Exploring graphs using parallel rotor walks

Dominik Pajak

LaBRI, Inria Bordeaux Sud-Ouest, France

Includes results of joint work with:

Dariusz Dereniowski, Ralf Klasing, Adrian Kosowski

Thomas Sauerwald and Przemysław Uznański

Cargése April 2014

Graph exploration

- A team of **agents** is placed on some subset of nodes of the network.
- The agents are propagated along edges of the network following a local set of rules defined for each node.
- Agents are searching for a treasure hidden in one of the nodes of the network.
- The goal of the agents is to visit each node (i.e. to explore the whole network).

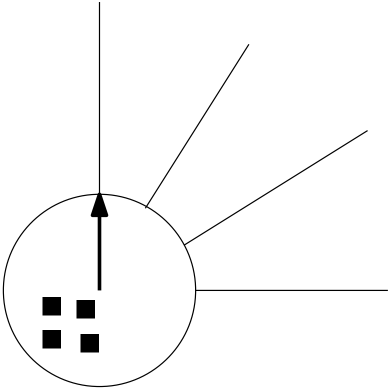
The Rotor-router model

- Each node v has a fixed local port numbering from 1 to $deg(v)$
- The state of each node v is a pointer $p(v) \in \{1, ..., deg(v)\}$.

Rotor-Router Mechanism

For each agent located at node v at the start of time round t:

- The agent is pushed to the neighbor along port $p(v)$
- Pointer $p(v)$ is incremented modulo the degree.



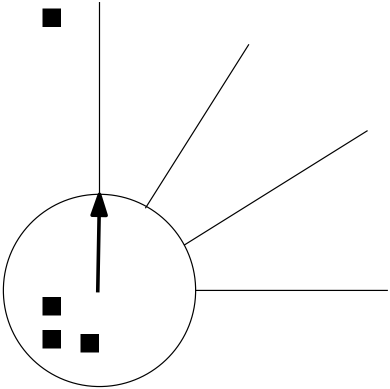
The Rotor-router model

- Each node v has a fixed local port numbering from 1 to $deg(v)$
- The state of each node v is a pointer $p(v) \in \{1, ..., deg(v)\}$.

Rotor-Router Mechanism

For each agent located at node v at the start of time round t:

- The agent is pushed to the neighbor along port $p(v)$
- Pointer $p(v)$ is incremented modulo the degree.



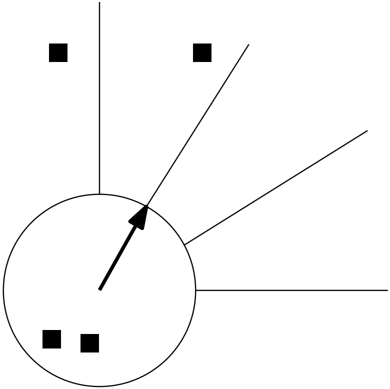
The Rotor-router model

- Each node v has a fixed local port numbering from 1 to $deg(v)$
- The state of each node v is a pointer $p(v) \in \{1, ..., deg(v)\}$.

Rotor-Router Mechanism

For each agent located at node v at the start of time round t:

- The agent is pushed to the neighbor along port $p(v)$
- Pointer $p(v)$ is incremented modulo the degree.



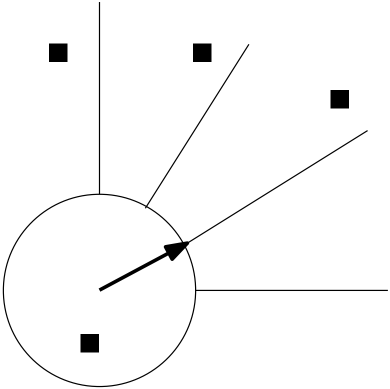
The Rotor-router model

- Each node v has a fixed local port numbering from 1 to $deg(v)$
- The state of each node v is a pointer $p(v) \in \{1, ..., deg(v)\}$.

Rotor-Router Mechanism

For each agent located at node v at the start of time round t:

- The agent is pushed to the neighbor along port $p(v)$
- Pointer $p(v)$ is incremented modulo the degree.



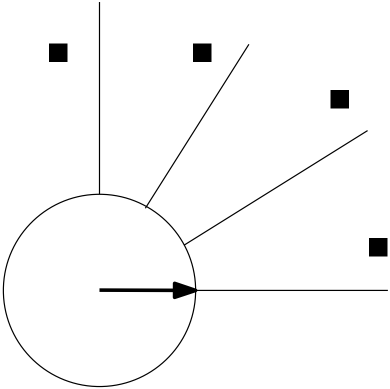
The Rotor-router model

- Each node v has a fixed local port numbering from 1 to $deg(v)$
- The state of each node v is a pointer $p(v) \in \{1, ..., deg(v)\}$.

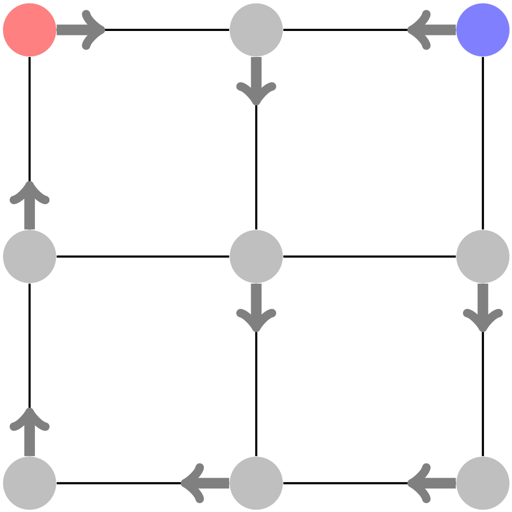
Rotor-Router Mechanism

For each agent located at node v at the start of time round t:

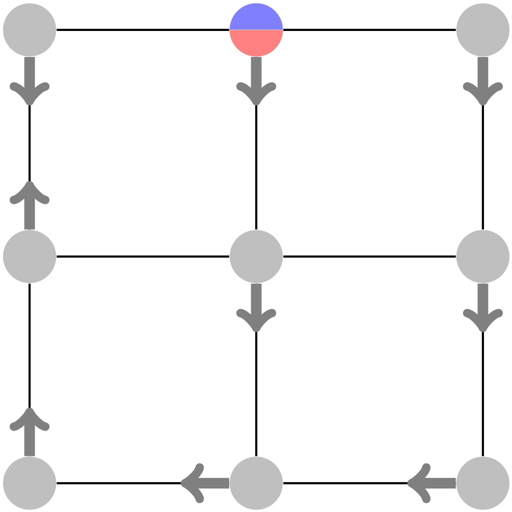
- The agent is pushed to the neighbor along port $p(v)$
- Pointer $p(v)$ is incremented modulo the degree.



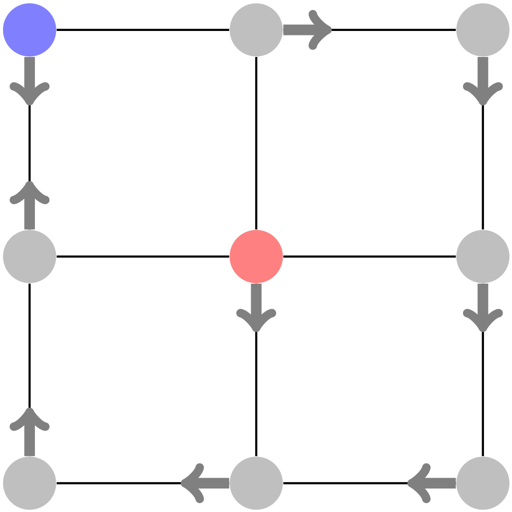
Example



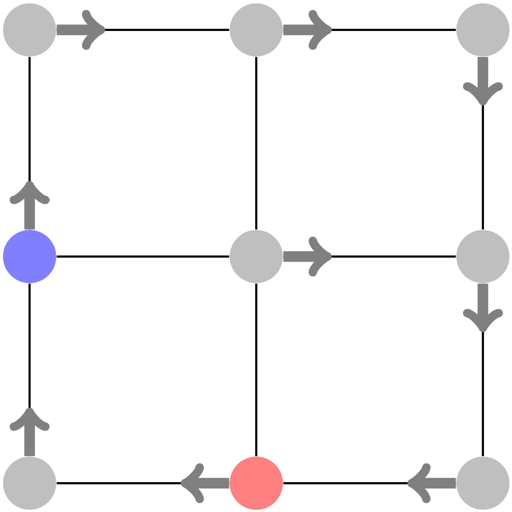
Example



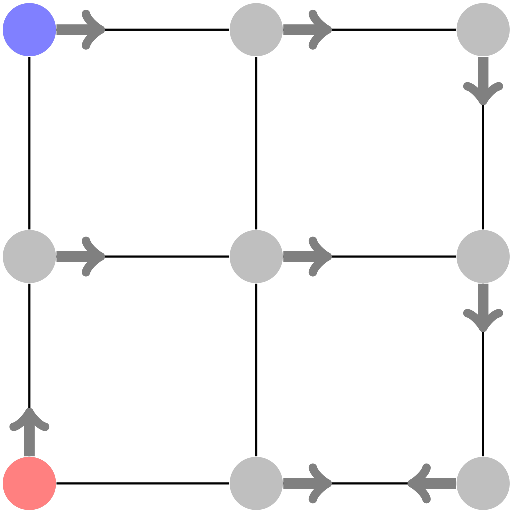
Example



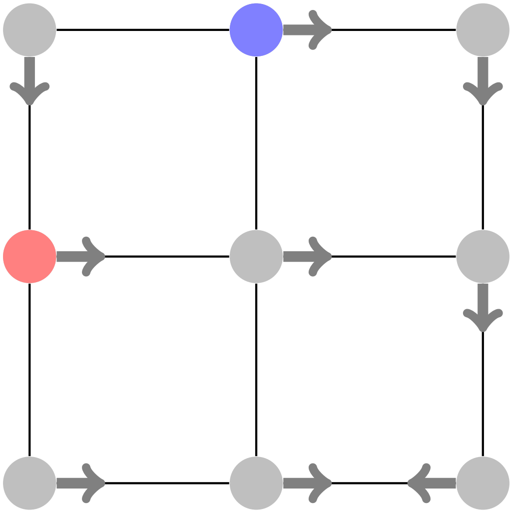
Example



Example



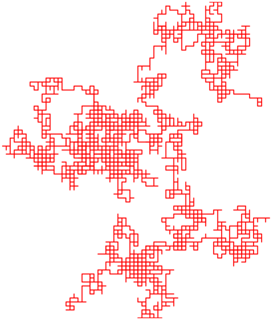
Example



Random walk

What is the random walk?

- The agent leaves each node along one of the adjacent links, chosen uniformly at random.
- From the perspective of a node it sends on average the same number of agents in each direction.



Question

Where does the rotor-router come from?

Answer 1

The rotor-router can be seen as a derandomization of the random walk.

Single random walk

Cover time of random walk

Expected time until agent visits all vertices.

Graph class	*Cover time*
Expander, Hypercube, Complete	$\Theta(n \log n)$
2-dim. torus	$\Theta(n \log^2 n)$
Cycle	$\Theta(n^2)$
Lollipop Graph	$\Theta(n^3)$
Any graph	$O(n^3)$, $\Omega(n \log n)$

Multiple random walks (k = number of agents)

Cover time of multiple random walks

Expected time until every node is visited by some agent.

Speedup

Ratio between the cover time for single walk and for multiple walks.

Graph class	*Speedup*
Expander, Hypercube, Complete, Random	k
Cycle	$\log k$
d-dim. torus ($d > 2$)	$k(k < n^{1-2/d})$

Table: Results from [Elsässer, Sauerwald, 2011] and [Alon, Avin, Koucky, Kozma, Lotker, Tuttle, 2008]

Multiple random walks (k = number of agents)

Cover time of multiple random walks

Expected time until every node is visited by some agent.

Speedup

Ratio between the cover time for single walk and for multiple walks.

Graph class	*Speedup*
Expander, Hypercube, Complete, Random	k
Cycle	$\log k$
d-dim. torus ($d > 2$)	$k (k < n^{1-2/d})$

Table: Results from [Elsässer, Sauerwald, 2011] and [Alon, Avin, Koucky, Kozma, Lotker, Tuttle, 2008]

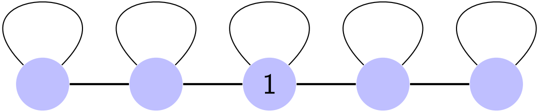
Conjecture [Alon, Avin, Koucky, Kozma, Lotker, Tuttle, 2008]

Speedup is $O(k)$ and $\Omega(\log k)$ for any graph.

Continuous Diffusion Model

- Each of the nodes v of the graph starts with a certain amount of resource $L_0(v)$ (real-valued, non-negative) – call it **load**.
- In each round, each of the nodes sends an equal part of its load to its neighbors

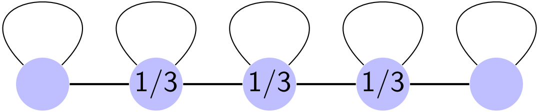
$$L_{t+1}(v) = \sum_{u \in N(v)} \frac{L_t(u)}{deg(u)}$$



Continuous Diffusion Model

- Each of the nodes v of the graph starts with a certain amount of resource $L_0(v)$ (real-valued, non-negative) – call it **load**.
- In each round, each of the nodes sends an equal part of its load to its neighbors

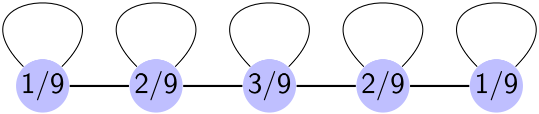
$$L_{t+1}(v) = \sum_{u \in N(v)} \frac{L_t(u)}{deg(u)}$$



Continuous Diffusion Model

- Each of the nodes v of the graph starts with a certain amount of resource $L_0(v)$ (real-valued, non-negative) – call it **load**.
- In each round, each of the nodes sends an equal part of its load to its neighbors

$$L_{t+1}(v) = \sum_{u \in N(v)} \frac{L_t(u)}{deg(u)}$$



Analysing continuous diffusion

Continuous diffusion is a linear and deterministic process:

$$L_{t+1} = ML_t \quad \Longrightarrow \quad L_t = M^t L_0,$$

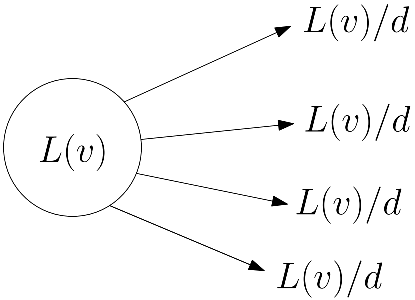
where M is the stochastic matrix ("random walk matrix") of the graph.

Problem: dealing with granular load.(not infinitely divisible)

- Assume load is expressed in multiple of unit values, each of which is propagated between neighboring nodes.
- We have k units in total, each node v starting with $L_0(v)$ units
- In general, it is no longer possible to follow the diffusion equation accurately.

Discrete diffusion rules

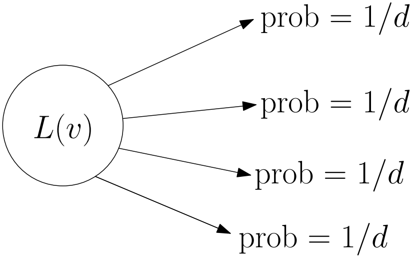
Reference point – continuous diffusion:



$d = deg(v)$, for a while, we will be considering regular graphs.

Rule 1

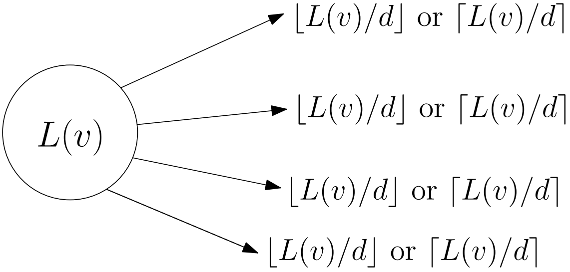
Independent random walk for each unit of load.



Expected number of units of load at each location for the random walk matches that in continuous diffusion: $\mathrm{E}[L_t] = M^t L_0$

Rule 2

Perform rounding of the continuous diffusion process.



(keeping sum intact)

Question

Where does the rotor-router come from?

Answer 2

Both rotor-router and the random walk can be seen as discretization of the continuous diffusion process.

The Rotor-router model

Configuration of the rotor-router

- Initialization of the port numbering
- Initial positions of agents.

When analysing the rotor-router we will always assume the worst possible initial configuration.

Parameters of the rotor-router

Cover time

When will have each node of the graph been reached by some agent, for a worst-case starting configuration?

Lock-in

- The rotor-router is a deterministic process with a finite number of states, hence it must stabilize to a periodic traversal of some cycle in its state space after some initialization phase
- After what time does the rotor-router enter its limit cycle?
- What is the length of the cycle?

Single agent rotor-router

Theorem [Yanovski, Wagner, Bruckstein, 2001]

- For any graph with diameter D and m edges, cover time and lock-in time are bounded by $O(mD)$.
- After this lock-in period, the rotor-router stabilizes to an **Eulerian traversal** of the directed version of the graph (traversing each edge once in each direction).

Theorem [Bampas, Gasieniec, Hanusse, Ilcinkas, Klasing, Kosowski]

- There exists an initial configuration of the rotor-router for which cover time and lock-in time are $\Omega(mD)$.

Single agent rotor-router

Theorem [Yanovski, Wagner, Bruckstein, 2001]

- For any graph with diameter D and m edges, cover time and lock-in time are bounded by $O(mD)$.
- After this lock-in period, the rotor-router stabilizes to an **Eulerian traversal** of the directed version of the graph (traversing each edge once in each direction).

Theorem [Bampas, Gasieniec, Hanusse, Ilcinkas, Klasing, Kosowski]

- There exists an initial configuration of the rotor-router for which cover time and lock-in time are $\Omega(mD)$.

Single agent rotor-router exhibits elegant structural properties.

For any graph, for the worst-case initial configuration

- Cover time is $\Theta(mD)$.
- Lock-in time is $\Theta(mD)$.
- Cycle length is $\Theta(D)$.

Rotor-router vs. random walk

For a single agent it is hard to see any correlation between cover time of the random walk and the rotor-router.

Graph class	*Cover time*	
	Random walk	*Rotor-router*
Cycle	$\Theta(n^2)$	$\Theta(n^2)$
Complete graph	$\Theta(n \log n)$	$\Theta(n^2)$
Star	$\Theta(n \log n)$	$\Theta(n)$
Grid $\sqrt{n} \times \sqrt{n}$	$\Theta(n \log^2 n)$	$\Theta(n^{3/2})$
Hypercube	$\Theta(n \log n)$	$\Theta(n \log^2 n)$

How about multiple agents?

Multiple agents are interacting with the same rotor-router model

- no independence of walks!
- can we have similar results for multi-agent rotor-router as for multiple random walks?

Goal

We want to study the speedup (as a function of k) of the cover time of the multi-agent rotor-router with respect to the single agent.

Multi-agent rotor-router

Lemma [Yanovski, Wagner, Bruckstein, 2001]

Adding an agent cannot decrease the number of visits at any node at any time.

Lemma [Klasing, Kosowski, P., Sauerwald, 2013]

Blocking some agents for some time steps cannot increase the number of visits at any node at any time.

Multi-agent rotor-router

Lemma [Yanovski, Wagner, Bruckstein, 2001]

Adding an agent cannot decrease the number of visits at any node at any time.

Lemma [Klasing, Kosowski, P., Sauerwald, 2013]

Blocking some agents for some time steps cannot increase the number of visits at any node at any time.

Delayed deployments

A process obtained from a rotor-router by defining how many agents to delay at which times and at which nodes.

The slow-down lemma

- $R[k]$ - k-agent rotor router system with an arbitrarily chosen initialization.
- We construct delayed deployment D such that:
  - deployment D explores the graph in at most T steps,
  - in at least τ of these steps all agents were active in D.

Theorem [Klasing, Kosowski, P., Sauerwald, 2013]

The cover time $C(R[k])$ of the system can be bounded by:
$\tau \leq C(R[k]) \leq T$.

Applications of the slow-down lemma

The slow-down lemma plays key part in our analysis of the multi agent rotor-router:

- We can analyze $R[k]$ by constructing some easy to analyze, delayed deployment D.
- This allows us to think of the rotor-router as an algorithm, rather than a process which is imposed upon us.
- If the deployment D is defined so that agents in D are delayed in at most a constant proportion of the first $C(D)$ rounds, then the above inequalities lead to an asymptotic bound on the value of the undelayed rotor-router, $C(R[k]) = \Theta(C(D))$.

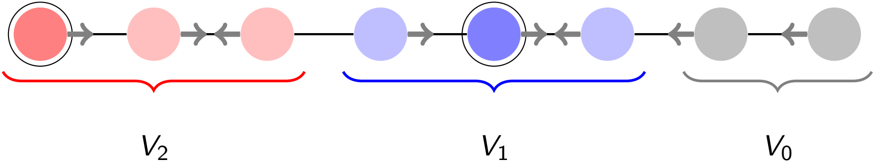
Multi-agent rotor-router on the ring

The rotor-router on the path (or ring) for $k \ll n$

- Intuition: Each agent occupies a "domain", which it patrols.
- A node v belongs to domain $V_i(t)$ of the i-th agent if this agent was the last agent visiting node v until round t, inclusive.
- A special domain $V_0(t)$ contains all nodes which have not yet been visited.
- One can show that domains either form spontaneously as segments, or by holding back a few agents we can force them to form (delayed deployment).[Klasing, Kosowski, P., Sauerwald, 2013]
- Within a domain, all ports are aligned "towards" the agent which is its owner.

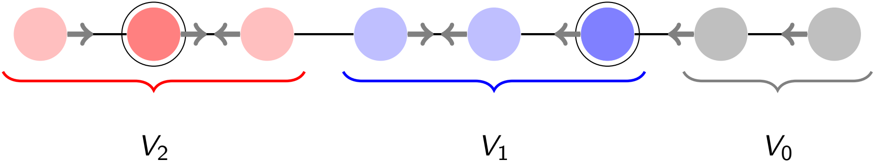
Agent domains

Example on the line, $k = 2$ (starting from some moment...)



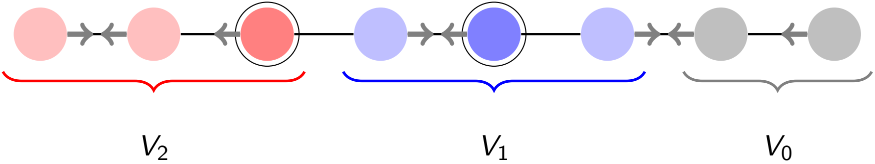
Agent domains

Example on the line, $k = 2$ (starting from some moment...)



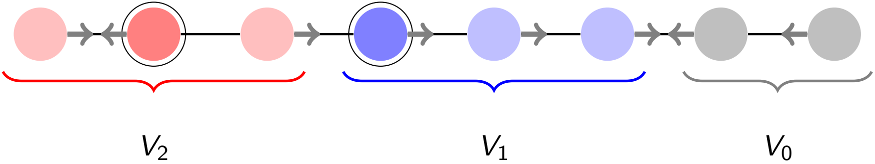
Agent domains

Example on the line, $k = 2$ (starting from some moment...)



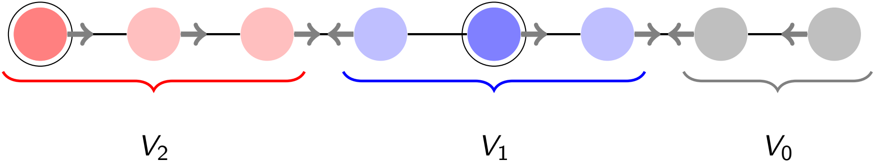
Agent domains

Example on the line, $k = 2$ (starting from some moment...)



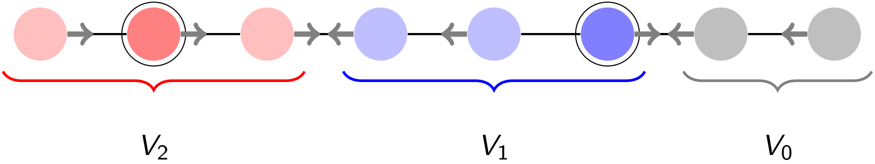
Agent domains

Example on the line, $k = 2$ (starting from some moment...)



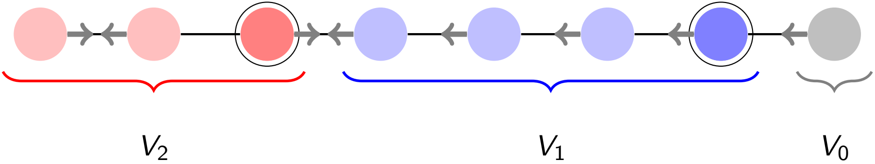
Agent domains

Example on the line, $k = 2$ (starting from some moment...)



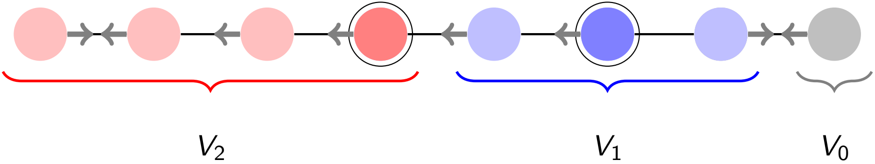
Agent domains

Example on the line, $k = 2$ (starting from some moment...)



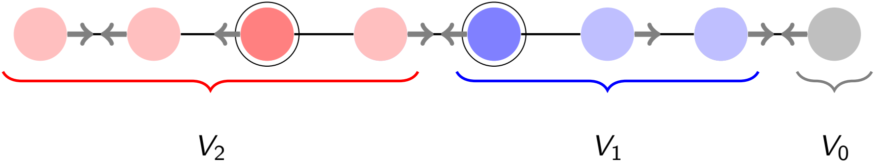
Agent domains

Example on the line, $k = 2$ (starting from some moment...)



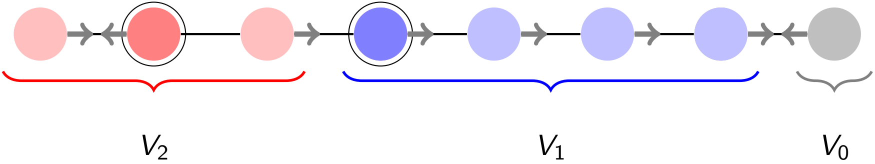
Agent domains

Example on the line, $k = 2$ (starting from some moment...)



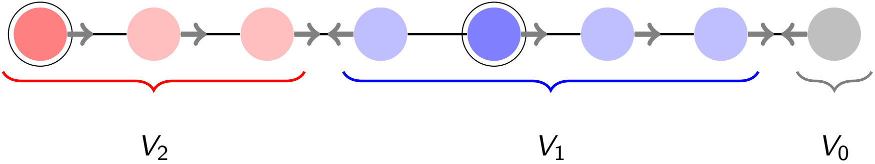
Agent domains

Example on the line, $k = 2$ (starting from some moment...)



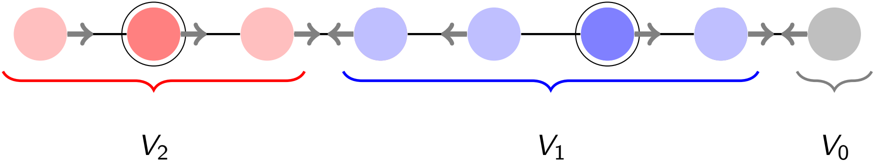
Agent domains

Example on the line, $k = 2$ (starting from some moment...)



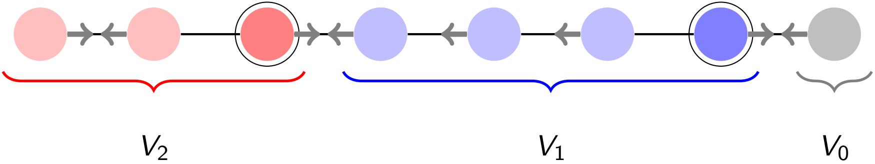
Agent domains

Example on the line, $k = 2$ (starting from some moment...)



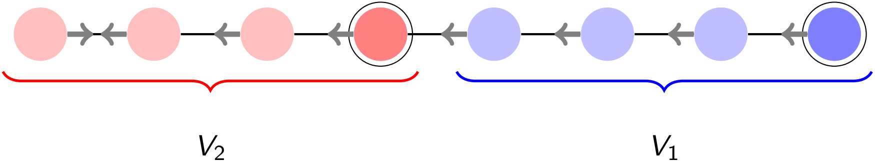
Agent domains

Example on the line, $k = 2$ (starting from some moment...)



Agent domains

Example on the line, $k = 2$ (starting from some moment...)



- Agents are traversing their domains and during each cycle can capture one node from neighboring domain (or at least one node not belonging to any domain).
- Agents with smaller domains will visit borders more frequently thus smaller domains will grow.
- Intuitively the system should converge to domains of equal sizes.

Continuous time approximation

- Roughly speaking, each agent i enlarges its own domain of size $n_i(t) = |V_i(t)|$ once every $n_i(t)$ steps (once at the left end, once at the right end)
- At each of the ends, the size of the domain is reduced by the adjacent agent (except from the side with $V_0(t)$, if applicable).
- We define the *continuous-time approximation*:

$$\frac{d\nu_i(t)}{dt} = \frac{1}{\nu_i(t)} - \frac{1}{2\nu_{i-1}(t)} - \frac{1}{2\nu_{i+1}(t)}, \qquad \text{for } 1 \le i \le k,$$

- This approximation is accurate in the sense that one can construct a delayed deployment which (almost) adheres to its solution.

Multi-agent rotor-router on the ring

Theorem [Kosowski, P., 2014][Klasing, Kosowski, P., Sauerwald, 2013]

Worst-case cover time for k agent rotor-router on the ring is $\Theta(n^2/\log k)$ when $k < 2^n$.

So the speedup for the ring is $\log k$.

Model	*Cover time* *worst placement*	*best placement*	*Return time*
k-agent rotor-router	$\Theta(n^2/\log k)$	$\Theta(n^2/k^2)$	$\Theta(n/k)$
k random walks (expectations)	$\Theta(n^2/\log k)$ in literature	$\Theta\left(n^2 \Big/ \frac{k^2}{\log^2 k}\right)$	$\Theta(n/k)$ in literature

Multi-agent rotor-router in general graphs

- Even less structure – forget about domains.
- Slowdown lemma still holds and proves useful.

Theorem [Dereniowski, Kosowski, P., Uznanski, 2014]

The k-agent rotor-router covers any graph in worst-case time $O(mD/\log k)$ and $\Omega(mD/k)$

- Both of these bounds are achieved for some graph classes.
- The range of speedup for the rotor-router corresponds precisely to the conjectured range of speedup for the random walk.

1 agent versus k agents: comparison of speedup

Graph class	Speedup of Rotor-Router *for cover time*	Speedup of Random Walk *for cover time*	*for max hitting time*
General case:	$\Omega(\log k)$, $O(k)$	$O(k^2)$, $O(k \log n)$	$O(k)$
Cycle:	$\Theta(\log k)$	$\Theta(\log k)$	$\Theta(\log k)$
Star:	$\Theta(k)$	$\Theta(k)$	$\Theta(k)$

(all results hold up to k polynomially large with respect to n)

Multi-agent rotor-router in different graph classes

To analyse the cover time of the multi agent rotor-router for other graph classes we tried a different approach.

Lemma

For any time t, the total number of visits until time t in the rotor-router and the cumulative load (=sum of loads) until time t in the continuous diffusion differ by at most
$\Psi_t = \max_{v \in V} \sum_{\tau=0}^{t} \sum_{(u_1,u_2) \in \vec{E}} |P_\tau(u_1, v) - P_\tau(u_2, v)|$.

where $P_t(u, v)$ is probability that the random walk starting at u after t steps is located at v.
$\Psi(G) = \Psi_\infty(G)$ is called **local divergence** and was defined in [Rabani, Sinclair, Wanka 1998].

$C^k_{rr}(G)$ – cover time of k agent rotor-router on graph G.

Lemma

Let t^* be the smallest time such that the cumulative load in the continuous diffusion until time t^* is more than Ψ_{t^*}, then

$$C^k_{rr}(G) \leq t^*$$

Multi-agent rotor-router in different graph classes

Let us define the following time

$$t_{1/4} = \max_{u \in V} \min \left\{ t : \forall_{u \in V} P_t(u, v) \geq \frac{\deg(v)}{4m} \right\},$$

If time is at least $t_{1/4}$ then the load at any node in the continuous diffusion starting with k units of load is at least $\frac{k \deg(v)}{4m}$.

Theorem

The cover time $C_{rr}^k(G)$ of a k-agent rotor-router with arbitrary initialization on any non-bipartite graph G satisfies

$$C_{rr}^k(G) \leq t_{1/4} + \frac{4\Delta}{\delta} \frac{n}{k} \Psi(G).$$

Where Δ – maximum degree, δ – minimum degree.
If we can bound $\Psi(G)$, we can bound the cover time!

Complete graph

Theorem

If G is a clique then

$$C_{rr}^{k}(G) = \begin{cases} \Theta\left(\frac{n^2}{k}\right) & \textit{for } k \leq n^2 \\ \Theta(1) & \textit{for } k > n^2 \end{cases}$$

Expander

Theorem

If G is a degree-restricted expander then

$$C_{rr}^{k}(G) = \begin{cases} \Theta\left(\frac{mD}{k}\right) & \text{for } k \leq m \\ \Theta(D) & \text{for } k > m \end{cases}$$

In expanders, the rotor-router parallelizes very well and achieves the cover time of $O(D)$ already for $k = m$.

Hypercube

For hypercubes we have an interval of linear speedup followed by an interval of slower speedup.

Theorem

If G is a hypercube with n vertices then

$$C_{rr}^{k}(G) = \begin{cases} \Theta\left(\frac{n \log^2 n}{k}\right) & \text{for } k < n\frac{\log n}{\log\log n} \\ \Theta(\log n \log\log n) & \text{for } k \in \left[n\frac{\log n}{\log\log n}, n2^{\log^{1-\epsilon} n}\right] \\ O(\log n \log\log n) & \text{for } k > n2^{\log^{1-\epsilon} n} \\ \Theta(\log n) = \Theta(D) & \text{for } k > (\log n)^{\log n} \end{cases}$$

Torus

We observed a very interesting phenomenon for constant dimensional tori.

- We have linear speedup up to $n^{1-1/d}$.
- Adding more agents above $n^{1-1/d}$ gives only logarithmic speedup.

Theorem

If G is a torus of constant dimension then

$$C_{rr}^{k}(G) = \begin{cases} \Theta\left(\frac{n^{1+1/d}}{k}\right) & \text{for } k \leq n^{1-1/d} \\ \Theta\left(\frac{n^{2/d}}{\log(k/n^{1-1/d})}\right) & \text{for } 2^{n^{1/d}} n^{1-1/d} \geq k > n^{1-1/d} \\ \Theta(n^{1/d}) = \Theta(D) & \text{for } k \geq 2^{n^{1/d}} n^{1-1/d} \end{cases}$$

Multi-agent rotor-router vs. multiple random walks

In terms of the speedup, the multi-agent rotor-router resembles very much multiple random walks.

Graph class	*Speedup (for small k)*	
	Random walk	*Rotor-router*
Cycle	$\log k$	$\log k$
Complete graph	k	k
Star	k	k
Grid $\sqrt{n} \times \sqrt{n}$	$\geq k$	k
Hypercube	k	k
General graph	Conjecture: $\Omega(\log k)$	$\Omega(\log k)$
	Conjecture: $O(k)$	$O(k)$

Graph	*k*	*Cover time*
General graph	$\leq poly(n)$	$O\left(\frac{mD}{\log k}\right)$
		$\Omega\left(\frac{mD}{k}\right)$
Cycle	$< 2^n$	$\Theta\left(\frac{n^2}{\log k}\right)$
	$\geq 2^n$	$\Theta(n)$
d-dim. torus	$< n^{1-1/d}$	$\Theta\left(\frac{n^{1+1/d}}{k}\right)$
	$\in [n^{1-1/d}, n^{1-1/d}2^{n^{1/d}}]$	$\Theta\left(\frac{n^{2/d}}{\log(k/n^{1-1/d})}\right)$
	$> n^{1-1/d}2^{n^{1/d}}$	$\Theta(n^{1/d})$
Hypercube	$< n^{\frac{\log n}{\log\log n}}$	$\Theta\left(\frac{n\log^2 n}{k}\right)$
	$\in \left[n^{\frac{\log n}{\log\log n}}, n2^{\log^{1-\varepsilon} n}\right]$ (for any $\varepsilon > 0$)	$\Theta(\log n \log\log n)$
	$> n2^{\log^{1-\varepsilon} n}$	$O(\log n \log\log n)$
	$> 2^{\log_2 n \log_2 \log_2 n}$	$\Theta(\log n)$

Table: Cover time of the k-agent rotor-router system for different values of k in a n-node graph with m edges and diameter D. The result for expanders concerns the case when the ratio of the maximum degree and the minimum degree of the graph is $O(1)$. The result for random graphs holds in the Erdős-Renyi model with edge probability $p > (1+\varepsilon)\frac{\log n}{n}$, $\varepsilon > 0$, a.s.

Graph	*k*	*Cover time*
Complete	$< n^2$	$\Theta\left(\frac{n^2}{k}\right)$
	$\geq n^2$	$\Theta(1)$
Expander	$< n \log n$	$\Theta\left(\frac{n \log^2 n}{k}\right)$
	$\geq n \log n$	$\Theta(\log n)$
Random graph	$< n \log n$	$\Theta\left(\frac{n \log^2 n}{k}\right)$
	$\geq n \log n$	$\Theta(\log n)$

Open problems

1. Finish the hypercube.
2. What if we have agents with no memory and nodes with whiteboards. Agents can perform rotor-router, but can we do better? What if agents can have constant number of bits of internal memory?
3. What is the frequency of visits at vertices in the limit cycle?
4. Can one show that the k agent rotor-router enters a short period (say, a divisor of $2m$) a.s. on a random graph with random pointer initialization?
5. Are there simple examples of graphs for which the speedup is different than $\log k$ and k?

Thank You!