.com/article/10.1007/s11721-014-0097-z

[12] Jonathan A. Kelner and Aleksander Madry. 2009. Faster Generation of Random Spanning Trees. In *50th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, 13–21.

[13] Brian J. Kirby. 2013. *Micro- and Nanoscale Fluid Mechanics* (reprint ed.). Cambridge University Press.

[14] Yin Tat Lee, Satish Rao, and Nikhil Srivastava. 2013. A new approach to computing maximum flows using electrical flows. In *Symposium on Theory of Computing Conference*. ACM, 755–764.

[15] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. 2009. *Markov Chains and Mixing Times*. American Mathematical Society.

[16] Laszlo Lovász. 1996. Random walks on graphs: a survey. In *Combinatorics, Paul Erdös is Eighty*, D. Miklós, V. T. Sós, and T. Szönyi (Eds.). Vol. 2. János Bolyai Mathematical Society, 353–398.

[17] Qi Ma, Anders Johansson, Atsushi Tero, Toshiyuki Nakagaki, and David JT Sumpter. 2013. Current-reinforced random walks for constructing transport networks. *Journal of the Royal Society Interface* 10, 80 (2013), 20120864.

[18] Carl D. Meyer. 2000. *Matrix Analysis and Applied Linear Algebra*. SIAM.

[19] Pragnesh Jay Modi, Wei-Min Shen, Milind Tambe, and Makoto Yokoo. 2003. An asynchronous complete method for distributed constraint optimization. In *2nd International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM, 161–168.

[20] Elchanan Mossel, Joe Neeman, and Omer Tamuz. 2014. Majority dynamics and aggregation of information in social networks. *Autonomous Agents and Multi-Agent Systems* 28, 3 (2014), 408–429.

[21] R. Andrew Russell. 1999. Ant trails - an example for robots to follow?. In *Proceedings 1999 IEEE International Conference on Robotics and Automation*, Vol. 4. 2698–2703 vol.4.

[22] Yousef Saad. 2003. *Iterative Methods for Sparse Linear Systems*. SIAM.

[23] Amirali Salehi-Abari and Craig Boutilier. 2014. Empathetic Social Choice on Social Networks. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems (AAMAS '14)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 693–700.

[24] Daniel A. Spielman and Nikhil Srivastava. 2011. Graph sparsification by effective resistances. *SIAM J. Comput.* 40, 6 (2011), 1913–1926.

[25] Damian Straszak and Nisheeth K. Vishnoi. 2016. Natural Algorithms for Flow Problems. In *Proc. of the 27th ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 1868–1883.

[26] Atsushi Tero, Ryo Kobayashi, and Toshiyuki Nakagaki. 2007. A mathematical model for adaptive transport network in path finding by true slime mold. *Journal of Theoretical Biology* 244 (2007), 553–564.

[27] Prasad Tetali. 1991. Random walks and the effective resistance of networks. *Journal of Theoretical Probability* 4, 1 (1991), 101–109.

[28] Kwok Ching Tsui and Jiming Liu. 2003. Multiagent diffusion and distributed optimization. In *2nd International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM, 169–176.

[29] Richard S. Varga. 2009. *Matrix Iterative Analysis*. Springer.

[30] Nisheeth K. Vishnoi. 2013. Lx = b: Laplacian solvers and their algorithmic applications. *Foundations and Trends in Theoretical Computer Science* 8, 1-2 (2013), 1–141.