Local Averaging in Distributed Networks - Hoping for the Best

by *Frederik Mallmann-Trenn*

Thanks to: P. Berenbrink, C. Cooper, C. Gava, D. Pajak, D. Kohan Marzagão, Y. Maus, N. Rivera, and T. Radzik

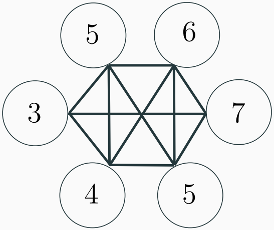
Outline

In this presentation:

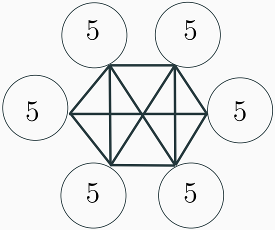
1. Model
2. Bi-lateral averaging (w/o noise)
3. Uni-lateral averaging (w/o noise)
4. Bi-lateral averaging with noise

The Model

- We are given a graph and each node has a value.



- Goal: Reach consensus, i.e., converge to the same value.



Motivation

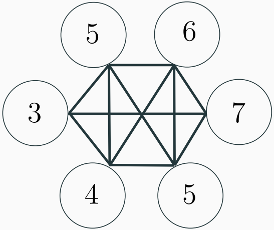
- **Robotics:** exploring the percentage of white tiles in an area, or microbots measuring the concentration of chemicals.
- **Sensor networks:** Infrared sensors, temperature sensors etc.
- **Social insects:** for ants, values could represent the individuals' different assessments of nest qualities when house hunting



PART II: Bi-lateral Updates

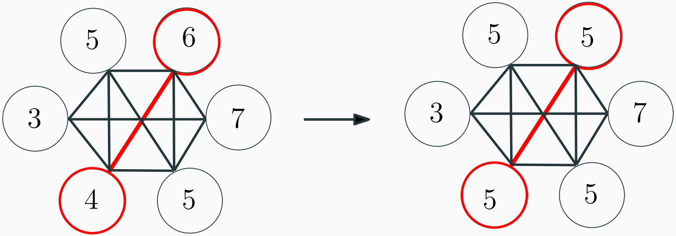
Noisefree Model

- Each node i has a value x_i. Let n be the total number of nodes
- In every round a **pair** of nodes (i, j) chosen **uniformly at random** interacts and exchange values
- The nodes quickly converge to the average



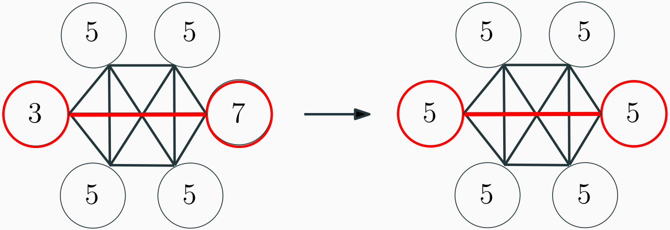
Noisefree Model

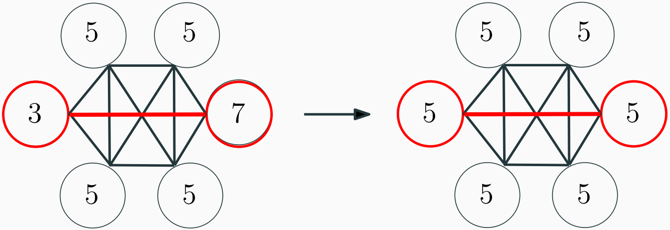
- Each node i has a value x_i. Let n be the total number of nodes
- In every round a **pair** of nodes (i, j) chosen **uniformly at random** interacts and exchange values
- The nodes quickly converge to the average



Noisefree Model

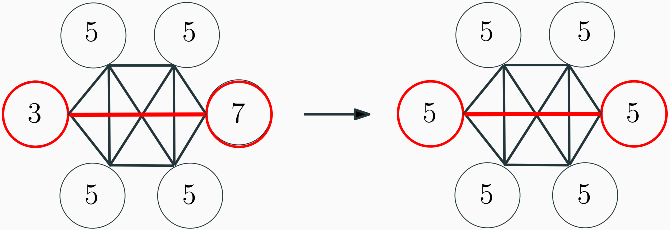
- Each node i has a value x_i. Let n be the total number of nodes
- In every round a **pair** of nodes (i, j) chosen **uniformly at random** interacts and exchange values
- The nodes quickly converge to the average





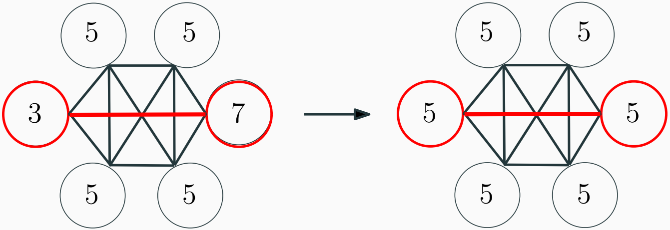
-
- How does the average change?

Change of the average



-
- How does the average change?
- It doesn't, i.e., $\varnothing^{(t)} = \varnothing^{(t-1)}$

Change of the average



-
- How does the average change?
- It doesn't, i.e., $\varnothing^{(t)} = \varnothing^{(t-1)}$
- In this setting it's well-known that the network quickly ε-converges in time

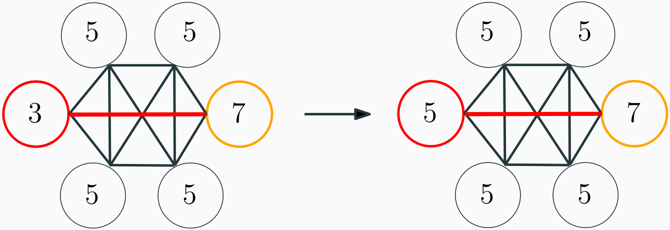
$$O\left(\frac{\log(K/\varepsilon)}{1-\lambda_2}\right)$$

K is the initial discrepancy

PART III: Uni-lateral Updates

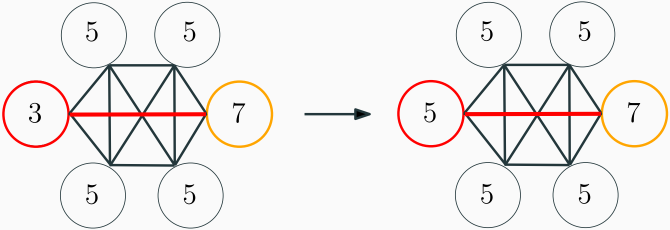
Uni-lateral Updates

- Enough Noise!
- What happens when only one side updates?



Uni-lateral Updates

- Enough Noise!
- What happens when only one side updates?



- The average changes ...
- Welcome to the uni-literal process
- In addition, this time we assume each node considers k other nodes at the same time.

Suppose people in the audience want to vote which gelato is better: strawberry or mango (although, we all know the answer is mango)

The Uni-Lateral Averaging Process

Suppose people in the audience want to vote which gelato is better: strawberry or mango (although, we all know the answer is mango)

the audience needs to agree which one to order

The Averaging Process

Attendees assign a value 0 to strawberry and 1 to mango

The Averaging Process

Attendees assign a value 0 to strawberry and 1 to mango

And each of them starts with either value based on their preference

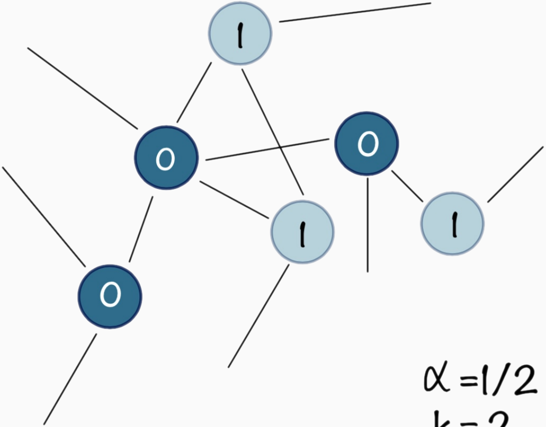
The Averaging Process

Attendees assign a value 0 to strawberry and 1 to mango

And each of them starts with either value based on their preference

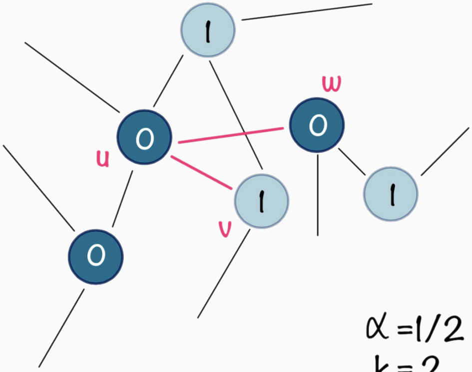
Then, they go for coffee and debate on which gelato is the best

The Averaging Process



$\alpha = 1/2$
$k = 2$

Among their acquaintances, every attendee, in turn, asks two of them their preference



The Averaging Process

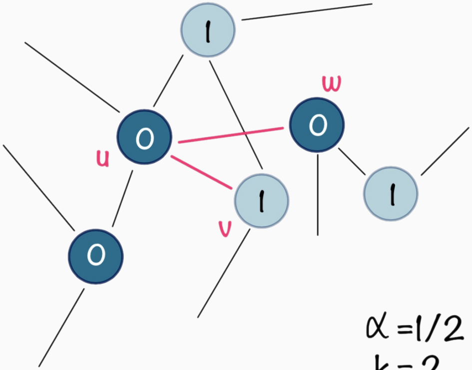
Among their acquaintances, every attendee, in turn, asks two of them their preference

They then update their preference to a weighted average:

$u' = \alpha u + (1-\alpha)(\frac{w+v}{2})$

1/4 = 1/2 * 0 + 1/2 * (0 + 1)/2

u' u w v



$\alpha = 1/2$

$k = 2$

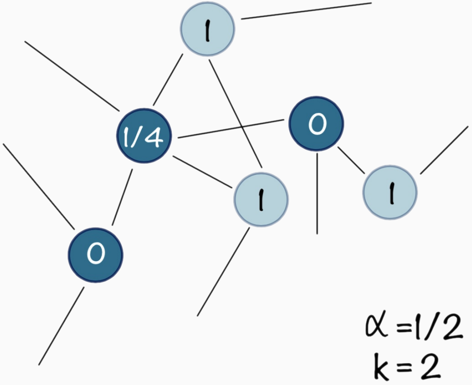
The Averaging Process

Among their acquaintances, every attendee, in turn, asks two of them their preference

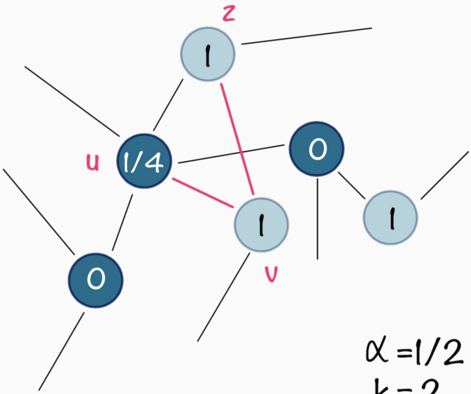
They then update their preference to a weighted average:

$u' = \alpha u + (1 - \alpha)(\frac{w+v}{2})$

$$\underset{u'}{1/4} = 1/2 * \underset{u}{0} + 1/2 * (\underset{w}{0} + \underset{v}{1})/2$$



One more time



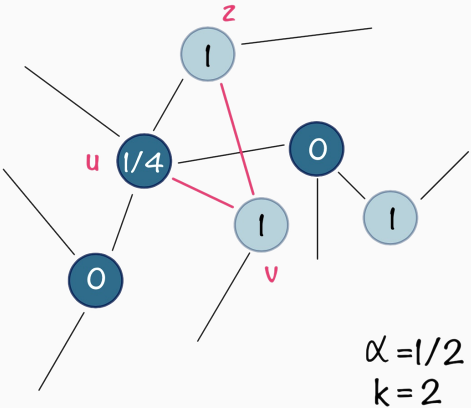
The Averaging Process

One more time

They then update their preference

$v' = \alpha v + (1-\alpha)(\frac{z+u}{2})$

$\underset{v'}{13/16} = 1/2 * \underset{v}{1} + 1/2 * (\underset{z}{1} + \underset{u}{1/4})/2$



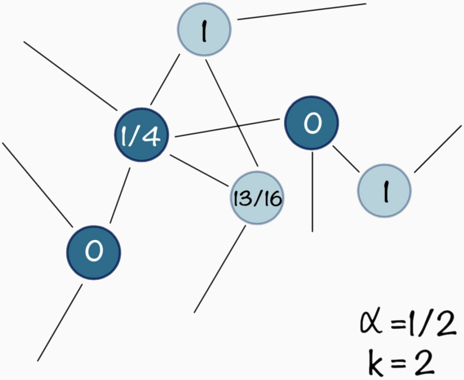
One more time

They then update their preference

$v' = \alpha v + (1 - \alpha)(\frac{z+u}{2})$

13/16 = 1/2 * 1 + 1/2 * (1 + 1/4)/2

v' v z u



Model and Contribution

Formally: We define

- $\alpha \in [0, 1]$ a weight on how strongly one holds on to their opinion
- k the number of attendees one talks to, namely $y_1, y_2, \ldots y_k$
- $\xi_u(t)$ the opinion of attendee u at time t. $\boldsymbol{\xi}(t)$ is the vector of all the opinions

The update step for node u is as follows

$$\boxed{\xi_u(t) = \alpha \xi_u(t-1) + \frac{(1-\alpha)}{k} \sum_{i=1}^{k} \xi_{y_i}(t-1),}$$

Some Motivations

This process is useful when multiple agents need to agree on some value and have

- Limited knowledge of the network
- Interact only with a subset of neighbors

It models real social dynamics, where individuals form opinions by talking to only a subset of their acquaintances.

Model and Contribution

Our work shows:

1. That our algorithm converges to a fixed point $F = \frac{1}{n}\sum_i^n \xi_i(0)$
2. Bounds on the convergence time T

$$T = O\left(\frac{n \log\left(\frac{n}{\epsilon} \cdot \|\xi(0)\|_2^2\right)}{(1-\lambda_2)}\right).$$

Where n is the number of nodes

Our work shows:

3. For G a regular graph, tight bounds on the variance of the fixed point

$$\mathbb{V}ar(F) = \Theta\left(\frac{\|\xi(0)\|_2^2}{n^2}\right)$$

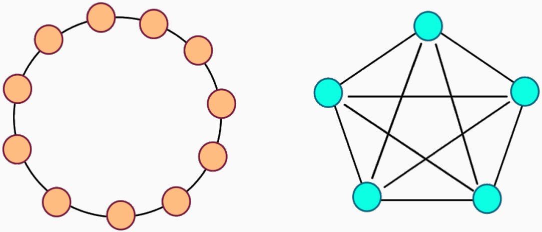
The Variance of the Fixed Point

This variance does not depend on the structure of G nor on k

The Variance of the Fixed Point

This variance does not depend on the structure of G nor on k

The variance for the cycle and for the clique are asymptotically the same



Scheme of the Proof

The variance in the Averaging Process is very hard to analyze

Scheme of the Proof

The variance in the Averaging Process is very hard to analyze

So we analyze the variance of two similar processes, and we show that these processes are all nicely related

Scheme of the Proof

The variance in the Averaging Process is very hard to analyze

So we analyze the variance of two similar processes, and we show that these processes are all nicely related

These are the *Diffusion Process* and the *Random Walk Process*

Scheme of the Proof

The variance in the Averaging Process is very hard to analyze

So we analyze the variance of two similar processes, and we show that these processes are all nicely related

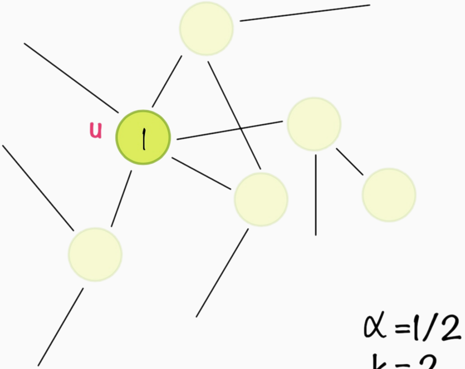
These are the *Diffusion Process* and the *Random Walk Process*

$$\mathbb{V}ar(F) \underset{\text{Step 1}}{\approx} \mathbb{V}ar(Diff) \underset{\text{Step 2}}{\approx} \mathbb{V}ar(RWalk) \underset{\text{Step 3}}{\approx} \Theta\left(\frac{\|\xi(0)\|_2^2}{n^2}\right)$$

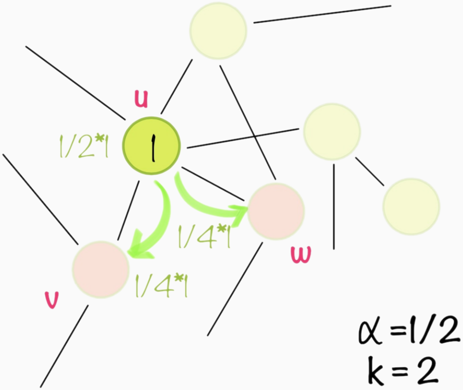
Scheme of the Proof

$$\mathbb{V}ar(F) \underset{\text{Step 1}}{\approx} \mathbb{V}ar(Diff) \underset{\text{Step 2}}{\approx} \mathbb{V}ar(RWalk) \underset{\text{Step 3}}{\approx} \Theta\left(\frac{\|\xi(0)\|_2^2}{n^2}\right)$$

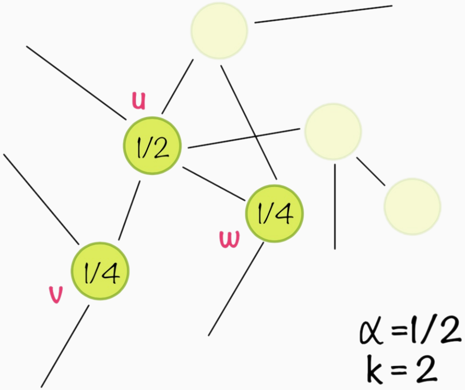
Start by placing a load **1** in every node



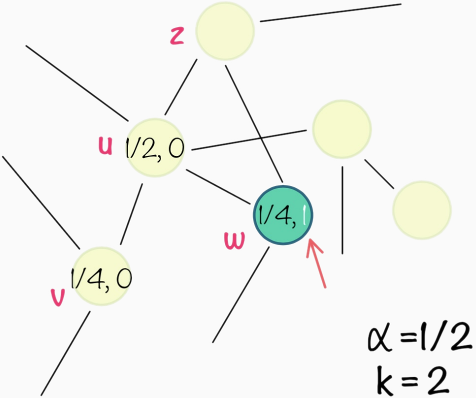
1 can be split up and diffused towards other nodes



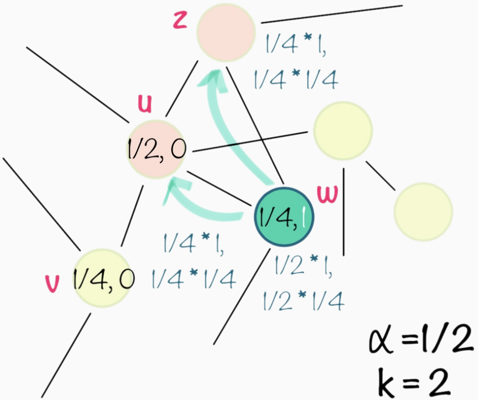
1 can be split up and diffused towards other nodes



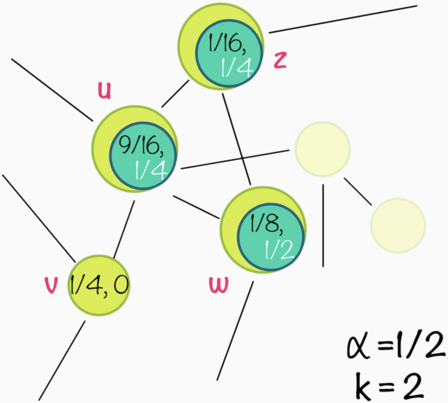
With loads from different nodes, this happens



With loads from different nodes, this happens

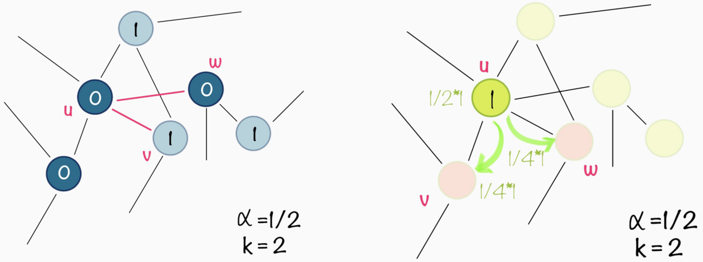


With loads from different nodes, this happens



In this way we can relate the Averaging Process and the Diffusion Process:

If we select nodes following a sequence $\mathcal{X} = (\mathcal{X}(1), \mathcal{X}(2), \ldots, \mathcal{X}(T))$, and we run the Averaging Process following $\mathcal{X}$ and the Diffusion Process following $\mathcal{X}^R$, then $\mathbb{V}ar(Avg) \approx \mathbb{V}ar(Diff)$

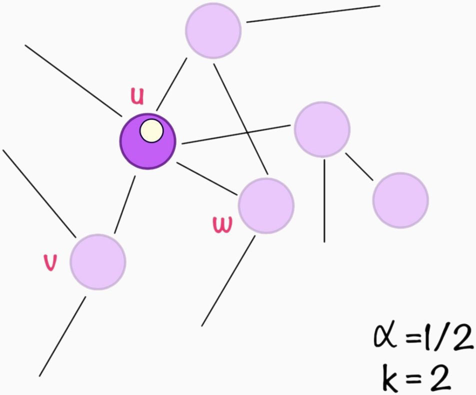


Scheme of the proof

We have

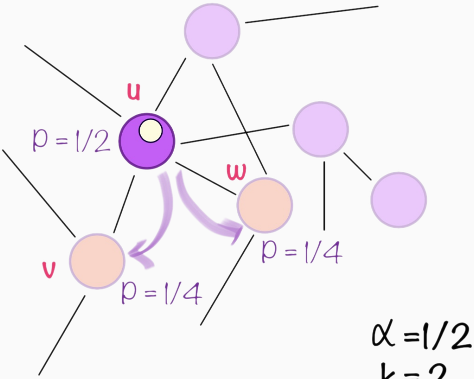
$$\underbrace{\mathbb{V}ar(Avg) \approx}_{\text{Step 1}} \underbrace{\mathbb{V}ar(Diff) \approx}_{\text{Step 2}} \underbrace{\mathbb{V}ar(RWalk) \approx}_{\text{Step 3}} \Theta\left(\frac{\|\xi(0)\|_2^2}{n^2}\right)$$

Start with placing a random walk q at every node

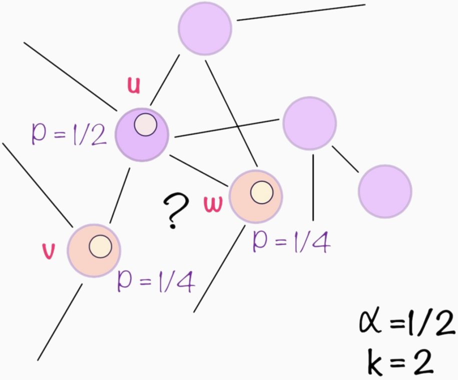


The Random Walk Process

Every q cannot be split, but it can either move to another node or stay put

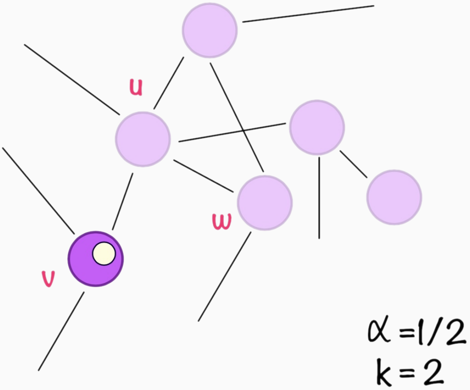


Every q cannot be split, but it can either move to another node or stay put

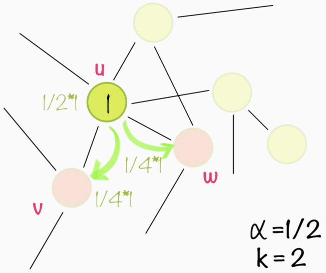


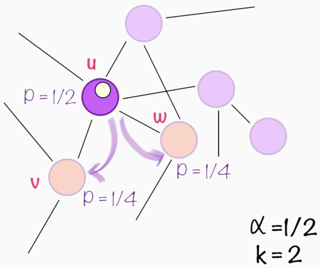
q decides to move w.p. $1-\alpha$,

whereas it stays put w.p. α



We see that we can relate the Diffusion Process and the Random Walk Process





The final location of a random walk will follow a probability distribution = the distribution of the corresponding diffused load

We have

$$\mathbb{V}ar(Avg) \underset{\text{Step 1}}{\approx} \mathbb{V}ar(Diff) \underset{\text{Step 2}}{\approx} \mathbb{V}ar(Rwalk) \underset{\text{Step 3}}{\approx} \Theta\left(\frac{\|\xi(0)\|_2^2}{n^2}\right)$$

The Q chain

Through step 2 we get to something like this

$$\mathbb{E}[X(t)Y(t)]$$

The Q chain

Through step 2 we get to something like this

$$\mathbb{E}[X(t)Y(t)]$$

With $X(t)$ and $Y(t)$ accounting for two random walks in the Random Walk Process

The Q chain

Through step 2 we get to something like this

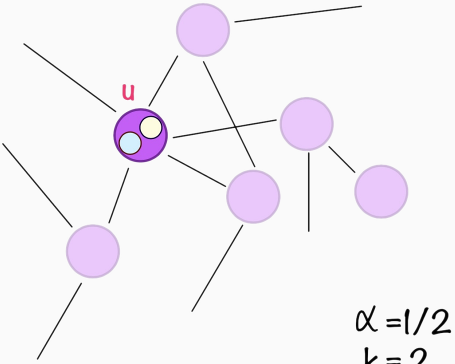
$$\mathbb{E}[X(t)Y(t)]$$

With $X(t)$ and $Y(t)$ accounting for two random walks in the Random Walk Process

Can we model how two random walks move on G?

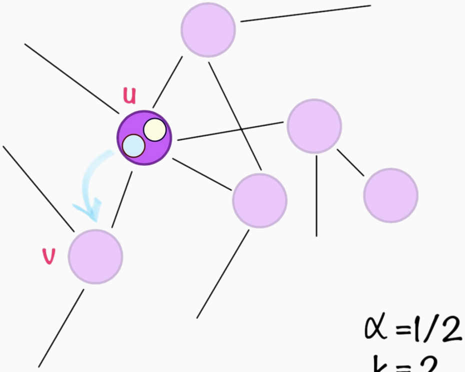
The Q Chain

Suppose we have two random walks on G

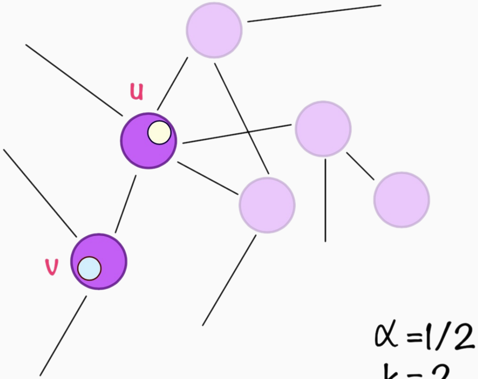


The Q Chain

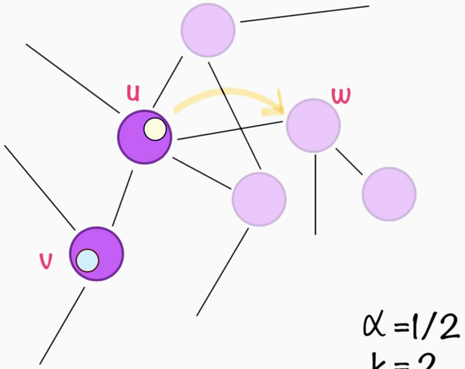
If a node containing any rw is sampled, then *every* walk in it can either move w.p. $(1-\alpha)$ or stay put w.p. α



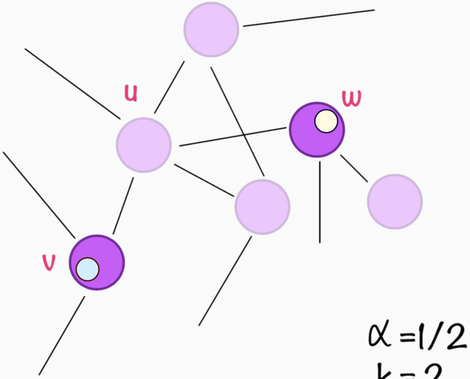
If a node containing any rw is sampled, then *every* walk in it can either move w.p. $(1-\alpha)$ or stay put w.p. α



If a node containing any rw is sampled, then *every* walk in it can either move w.p. $(1-\alpha)$ or stay put w.p. α



If a node containing any rw is sampled, then *every* walk in it can either move w.p. $(1 - \alpha)$ or stay put w.p. α



Every snapshot is a state of a Markov Chain

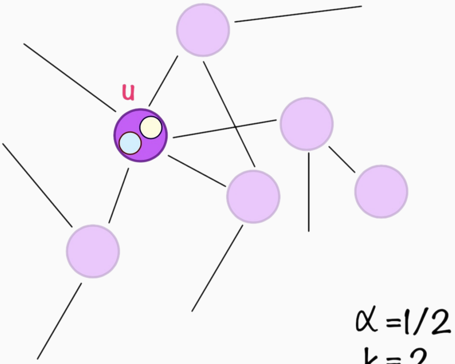
How can we describe the state?

Every snapshot is a state of a Markov Chain

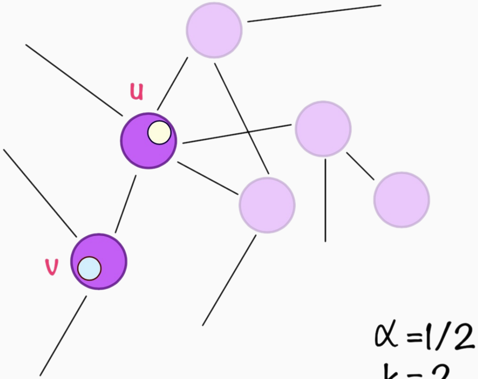
How can we describe the state?

By the relative distance between the two random walks

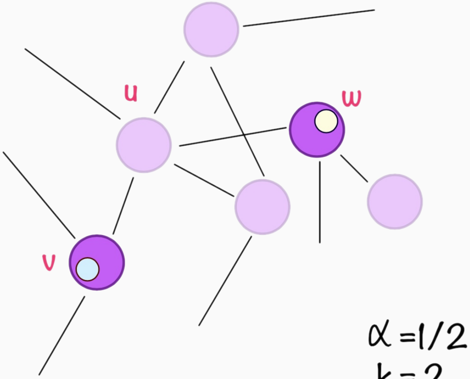
If the two rw are on the same node, they have distance 0



If the two rw are on neighboring nodes, they have distance 1

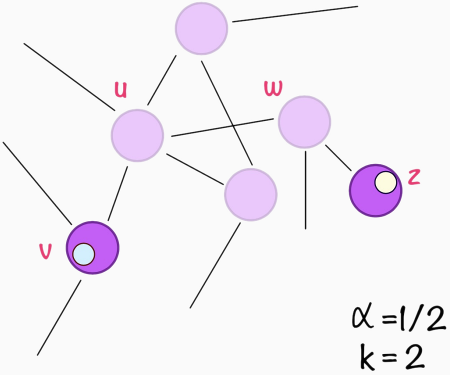


If the two rw are on nodes with a neighbor in common, they have distance 2



The Q Chain

And so on...



The Q Chain

Could we find the probability of each of these states presenting?

And could this probability remain unchanged over time? Unfortunately, the chain is not reversible :-(

The Q Chain

Could we find the probability of each of these states presenting?

And could this probability remain unchanged over time? Unfortunately, the chain is not reversible :-(

This Markov Chain is *irreducible* and *aperiodic*

The Q Chain

Could we find the probability of each of these states presenting?
And could this probability remain unchanged over time? Unfortunately, the chain is not reversible :-(

This Markov Chain is *irreducible* and *aperiodic*

Hence, it has a unique stationary distribution

The Q Chain

Could we find the probability of each of these states presenting?

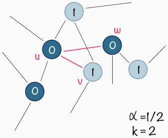
And could this probability remain unchanged over time? Unfortunately, the chain is not reversible :-(

This Markov Chain is *irreducible* and *aperiodic*

Hence, it has a unique stationary distribution 🥳

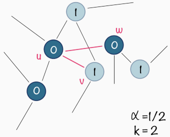
We finally have

Our original process: The Averaging Process

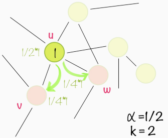


We finally have

Our original process: The Averaging Process

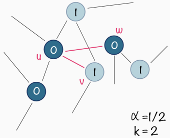


The Diffusion Process

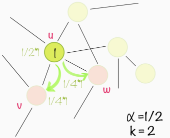


We finally have

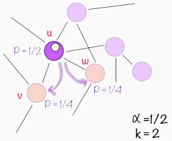
Our original process: The Averaging Process



The Diffusion Process

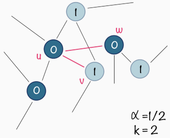


The Random Walk Process

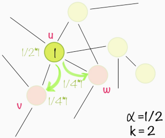


We finally have

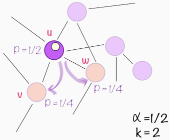
Our original process: The Averaging Process



The Diffusion Process



The Random Walk Process



The Stationary Distribution of the Q chain

We arrive to the result:

$$\mathbb{V}ar(F) = \Theta\left(\frac{\|\xi(0)\|_2^2}{n^2}\right)$$

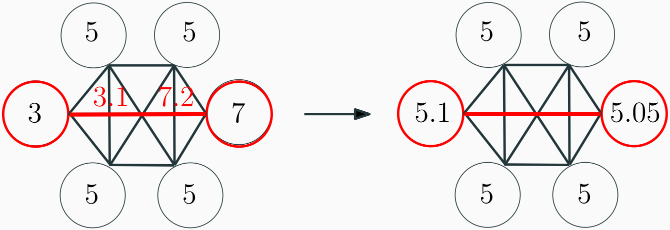
For G a regular graph.

PART III: Bi-lateral Updates with noise (Noidy Conmunixatipn)

- Bi-lateral averaging with noise
- In a nutshell: Averaging population over a noisy channel
- In rounds: two nodes exchange values and average.
- Catch: There is a noisy channel

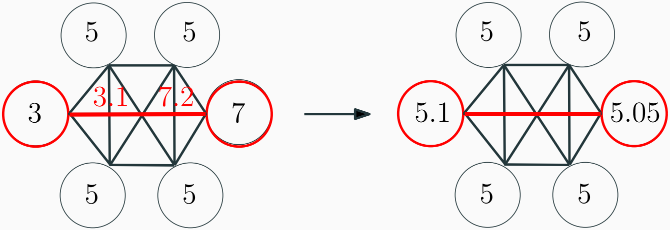
What Happens When we Add Noise?

- Say we have n nodes
- Let $x_i^{(t)}$ be the value of node i at time t. $N_1, N_2 \sim \mathcal{N}(0, 1)$



What Happens When we Add Noise?

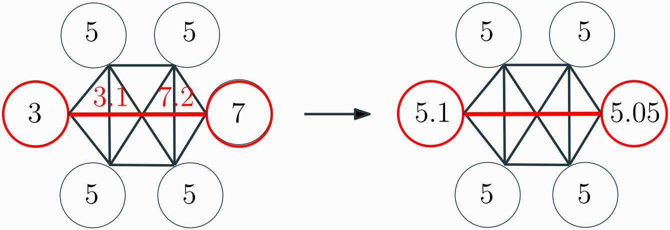
- Say we have n nodes
- Let $x_i^{(t)}$ be the value of node i at time t. $N_1, N_2 \sim \mathcal{N}(0,1)$



- $x_1^{(t)} = \frac{x_1^{(t-1)} + (x_2^{(t-1)} + N_1)}{2}$

What Happens When we Add Noise?

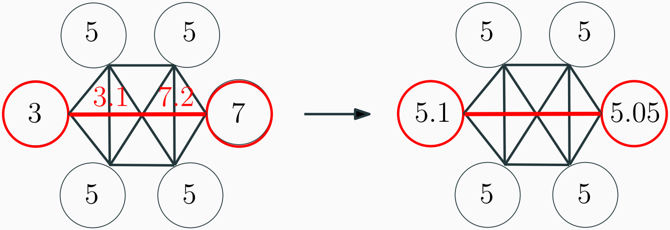
- Say we have n nodes
- Let $x_i^{(t)}$ be the value of node i at time t. $N_1, N_2 \sim \mathcal{N}(0, 1)$



- $x_1^{(t)} = \frac{x_1^{(t-1)} + (x_2^{(t-1)} + N_1)}{2}$
- $x_2^{(t)} = \frac{(x_1^{(t-1)} + N_2) + x_2^{(t-1)}}{2}$

What Happens When we Add Noise?

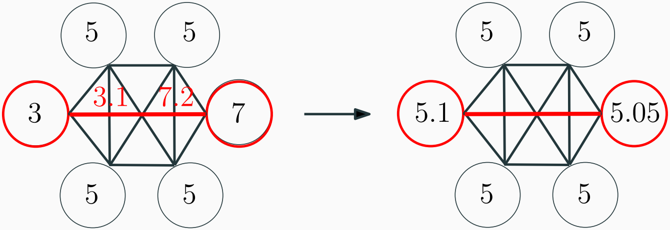
- Say we have n nodes
- Let $x_i^{(t)}$ be the value of node i at time t. $N_1, N_2 \sim \mathcal{N}(0,1)$



- $x_1^{(t)} = \frac{x_1^{(t-1)} + (x_2^{(t-1)} + N_1)}{2}$
- $x_2^{(t)} = \frac{(x_1^{(t-1)} + N_2) + x_2^{(t-1)}}{2}$
- $\varnothing^{(t)} = \frac{\sum_i^n x_i^{(t-1)} + \frac{N_1 + N_2}{2}}{n} = \varnothing^{(t-1)} + \frac{N_1 + N_2}{2n}$

What Happens When we Add Noise?

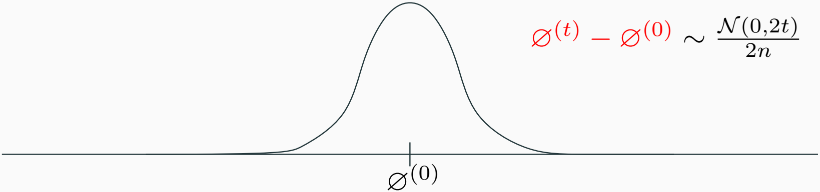
- Say we have n nodes
- Let $x_i^{(t)}$ be the value of node i at time t. $N_1, N_2 \sim \mathcal{N}(0,1)$



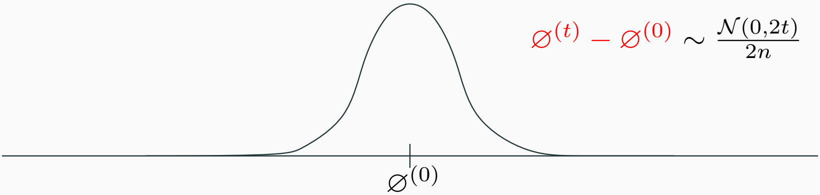
- $x_1^{(t)} = \frac{x_1^{(t-1)} + (x_2^{(t-1)} + N_1)}{2}$
- $x_2^{(t)} = \frac{(x_1^{(t-1)} + N_2) + x_2^{(t-1)}}{2}$
- $\varnothing^{(t)} = \frac{\sum_i^n x_i^{(t-1)} + \frac{N_1 + N_2}{2}}{n} = \varnothing^{(t-1)} + \frac{N_1 + N_2}{2n}$
- $= \varnothing^{(0)} + \sum_{\tau=1}^{2t} \frac{N_\tau}{2n}$

Are we doomed?

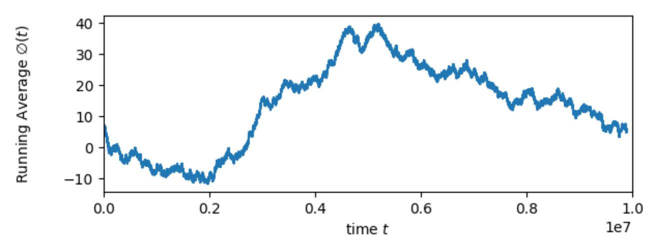
- $\varnothing^{(t)} = \varnothing^{(0)} + \sum_{\tau=1}^{2t} \frac{N_\tau}{2n}$
- From this we can conclude



$$\varnothing^{(t)} - \varnothing^{(0)} \sim \frac{\mathcal{N}(0,2t)}{2n}$$

- 

- With constant probability, $|\varnothing^{(t)} - \varnothing^{(0)}| = \Omega(\sqrt{t}/n)$.
- This diverges for large t … :(



Maybe the nodes can at least 'agree'?

- Consider the *running average*

$$\varnothing_{running}^{(t)} = \sum_i x_i^{(t)} / n$$

  Example: mean salary over time

- In contrast, the *initial average* is

$$\varnothing_{init} = \sum_i x_i^{(0)} / n$$

- So what about the running average?

Maybe the nodes can at least ‘agree’?

- Consider the *running average*

$$\varnothing^{(t)}_{running} = \sum_i x_i^{(t)} / \boldsymbol{n}$$

- There will always be some noise
- We won’t be able to ensure that all nodes have eventually this value
- However, we can show that we can ‘converge’ to something close

Quality Measure

- We care about the following two quantities ... (**squared L2-norm**)
  - *running total sum of squares*

$$TSS_{running}(t) = \sum_i (x_i^{(t)} - \varnothing^{(t)})^2,$$

  - *inital total sum of squares*

$$TSS_{initial}(t) = \sum_i (x_i^{(t)} - \varnothing^{(0)})^2,$$

Related Work

- Studied a lot in control theory and sensor networks (though mostly parallel rounds)
- Eventual 'convergence' of the running total sum of squares

$$TSS_{running}(t) = \sum_i (x_i^{(t)} - \varnothing^{(t)})^2,$$

- Divergence of $|\varnothing^{(t)} - \varnothing^{(0)}|$

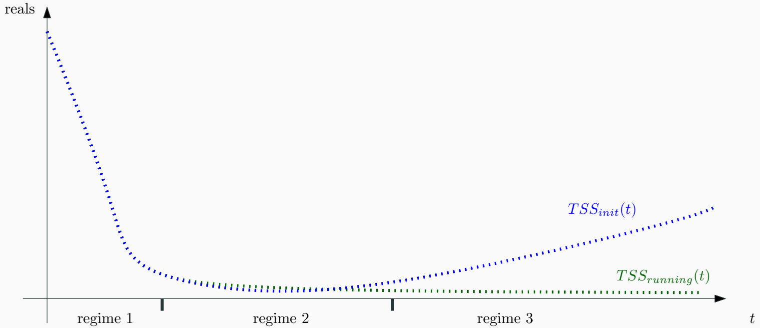
Take-Home Message

- Studied a lot in control theory and sensor networks (though mostly parallel rounds)
- Eventual 'convergence' of the running total sum of squares

$$TSS_{running}(t) = \sum_i (x_i^{(t)} - \varnothing^{(t)})^2,$$

- Divergence of $\varnothing_{init}$ **at a much lower rate though!**

Take-Home Message



- $TSS_{initial}(t)$ is large at the beginning (regime 1)
- small for a long time (regime 2)
- large as t goes to infinity (regime 3)

Proof Idea

- We have

$$TSS_{initial}(t) = TSS_{running}(t) + n(\varnothing^{(t)} - \varnothing^{(0)}).$$

- Give precise bounds on $n(\varnothing^{(t)} - \varnothing^{(0)}) \sim \mathcal{N}(0, 2t)$. Easy.
- Give precise bounds on $TSS_{running}$. Not so easy :(

Bounding $TSS_{running}$

- Step 1: look at change in one round. Long formula

Bounding $TSS_{running}$

- Step 1: look at change in one round. Long formula
- Step 2: Consider interval $[t_0, t_1]$ with $t = t_1 - t_0$ and bound the change (using Step 1)

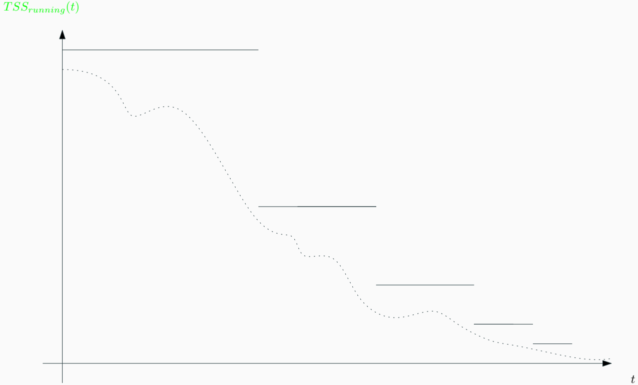
$$TSS_{running}(t_1) \leq \left(1 - \frac{S_1}{t}\right)^t TSS_{running}(t_0) + S2 + S3,$$

where $S_1(t), S_2(t), S_3(t)$ follow some non-trivial probability distributions.

- the longer the interval, the better the shrinking of the first part
- but the worse the additive error of S_2 and S_3

Analysis

- Use smaller and smaller intervals



- Solid lines are bounds that hold w.h.p.
- Dotted lines is the actual size of the potential

Theorem:

After little time, $TSS_{running}$ is small.

Corollary:

After little time, $TSS_{initial}(t)$ is also small. But don't wait too long, since it'll explode eventually.

Proof:

$$TSS_{initial}(t) = TSS_{running}(t) + n(\varnothing^{(t)} - \varnothing^{(0)}),$$

Extensions

- We consider (almost) any noise function with zero-mean
- Works for many distributions (zero-mean versions): sub-Gaussian, Poisson, geometric, ...

Extensions

- In particular discrete distributions
- Nodes can use rounding.
- Say node with value x_1 receives x_2. Then,

$$x_1' = \begin{cases} \lceil \frac{x_1+x_2}{2} \rceil & \text{w.p. } 1/2 \\ \lfloor \frac{x_1+x_2}{2} \rfloor & \text{otherwise} \end{cases}$$

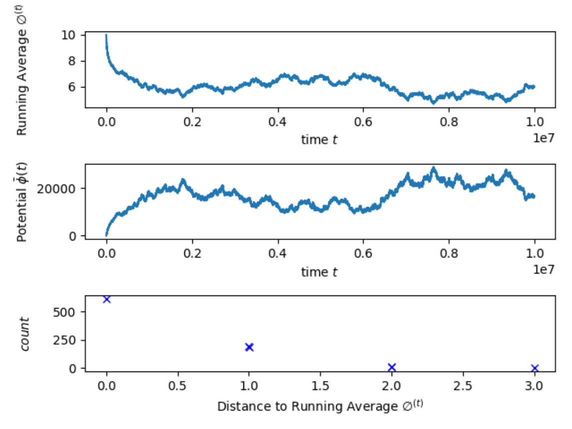
- Motivation: Bounded memory

Extensions

- Everything still works in the synchronous setting where all nodes communicate at the same time (in pairs of two)

Bounded Values

- Values only live in $[0, 10]$. Initially all nodes have value 10



-
- The average drifts super fast away from the borders. Due to the bias
- Open questions to prove this.

Thank you!

Quick recap

1. Bi-lateral averaging (w/o noise):
   $\varnothing(t) = \varnothing(0)$; $x_i^{(t)}$ converge to $\varnothing(0)$
2. Uni-lateral averaging (w/o noise):
   $\varnothing(t) \neq \varnothing(0)$; $x_i^{(t)}$ converge to fixed point F
3. Bi-lateral averaging with noise:
   $\varnothing(t) \neq \varnothing(0)$; $x_i^{(t)}$ are close to $\varnothing(t)$

randomlab.uk/