Pooling or Sampling: Collective Dynamics for Electrical Flow Estimation

Emanuele Natale[◇,*]
joint work with L. Becchetti[†] and V. Bonifaci[⋆]

† SAPIENZA UNIVERSITÀ DI ROMA

⋆ National Research Council of Italy

* Berkeley UNIVERSITY OF CALIFORNIA

◇ max planck institut informatik

INSTITUT DE RECHERCHE EN INFORMATIQUE FONDAMENTALE

IRIF
4 May 2018
Paris

Electrical Networks for Optimization

Computation of currents and voltages in resistive electrical network is a crucial primitive in many optimization algorithms

- Maximum flow
  - Christiano, Kelner, Madry, Spielman and Teng, STOC'11
  - Lee, Rao and Srivastava, STOC'13
- Network sparsification
  - Spielman and Srivastava, SIAM J. of Comp. 2011
- Generating spanning trees
  - Kelner and Madry, FOCS'09

Electrical Networks for Optimization

Computation of currents and voltages in resistive electrical network is a crucial primitive in many optimization algorithms

- Maximum flow
  - Christiano, Kelner, Madry, Spielman and Teng, STOC'11
  - Lee, Rao and Srivastava, STOC'13
- Network sparsification
  - Spielman and Srivastava, SIAM J. of Comp. 2011
- Generating spanning trees
  - Kelner and Madry, FOCS'09

and as model of biological computation

- Physarum polycephalum
- Ants

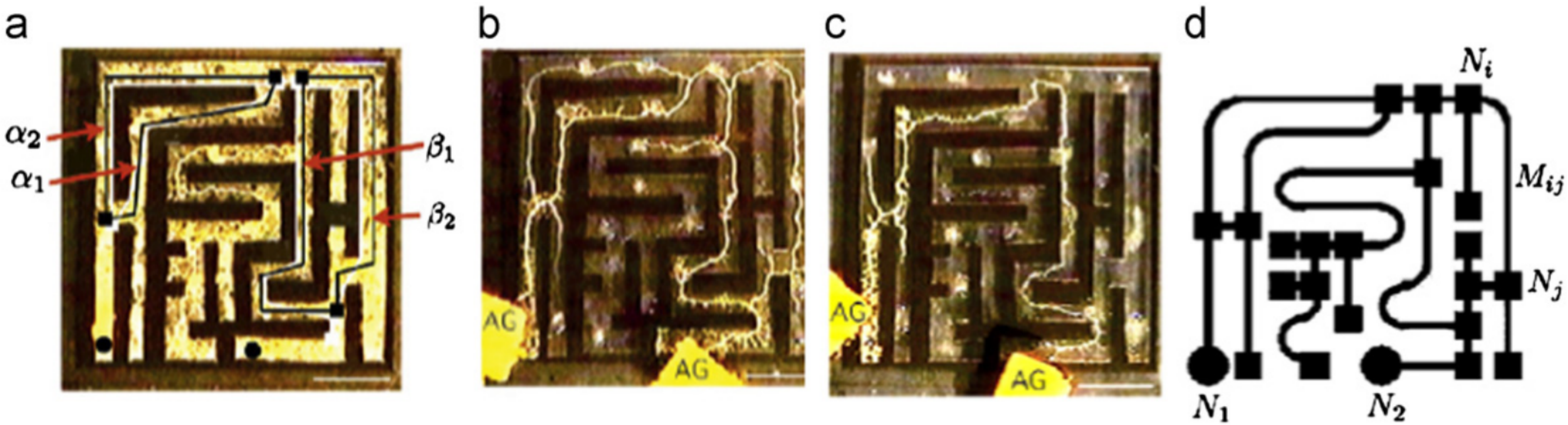
implicitly solving electrical flow while forming food-transportation networks

The Slime Mold (Physarum Polycephalum)

[Video Interlude]

The Slime Mold (Physarum Polycephalum)

Nakagaki, Yamada and Toth, Nature 2000
Tero, Kobayashi and Nakagaki J. of Theo. Bio. 2007



For each edge e: ℓ_e length, x_e diameter, $r_e = \ell_e / x_e$ resistance, q_e current

Ohm's law: $q_{(u,v)} = (p_u - p_v)/r_e$
Flow conservation: $\sum_{v \sim u} q_{(u,v)} = b(u)$
$b(u)$ 1 on source, -1 on sink, 0 o/w

Dynamics:
$\dot{x}_e = |q_e| - x_e$

Physarum Dynamics as an Algorithm

Bonifaci, Mehlhorn and Varma SODA'12:
Physarum dynamics converges on all graphs
(elegant proof in Bonifaci IPL'13)

Physarum Dynamics as an Algorithm

Bonifaci, Mehlhorn and Varma SODA'12:
Physarum dynamics converges on all graphs
(elegant proof in Bonifaci IPL'13)

Euler's discretization

$$x(t+1) - x(t) = h(|q(t)| - x(t))$$

Becchetti, Bonifaci, Dirnberger, Karrenbauer and Mehlhorn ICALP'13:
Discretized physarum computes $(1+\epsilon)$-apx.
in $\mathcal{O}(mL(\log n + \log L)/\epsilon^3)$

More Research on Physarum

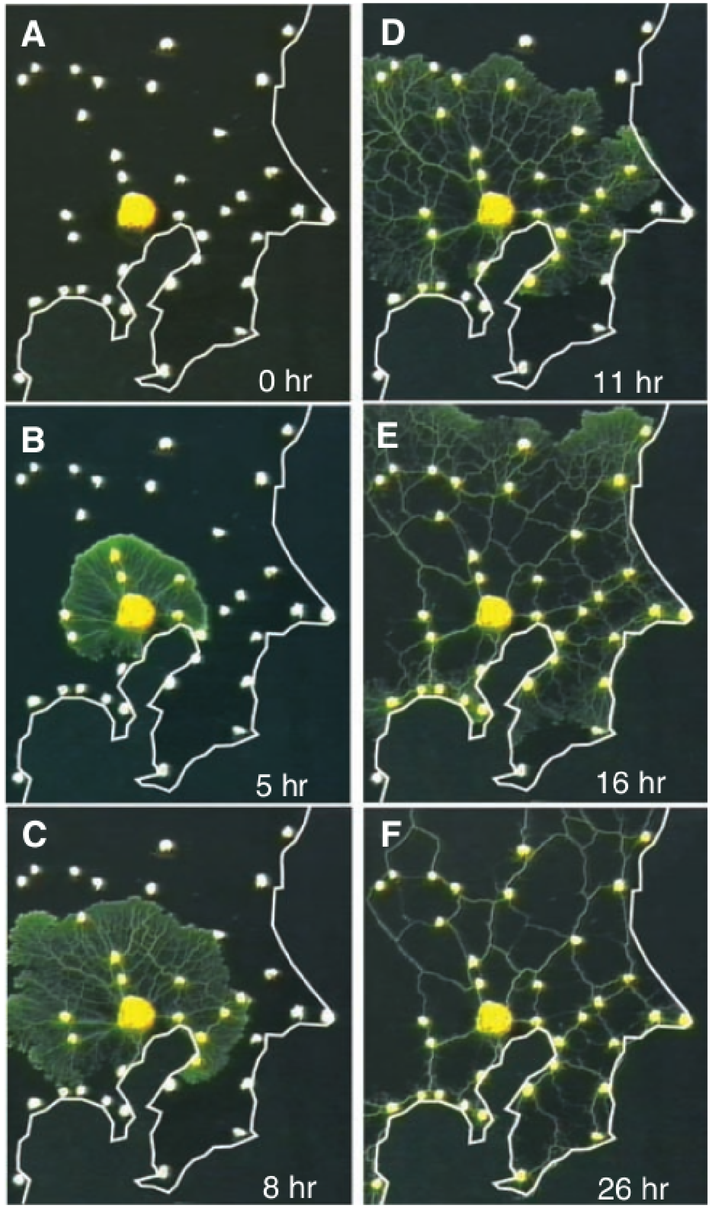
Many sequels in TCS

- Bonifaci IPL'13,
- Straszak and Vishnoi ITCS'16,
- Straszak and Vishnoi SODA'16
- Becker et al. ESA'17
- ...

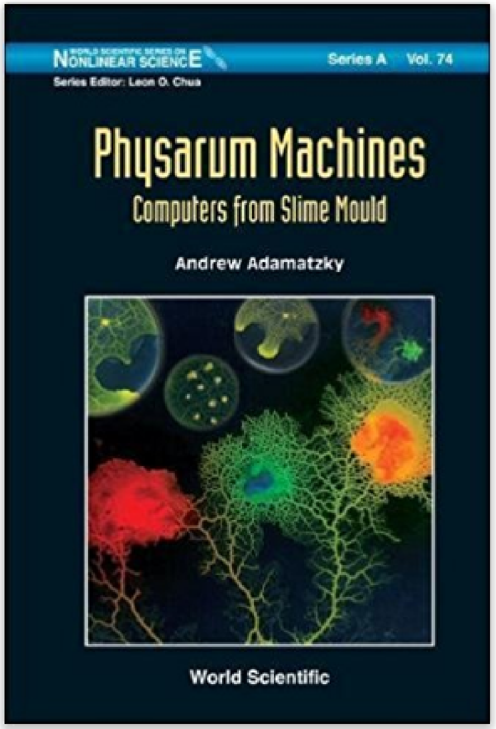
More Research on Physarum

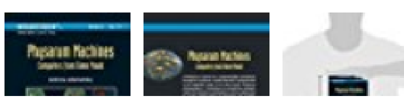
Many sequels in TCS
- Bonifaci IPL'13,
- Straszak and Vishnoi ITCS'16,
- Straszak and Vishnoi SODA'16
- Becker et al. ESA'17
- ...

Some sequels elsewhere...
Tero et al. Science 2010: Physarum re-builds Tokyo's rail network!



More Research on Physarum





See all 3 images

Physarum Machines: Computers from Slime Mould (World Scientifi Nonlinear Science, Series A) Hardcover – August 26, 2010

by Andrew Adamatzky (Author)

Be the first to review this item

See all 2 formats and editions

Hardcover	Paperback
$60.82	$104.00
30 Used from $62.10	5 Used from $99.66
34 New from $60.82	4 New from $99.66

A Physarum machine is a programmable amorphous biological computer experimentally implemented in the vegetative state of true slime mould Physarum polycephalum. It comprises an amorphous yellowish mass with networks of protoplasmic veins, programmed by spatial configurations of attracting and repelling gradients.

This book demonstrates how to create experimental Physarum machines for computational geometry and optimization, distributed manipulation and transportation, and general-purpose computation. Being very cheap to make and easy to maintain, the machine also functions on a wide range of substrates and in a broad scope of environmental conditions. As such a Physarum machine is a green and environmentally friendly unconventional computer.

The book is readily accessible to a nonprofessional reader, and is a priceless source of experimental tips and inventive theoretical ideas for anyone who is inspired by novel and emerging non-silicon computers and robots.

Read less

Electrical Networks as Biological Models?...

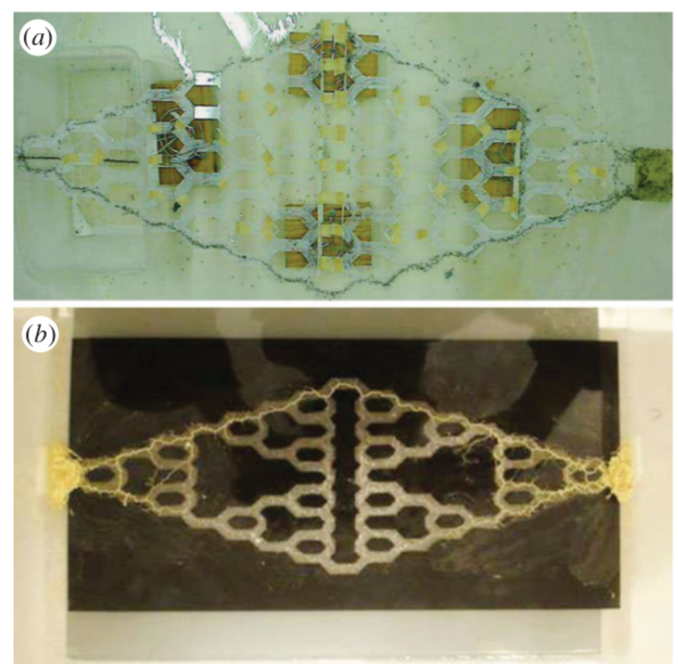
hydraulic	electric
volume V [m^3]	charge q [C]
pressure p [Pa=J/m^3=N/m^2]	potential ϕ [V=J/C=W/A]
Volumetric flow rate Φ_V [m^3/s]	current I [A=C/s]
velocity v [m/s]	current density j [C/(m^2·s) = A/m^2]
Poiseuille's law $\Phi_V = \frac{\pi r^4}{8\eta}\frac{\Delta p^\star}{\ell}$	Ohm's law $j = -\sigma\nabla\phi$

(stolen from Wikipedia/Hydraulic_analogy)

From Slime Molds to Ants

electric network	*Physarum*	ant trails
length in space	length in space	length in space
potential/voltage	amount of nutrient	number of ants
current	flow of nutrient	flow of ants
conductivity	thickness of tube	pheromone concentration
capacitance	transport efficiency	total pheromone density
reinforcement intensity	tube expansion rate	pheromone drop rate
conductivity decrease rate	tube decay rate	evaporation rate

Ma, Johansson, Tero, Nakagaki and Sumpter, J. of the Royal Society Interface '13



How to Compute with Electrical Networks

Physarum have to solve Kirchhoff's equations

$\sum_{v \sim u} q_{(u,v)} = \sum_{v \sim u} (p_u - p_v)/r_e = b(u)$

or $\boxed{Lp = b}$ with $L = BCB^{\mathsf{T}} = D - A$

$C = diag(x_e/\ell_e)$, B signed incidence

How to Compute with Electrical Networks

Physarum have to solve Kirchhoff's equations

$\sum_{v \sim u} q_{(u,v)} = \sum_{v \sim u} (p_u - p_v)/r_e = b(u)$

or $\boxed{Lp = b}$ with $L = BCB^\intercal = D - A$

$C = diag(x_e/\ell_e)$, B signed incidence

Previous approaches: centralized computation!

- Can be accomplished if every node is agent that follows elementary protocol? (*biologically*: what happens microscopically?)
- If yes, what is convergence time and communication overhead?

Distributed Jacobi's Method

Jacobi's iterative method (Varga, 2009):
Bound on convergence rate w.r.t. *graph conductance*
exploiting structure of laplacian
(cfr. also DeGroot's model)

Distributed Jacobi's Method

Jacobi's iterative method (Varga, 2009):
Bound on convergence rate w.r.t. *graph conductance*
exploiting structure of laplacian
(cfr. also DeGroot's model)

$$Lp = (D - A)p = b \implies p = D^{-1}(Ap + b)$$

Jacobi's:
$\tilde{p}_u(t+1) = (\sum_{v \sim u} c_{(u,v)} \tilde{p}(t) + b_u)/vol(u)$

Distributed Jacobi's Method

Jacobi's iterative method (Varga, 2009):
Bound on convergence rate w.r.t. *graph conductance*
exploiting structure of laplacian
(cfr. also DeGroot's model)

$$Lp = (D - A)p = b \implies p = D^{-1}(Ap + b)$$

Jacobi's:
$\tilde{p}_u(t+1) = (\sum_{v \sim u} c_{(u,v)} \tilde{p}(t) + b_u)/vol(u)$

Error $e(t) = p - \tilde{p}(t) = e_{\perp}(t) + \alpha 1$
(p *doesn't care about* α)

Jacobi's Convergence

The new error is

$$\begin{aligned}
&\vec{p} - \vec{e}_{\perp}(t+1) - \alpha(t+1) \cdot \vec{1} \\
&= \tilde{\vec{p}}(t+1) \\
&= D^{-1}\left(A\tilde{\vec{p}}(t) + \vec{b}\right) \\
&= D^{-1}\left(A\left(\vec{p} - \vec{e}_{\perp}(t) - \alpha(t) \cdot \vec{1}\right) + \vec{b}\right) \\
&= \vec{p} - P\vec{e}_{\perp}(t) - \alpha(t) \cdot \vec{1},
\end{aligned}$$

thus $\vec{e}_{\perp}(t+1) = P\vec{e}_{\perp}(t) - (\alpha(t+1) - \alpha(t))\vec{1}$

Jacobi's Convergence

Projecting the error

$$\vec{e}_\perp(t+1) = \left(I - \frac{1}{n}\vec{1}\vec{1}^\intercal\right) P\vec{e}_\perp(t)$$

$$= \left(I - \frac{1}{n}\vec{1}\vec{1}^\intercal\right) P \left(I - \frac{1}{n}\vec{1}\vec{1}^\intercal\right) P\vec{e}_\perp(t-1)$$

$$= \left(I - \frac{1}{n}\vec{1}\vec{1}^\intercal\right) \left(P - \frac{1}{n}\vec{1}\vec{1}^\intercal\right) P\vec{e}_\perp(t-1)$$

$$= \left(I - \frac{1}{n}\vec{1}\vec{1}^\intercal\right) P^2\vec{e}_\perp(t-1),$$

thus $\vec{e}_\perp(t) = \left(I - \frac{1}{n}\vec{1}\vec{1}^\intercal\right) P^t\vec{e}_\perp(0).$

Jacobi's Convergence

$P = D^{-1}A$ is similar to $N = D^{-1/2}AD^{-1/2}$.
Thus

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

Observe that
- N has n orthonormal eigenvec. $\vec{x}_1, \ldots, \vec{x}_n$, corresponding to eigenvectors $\vec{y}_1, \ldots, \vec{y}_n$ of P via $\vec{x}_i = D^{1/2}\vec{y}_i$ for each i.
- Both $\vec{x}_i$ and $\vec{y}_i$, for each i, are associated to the same eigenvalue ρ_i of P.

Jacobi's Convergence

$$
\begin{aligned}
\|\vec{e}_\perp(t)\| &= \left\| \left(I - \frac{1}{n}\vec{1}\vec{1}^\intercal \right) P^t \vec{e}_\perp(0) \right\| \\
&= \left\| \left(I - \frac{1}{n}\vec{1}\vec{1}^\intercal \right) D^{-\frac{1}{2}} N^t D^{\frac{1}{2}} \vec{e}_\perp(0) \right\| \\
&= \left\| \left(I - \frac{1}{n}\vec{1}\vec{1}^\intercal \right) D^{-\frac{1}{2}} \left(\sum_{i=2}^{n} \rho_i^t \vec{x}_i \vec{x}_i^\intercal \right) D^{\frac{1}{2}} \vec{e}_\perp(0) \right\| \\
&\leq \left\| \left(I - \frac{1}{n}\vec{1}\vec{1}^\intercal \right) \right\| \left\| D^{-\frac{1}{2}} \right\| \left\| \sum_{i=2}^{n} \rho_i^t \vec{x}_i \vec{x}_i^\intercal \right\| \left\| D^{\frac{1}{2}} \right\| \|\vec{e}_\perp(0)\| \\
&\leq \sqrt{\frac{\mathrm{vol}_{\max}}{\mathrm{vol}_{\min}}} \max(|\rho_2|, |\rho_n|)^t \|\vec{e}_\perp(0)\|,
\end{aligned}
$$

Jacobi's Convergence

$$\|\vec{e}_\perp(t)\| = \left\| \left(I - \frac{1}{n}\vec{1}\vec{1}^\intercal \right) P^t \vec{e}_\perp(0) \right\|$$

$$= \left\| \left(I - \frac{1}{n}\vec{1}\vec{1}^\intercal \right) D^{-\frac{1}{2}} N^t D^{\frac{1}{2}} \vec{e}_\perp(0) \right\|$$

$$= \left\| \left(I - \frac{1}{n}\vec{1}\vec{1}^\intercal \right) D^{-\frac{1}{2}} \left(\sum_{i=2}^{n} \rho_i^t \vec{x}_i \vec{x}_i^\intercal \right) D^{\frac{1}{2}} \vec{e}_\perp(0) \right\|$$

$$\leq \left\| \left(I - \frac{1}{n}\vec{1}\vec{1}^\intercal \right) \right\| \left\| D^{-\frac{1}{2}} \right\| \left\| \sum_{i=2}^{n} \rho_i^t \vec{x}_i \vec{x}_i^\intercal \right\| \left\| D^{\frac{1}{2}} \right\| \|\vec{e}_\perp(0)\|$$

$$\leq \sqrt{\frac{\mathrm{vol}_{\max}}{\mathrm{vol}_{\min}}} \max(|\rho_2|, |\rho_n|)^t \|\vec{e}_\perp(0)\|,$$

Conductance by Cheeger's ineq.

Randomized Token Diffusion Process

Doyle and Snell, '84 & Tetali, '91:
Times a random walk transits through given edge
until hitting the sink
- *global* requirement
- no accuracy and msg. complexity bounds

Our's:
How many tokens are on a node
- *local* requirement
- accuracy and msg. complexity w.r.t. *edge expansion*

Randomized Token Diffusion Process

Algorithm:

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 $u = \textit{source}$ **then**

  $Z(u) = Z(u) + K$

end

if $u = \textit{sink}$ **then**

  $Z(u) = 0$

end

Estimator:

Input: $u \in \mathcal{V}$, $K \in \mathbb{N}$

Return

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

Expected Behavior

Define inductively $\mathbf{p}^{(t)}$ by

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

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

Lemma. If $V_K^{(t)}(u) = \frac{Z_K^{t)}(u)}{K \mathrm{vol}(u)}$, then $\mathbb{E}[V_K^{(t)}(u)] = p_u^{(t)}$.

Correctness of Token Diffusion

We can write

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

with $\underline{P}$ and $\underline{\vec{b}}$ obtained by zeroing out entries on row and column of sink.

Lemma. The spectral radius of $\underline{P}$, $\underline{\rho}$, satisfies $\underline{\rho} = 1 - \sum_{i=1}^{n} v_i \cdot P_{i,\text{sink}} / \|v_1\|$, where $\vec{v}_1$ is left Perron eigenvector of $\underline{P}$.

Theorem. System above converges to a valid potential with rate $\underline{\rho}$.

Time and Message Complexity

Theorem. $1 - \underline{\rho} \geq \frac{\overline{\lambda}_2}{2\mathrm{vol}_{\max}(n-1)} \sum_i \frac{w_{in}}{w_{in}+\overline{\lambda}_2}$
where $\overline{\lambda}_2$ is 2nd smallest eigenv. of laplacian of graph with sink removed.

Connecting with *edge expansion*: it is known
$\lambda_2(\mathcal{G}) \geq \mathrm{vol}_{\max} - (\mathrm{vol}_{\max}^2 - \theta(\mathcal{G})^2)^{1/2}$.

As $t \to \infty$, the expected message complexity per round of Token Diffusion Algorithm is $O(K\, n\, \mathrm{vol}_{\max} \cdot E)$, where $E = \vec{p}^{\intercal} L \vec{p}$ is the *energy* of the electrical flow.

Stochastic Accuracy

X gives (ϵ, δ)-approximation of Y if
$\mathbf{P}\left(|X - Y| > \epsilon Y\right) \leq \delta$.

Lemma. For any K, $0 < \epsilon, \delta < 1$, t and u, such that $p_u^{(t)} \geq \frac{3}{\epsilon^2 K \mathrm{vol}(u)} \ln \frac{2}{\delta}$, the estimator provides an (ϵ, δ)-approximation of $p_u^{(t)}$.

Vice versa, (ϵ, δ)-approximation of the potentials $p_u^{(t)}$ greater than $p_\star^{(t)}$ is achieved by setting
$K \geq \frac{3}{\epsilon^2 p_\star^{(t)} \mathrm{vol}(u)} \ln \frac{2}{\delta}$.

Open Problem: Analysis of *Distributed* Physarum?

Physarum dynamics et sim.:
Compute electrical flow,
then *update edge-weigths*

$\Longrightarrow$ if we do:
comp.-electr.-flow, update-edges, comp.-electr.-flow,
update-edges, comp.-electr.-flow...
then
- Convergence time for each comp.-electr.-flow phase?
- Global convergence time?

Thank You!