max planck institut
informatik

The Physarum Computer

Kurt Mehlhorn

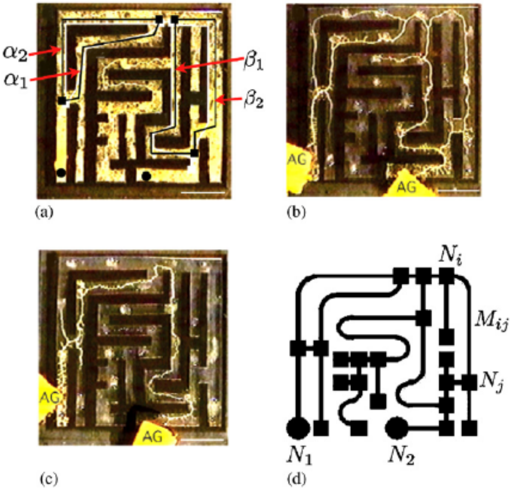
Max Planck Institute for Informatics and Saarland University

joint work with Vincenzo Bonifaci and Girish Varma

paper available on my homepage

August 30, 2011

The Physarum Computer



Physarum, a slime mold,

single cell, several nuclei

builds evolving networks

Nakagaki, Yamada, Tóth, Nature 2000

show video

2008 Ig Nobel Prize

For achievements that first make people LAUGH
then make them THINK

COGNITIVE SCIENCE PRIZE: Toshiyuki Nakagaki, Ryo Kobayashi, Atsushi Tero, Ágotá Tóth
for discovering that slime molds can solve puzzles.

REFERENCE: "Intelligence: Maze-Solving by an Amoeboid Organism," Toshiyuki Nakagaki, Hiroyasu Yamada, and Ágota Tóth, Nature, vol. 407, September 2000, p. 470.

Mathematical Model (Tero et al.)

- $G = (V, E)$ undirected graph
- each edge e has a positive length L_e (fixed) and a positive diameter $D_e(t)$ (dynamic)
- send one unit of flow from s_0 to s_1 in an electrical network where resistance of e equals

$$R_e(t) = L_e / D_e(t).$$

- $Q_e(t)$ is flow across e at time t
- Dynamics:

$$\dot{D}_e(t) = \frac{dD_e(t)}{dt} = |Q_e(t)| - D_e(t).$$

- 1 and 3 links

Tero et al., J. of Theoretical Biology, 553 – 564, 2007

Mathematical Model II: The Node Potentials

- electrical flows are driven by node potentials
- $Q_e = D_e(p_u - p_v)/L_e$ is flow on edge $\{ u, v \}$ from u to v
- flow conservation gives n equations, one for each vertex u

$$\sum_{e=\{ u,v \} \in E} D_e(p_u - p_v)/L_e = \delta_u$$

- $\delta_u = \pm 1$ if $u \in \{ s_0, s_1 \}$ and $\delta_u = 0$, otherwise
- together with $p_{s_1} = 0$, the above defines the p_v's uniquely
- can be computed by solving a linear system

Mathematical Model II: The Node Potentials

- electrical flows are driven by node potentials
- $Q_e = D_e(p_u - p_v)/L_e$ is flow on edge $\{ u, v \}$ from u to v
- flow conservation gives n equations, one for each vertex u

$$\sum_{e=\{ u,v \}\in E} D_e(p_u - p_v)/L_e = \delta_u$$

- $\delta_u = \pm 1$ if $u \in \{ s_0, s_1 \}$ and $\delta_u = 0$, otherwise
- together with $p_{s_1} = 0$, the above defines the p_v's uniquely
- can be computed by solving a linear system

Computer Experiments (Discrete Time)

initialize potentials
while true **do**
  update diameters: $D_e(t+1) = D_e(t) + \epsilon(|Q_e(t)| - D_e(t))$
  recompute potentials
end while

In simulations, the system converges (Miyaji/Ohnishi 07/08)

- e on shortest s_0-s_1 path: D_e converges to 1
- e not on shortest path: D_e converges to 0

Miyaji/Ohnishi ran simulations only on small graphs

We ran experiments on thousands of graphs of size up to 50,000 vertices and 200,000 edges. Confirmed their findings.

The Questions

Does system convergence for all (!!!) initial conditions?

How fast is the convergence?

Details of the convergence process?

Beyond shortest paths?

Inspiration for distributed algorithms?

Convergence against Shortest Path

Theorem (Convergence)

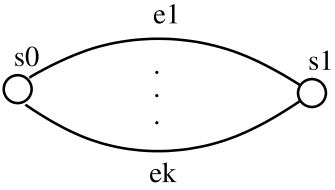
Dynamics converge against shortest path, i.e.,

$D_e \to 1$ *for edges on shortest source-sink path and* $D_e \to 0$ *otherwise.*

this assumes that shortest path is unique; otherwise converge against a flow of value 1 using only shortest source-sink paths

Miyaji/Onishi previously proved convergence for planar graphs with source and sink on the same face

Parallel Links (Miyaji/Ohnishi 07)



parallel links with lengths $L_1 < L_2 < \ldots < L_k$

$D_1 \to 1$, $D_2, \ldots, D_k \to 0$

$p_{s_0} - p_{s_1} \to L_1$

but $D_2, \ldots, D_{k-1}$ do not necessarily converge monotonically

What did Evolution Optimize?

Evolution optimized dynamics so as to achieve a global objective. Which? (Lyapunov Function)

First idea: the energy of the flow $\sum_e Q_e \Delta_e$ decreases over time

not true, even for parallel links

Theorem

For the case of parallel links: $\sum_i Q_i L_i$, $\sum_i D_i L_i / \sum_i D_i$, and $(p_s - p_t) \sum_i D_i L_i$ decrease over time

computer experiment: the obvious generalizations (replace i by e) to general graphs do not work

What did Evolution Optimize?

Evolution optimized dynamics so as to achieve a global objective. Which? (Lyapunov Function)

First idea: the energy of the flow $\sum_e Q_e \Delta_e$ decreases over time

not true, even for parallel links

Theorem

For the case of parallel links: $\sum_i Q_i L_i$, $\sum_i D_i L_i / \sum_i D_i$, and $(p_s - p_t) \sum_i D_i L_i$ decrease over time

computer experiment: the obvious generalizations (replace i by e) to general graphs do not work

What did Evolution Optimize?

Evolution optimized dynamics so as to achieve a global objective. Which? (Lyapunov Function)

First idea: the energy of the flow $\sum_e Q_e \Delta_e$ decreases over time

not true, even for parallel links

Theorem

For the case of parallel links: $\sum_i Q_i L_i$, $\sum_i D_i L_i / \sum_i D_i$, *and* $(p_s - p_t) \sum_i D_i L_i$ *decrease over time*

computer experiment: the obvious generalizations (replace i by e) to general graphs do not work

What did Evolution Optimize?

Evolution optimized dynamics so as to achieve a global objective. Which? (Lyapunov Function)

First idea: the energy of the flow $\sum_e Q_e \Delta_e$ decreases over time

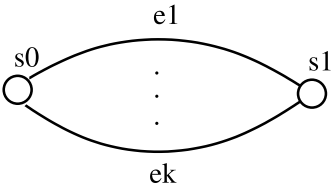
not true, even for parallel links

Theorem

For the case of parallel links: $\sum_i Q_i L_i$, $\sum_i D_i L_i / \sum_i D_i$, and $(p_s - p_t) \sum_i D_i L_i$ decrease over time

computer experiment: the obvious generalizations (replace *i* by *e*) to general graphs do not work

A not so Obvious Generalization



$$\frac{\sum_i D_i L_i}{\sum_i D_i} \quad \Rightarrow \quad \frac{\sum_e D_e L_e}{\text{value of min } s_0\text{-}s_1 \text{ cut with } cap_e = D_e}$$

What did Evolution Optimize?

Computer experiment:

$$V := \frac{\sum_e D_e L_e}{\text{value of min } s_0\text{-}s_1 \text{ cut with } cap_e = D_e} \quad \text{decreases}$$

Theorem (Lyapunov Function)

$$V + \left(\sum_{e \in \delta(\{ s_0 \})} D_e - 1\right)^2 \quad \textit{decreases.}$$

Derivative of V (essentially) satisfies

$$\dot{V} \le -c \cdot \sum_e (D_e - |Q_e|)^2.$$

Proof is brute-force except for applications of min-cut-max-flow and . . .

What did Evolution Optimize?

Computer experiment:

$$V := \frac{\sum_e D_e L_e}{\text{value of min } s_0\text{-}s_1 \text{ cut with } cap_e = D_e} \quad \text{decreases}$$

Theorem (Lyapunov Function)

$$V + \left(\sum_{e \in \delta(\{s_0\})} D_e - 1\right)^2 \quad \textit{decreases.}$$

Derivative of V (essentially) satisfies

$$\dot{V} \leq -c \cdot \sum_e (D_e - |Q_e|)^2.$$

Proof is brute-force except for applications of min-cut-max-flow and ...

What did Evolution Optimize?

Computer experiment:

$$V := \frac{\sum_e D_e L_e}{\text{value of min } s_0\text{-}s_1 \text{ cut with } cap_e = D_e} \quad \text{decreases}$$

Theorem (Lyapunov Function)

$$V + \left(\sum_{e \in \delta(\{ s_0 \})} D_e - 1\right)^2 \quad \textit{decreases.}$$

Derivative of V (essentially) satisfies

$$\dot{V} \leq -c \cdot \sum_e (D_e - |Q_e|)^2.$$

Proof is brute-force except for applications of min-cut-max-flow and ...

Convergence against Shortest Path

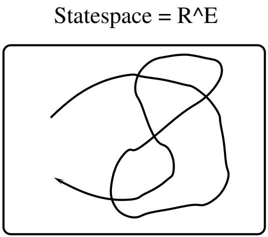
Corollary (Convergence)

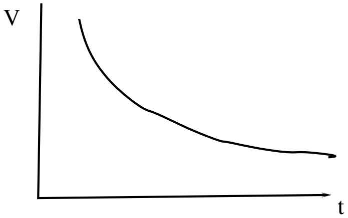
Dynamics converge against shortest path, i.e.,

$D_e \to 1$ *for edges on shortest s-t path and* $D_e \to 0$ *otherwise.*

this assumes that shortest path is unique; otherwise, ...

Miyaji/Onishi previously proved convergence for planar graphs.





- V decreases and stays positive $\Rightarrow$ $\dot{V} \rightarrow 0$
- $\dot{V} \leq -c \cdot \sum_e (D_e - |Q_e|)^2$
- $|D_e - |Q_e||$ goes to zero for all e
- $Q_e = (D_e/L_e)\Delta_e$ and hence $\Delta_e \approx L_e$ for Q_e and t large
- $\Delta_{s_0 s_1}$ converges to length of some source-sink path
- $\Delta_{s_0 s_1}$ converges to length of shortest path
- ...

Stable Topology (Miyaji/Ohnishi)

How fast is the convergence?

Definition: An edge $e = (u, v)$ stabilizes if for all $\varepsilon > 0$ either

- $p_u(T) \geq p_v(T) - \varepsilon$ for all large T or
- $p_v(T) \geq p_u(T) - \varepsilon$ for all large T.
- $|p_v(T) - p_u(T)| \leq \varepsilon$ for all large T.
- slightly more general than Miyaji/Ohnishi

Definition: A network stabilizes if all edges stabilize

A Path with Fixed Potential Difference

- assume p_a and p_b are fixed
- $L(P)$ length of path from a to b.
- define $f = (p_a - p_b)/L(P)$ and assume $f < 1$
- then for all edges of p: D decays like $\exp((f-1)t)$
- p_v converges to $p_b + (p_a - p_b)dist(v, b)/L(P)$

Stable Topology III

Theorem

If network stabilizes, network converges as defined next.

- decompose *undirected* G into paths: P_0 = shortest s-t path
- for $v \in P_0$: $p_v \to dist(v,t)$ for $e \in P_0$: $D_e \to 1$
- assume $P_0, \ldots, P_{i-1}$ are defined. Then
  - P_i has endpoints a and b on $P_0 \cup \ldots \cup P_{i-1}$
  - internal nodes and edges are fresh
  - maximizes $f_i := (p_a - p_b)/L(P_i)$ this is less than one
  - for $v \in P_i$: $p_v \to p_b + (p_a - p_b)dist(v,b)/L(P_i)$
  - for $e \in P_i$: $D_e \to 0$, exponentially with rate $f_i - 1$.
  - direct edges in P_i in direction from from a to b

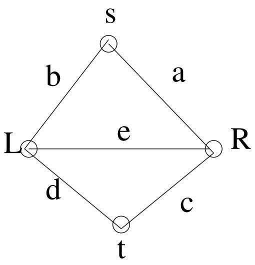
Open Problems

Do networks stabilize?

If so, after what time?

More generally, how long does it take for the dynamics to converge?

Wheatstone Graph

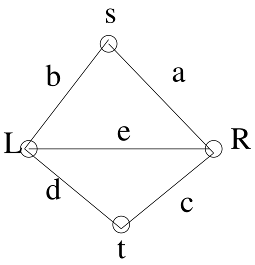


- simplest graph where flow directions are not clear
- direction of flow on e ????
- potentials evolve non-monotonically; run NonMonotone
- state space is cyclic; run TwoChanges

Theorem

Wheatstone network stabilizes

Wheatstone Graph



- simplest graph where flow directions are not clear
- direction of flow on e ????
- potentials evolve non-monotonically; run NonMonotone
- state space is cyclic; run TwoChanges

Theorem

Wheatstone network stabilizes

Wheatstone: Middle Edge Stabilizes

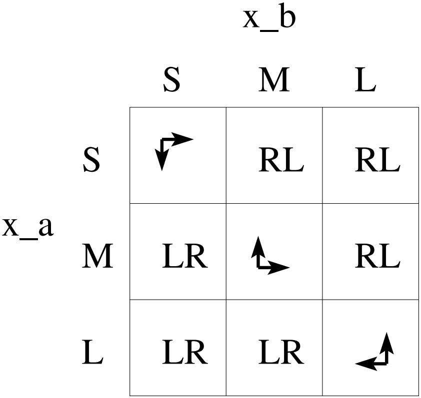
- $R_i = L_i/D_i$ = resistance of edge i.
- $x_a = R_a/(R_a + R_c)$, similarly for b.
- if $x_a < x_b$, direction of e is RL
  if $x_a > x_b$, direction of e is LR
- $x_a^* = L_a/(L_a + L_c)$, similarly for b. assume $x_a^* \le x_b^*$
  - split $[0, 1]$ into $S = [0, x_a^*)$, $M = [x_a^*, x_b^*]$ and $L = [x_b^*, 1]$
  - consider evolvement of (x_a, x_b)
  - in $S \times S$, both grow:
  - in $M \times M$, x_a decreases and x_b grows
  - in $L \times L$, both shrink

Wheatstone: Middle Edge Stabilizes

- split [0, 1] into

$$S = [0, x_a^*)$$
$$M = [x_a^*, x_b^*]$$
$$L = [x_b^*, 1]$$

- consider evolvement of (x_a, x_b)
- in $S \times S$, both grow:
- in $M \times M$, x_a decreases and x_b grows
- in $L \times L$, both shrink

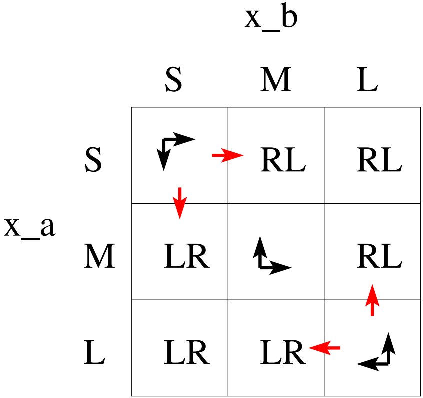


Wheatstone: Middle Edge Stabilizes

- split [0, 1] into

$$S = [0, x_a^*)$$
$$M = [x_a^*, x_b^*]$$
$$L = [x_b^*, 1]$$

- consider evolvement of (x_a, x_b)
- in $S \times S$, both grow:
- in $M \times M$, x_a decreases and x_b grows
- in $L \times L$, both shrink

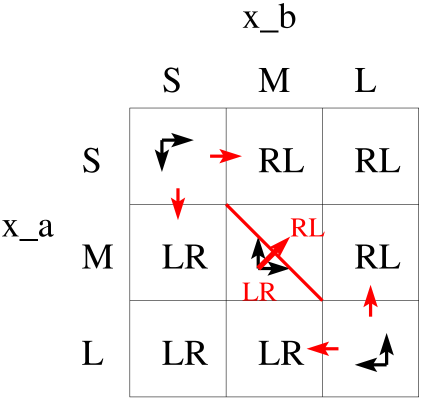


Wheatstone: Middle Edge Stabilizes

- split $[0,1]$ into

$$S = [0, x_a^*)$$
$$M = [x_a^*, x_b^*]$$
$$L = [x_b^*, 1]$$

- consider evolvement of (x_a, x_b)
- in $S \times S$, both grow:
- in $M \times M$, x_a decreases and x_b grows
- in $L \times L$, both shrink

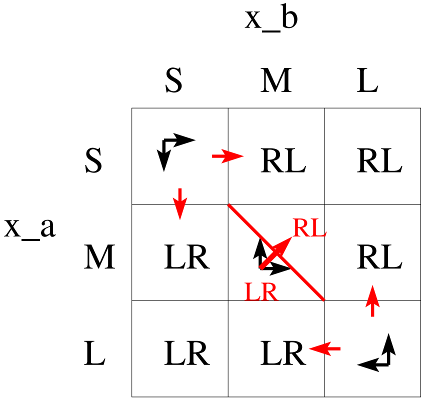


Wheatstone: Middle Edge Stabilizes

- split [0, 1] into

$$S = [0, x_a^*)$$
$$M = [x_a^*, x_b^*]$$
$$L = [x_b^*, 1]$$

- consider evolvement of (x_a, x_b)
- in $S \times S$, both grow:
- in $M \times M$, x_a decreases and x_b grows
- in $L \times L$, both shrink



what if system stays in $S \times S$ or $L \times L$?

The Transportation Problem

- undirected graph $G = (V, E)$
- $b : V \to \mathbb{R}$ such that $\sum_v b_v = 0$
- v supplies flow b_v if $b_v > 0$
- v extracts flow $|b_v|$ if $b_v < 0$
- find a cheapest flow where cost of sending x units across an edge of length L is Lx

Dynamics of Physarum solves transportation problem.

D_e's converge against a mincost solution of transportation problem.

Open Problems and Related Work

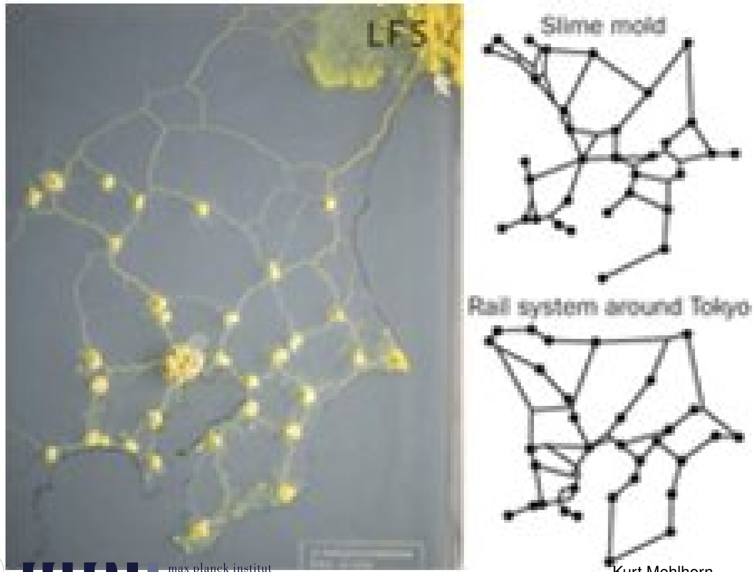
Open Problems

- show: flow directions stabilize
- show: convergence is exponential
- Physarum apparently can do more, e.g., network design. Prove it.
- inspiration for the design of distributed algorithms

Related Work

Ito/Johansson/Nakagaki/Tero: Convergence Properties for the Physarum Solver, January 28th, 2011, they change $|Q_e|$ into Q_e and prove convergence for all graphs

Network Design: Science 2010



Natural Computation

- Humans and Animals are not Turing Machines

  Part of their computational capabilities is based on their bodies

  Other Models of Computation are Relevant

- Suggestions for distributed algorithms

- CS methods can help analyzing such systems, do not leave it to physicists and biologists