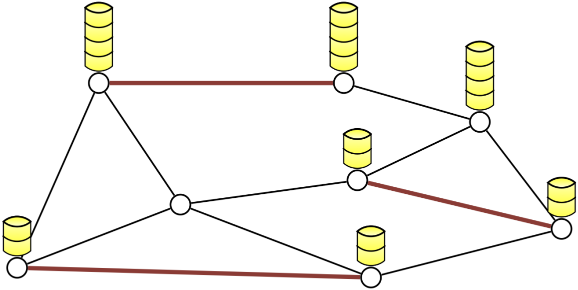


ADGA 2013

2nd Workshop on Advances on Distributed Graph Algorithms

Distributed Load Balancing on Graphs

Thomas Sauerwald

UNIVERSITY OF CAMBRIDGE

Outline

Introduction

Load Balancing on Hypercubes

Load Balancing on Arbitrary Graphs

Conclusions

Outline

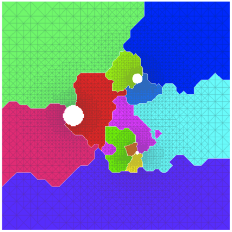
Introduction

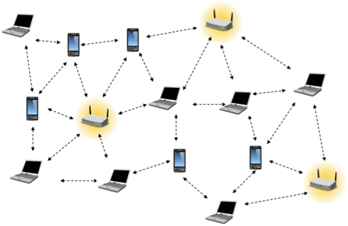
Load Balancing on Hypercubes

Load Balancing on Arbitrary Graphs

Conclusions

Load Balancing

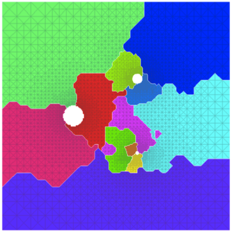


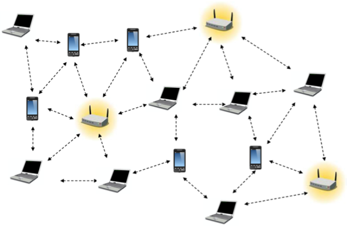


Applications

- Numerical Simulations
- Traffic in Communication Network
- Data Management in P2P network

Load Balancing





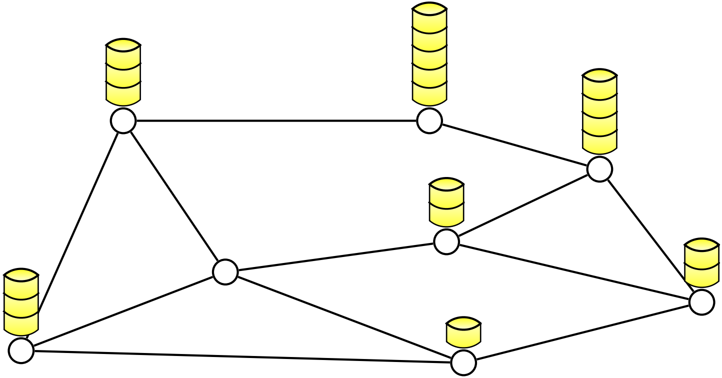
Applications

- Numerical Simulations
- Traffic in Communication Network
- Data Management in P2P network

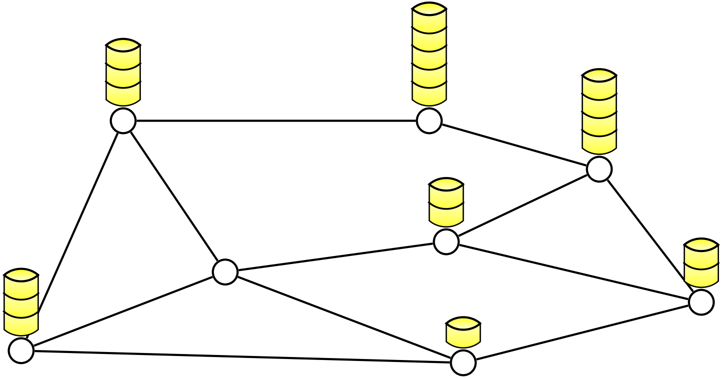
Conditions

- network structure and load distribution unknown
- node can only communicate with neighbors

Discrete Load Balancing with Unit-Size-Token



Discrete Load Balancing with Unit-Size-Token

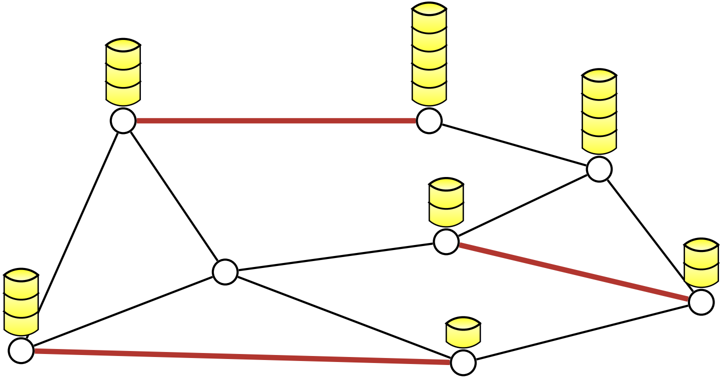


Protocol

For every round $t = 1, 2, 3, \ldots$

- Generate a matching
- Matched vertices average load

Discrete Load Balancing with Unit-Size-Token

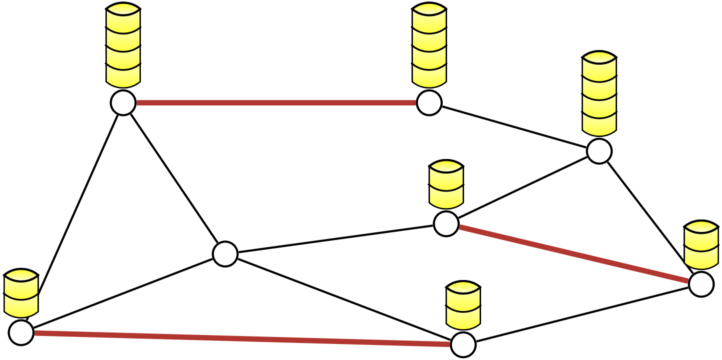


Protocol

For every round $t = 1, 2, 3, \ldots$

- Generate a matching
- Matched vertices average load

Discrete Load Balancing with Unit-Size-Token

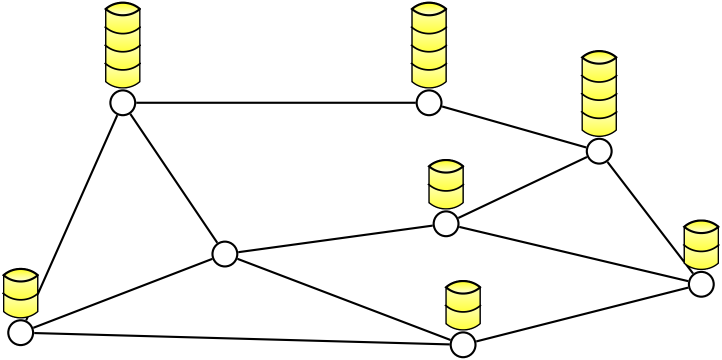


Protocol

For every round $t = 1, 2, 3, \ldots$

- Generate a matching
- Matched vertices average load

Discrete Load Balancing with Unit-Size-Token

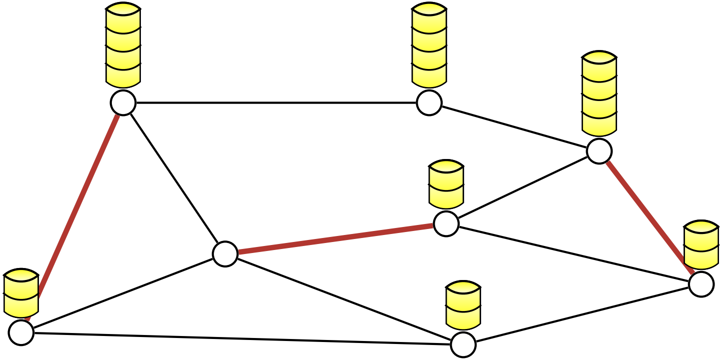


Protocol

For every round $t = {\color{red}1}, 2, 3, \ldots$

- Generate a matching
- Matched vertices average load

Discrete Load Balancing with Unit-Size-Token

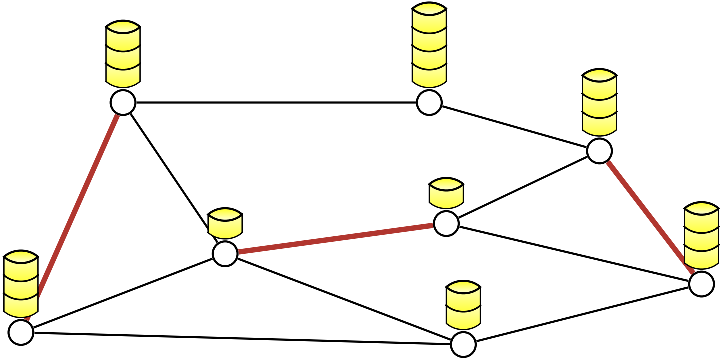


Protocol

For every round $t = 1, 2, 3, \ldots$

- Generate a matching
- Matched vertices average load

Discrete Load Balancing with Unit-Size-Token

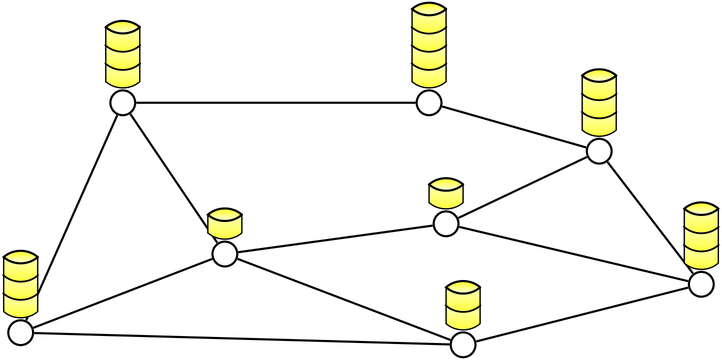


Protocol

For every round $t = 1, 2, 3, \ldots$

- Generate a matching
- Matched vertices average load

Discrete Load Balancing with Unit-Size-Token

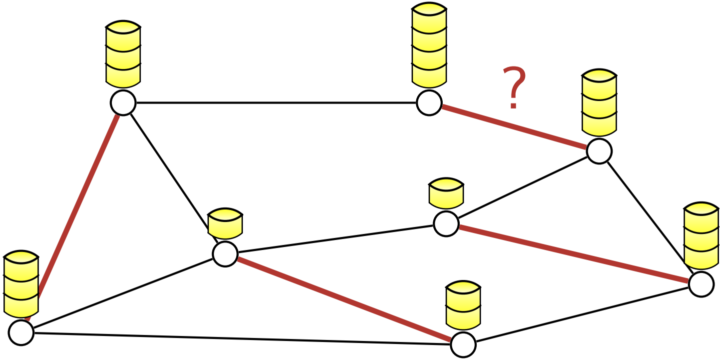


Protocol

For every round $t = 1, 2, 3, \ldots$

- Generate a matching
- Matched vertices average load

Discrete Load Balancing with Unit-Size-Token

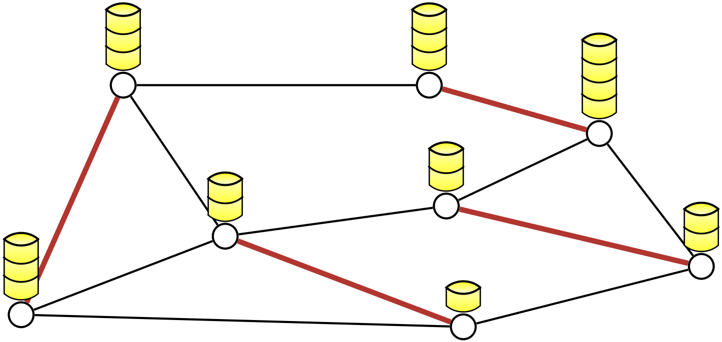


Protocol

For every round $t = 1, 2, 3, \ldots$

- Generate a matching
- Matched vertices average load

Discrete Load Balancing with Unit-Size-Token

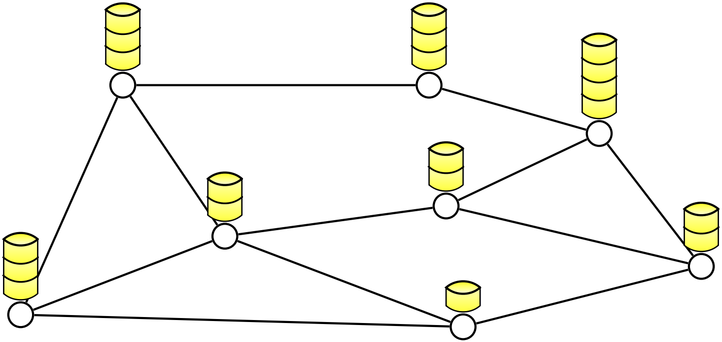


Protocol

For every round $t = 1, 2, 3, \ldots$

- Generate a matching
- Matched vertices average load

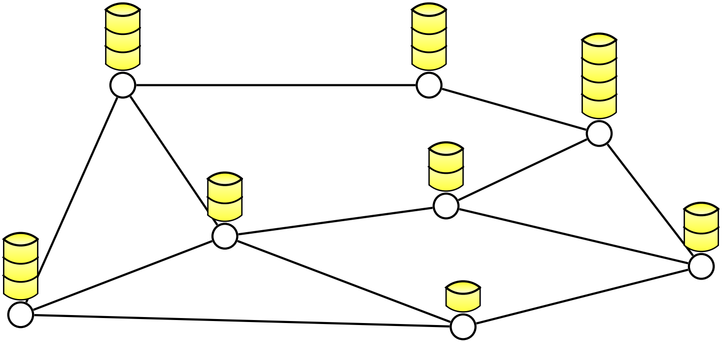
Discrete Load Balancing with Unit-Size-Token



Conditions

- ✓ Graphs's structure and load is unknown to every node
- ✓ Nodes can only communicate with neighbors

Discrete Load Balancing with Unit-Size-Token

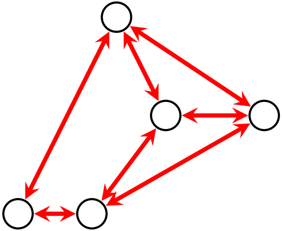


How should we generate the matchings?

Communication Models

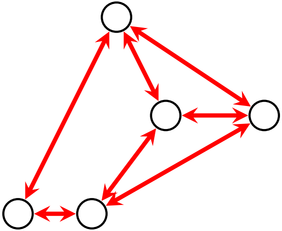
Diffusion

+ natural
– high communication

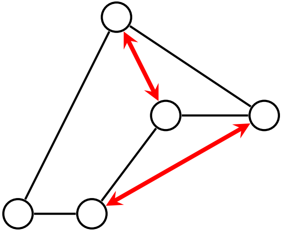


Communication Models

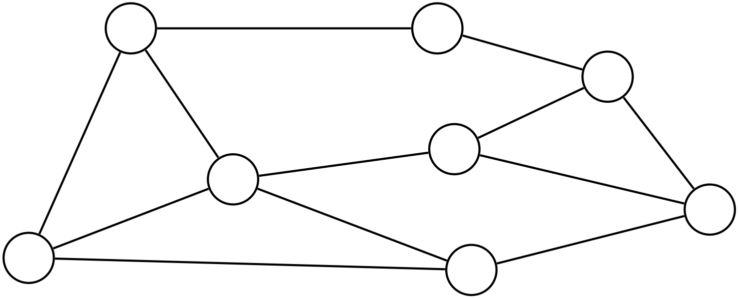
Diffusion
- + natural
- – high communication



Matching Model
- + less communication
- + monotone
- – matchings have to be specified

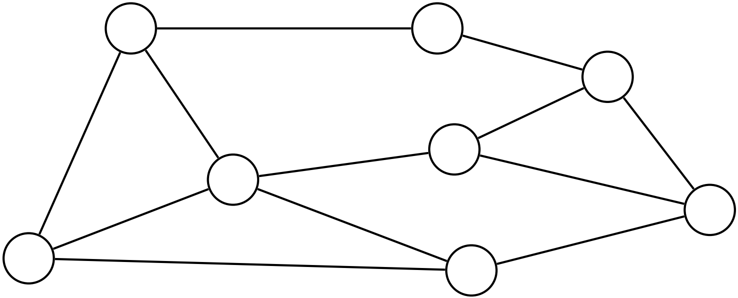


Generating Matching Using Edge Coloring



Balancing Circuit Model

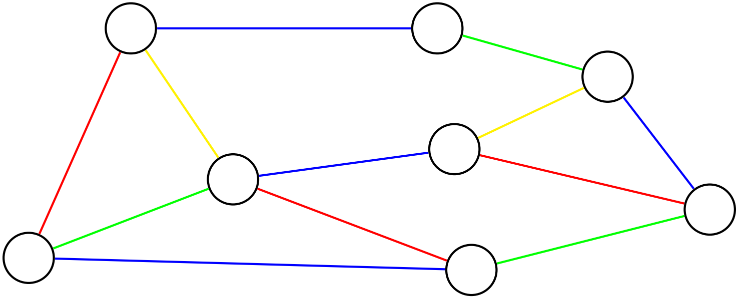
Generating Matching Using Edge Coloring



Balancing Circuit Model

1. Take an edge coloring with $c \leqslant \text{maxdeg} + 1$ colors.

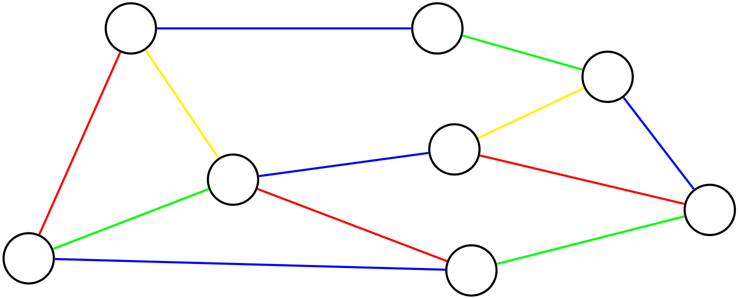
Generating Matching Using Edge Coloring



Balancing Circuit Model

1. Take an edge coloring with $c \leqslant \text{maxdeg} + 1$ colors.

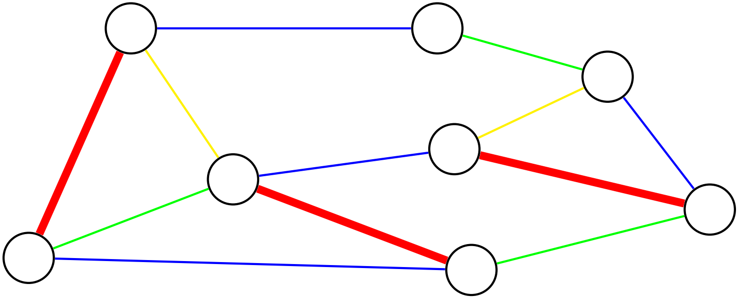
Generating Matching Using Edge Coloring



Balancing Circuit Model

1. Take an edge coloring with $c \leqslant \text{maxdeg} + 1$ colors.
2. In round r use matching induced by color class $r \bmod c$.

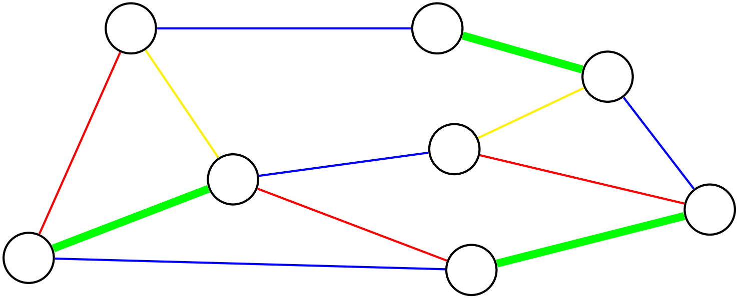
Generating Matching Using Edge Coloring



Balancing Circuit Model

1. Take an edge coloring with $c \leqslant \text{maxdeg} + 1$ colors.
2. In round r use matching induced by color class $r \bmod c$.

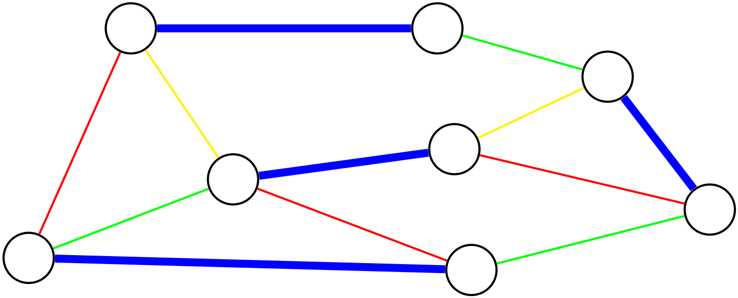
Generating Matching Using Edge Coloring



Balancing Circuit Model

1. Take an edge coloring with $c \leqslant \text{maxdeg} + 1$ colors.
2. In round r use matching induced by color class $r \bmod c$.

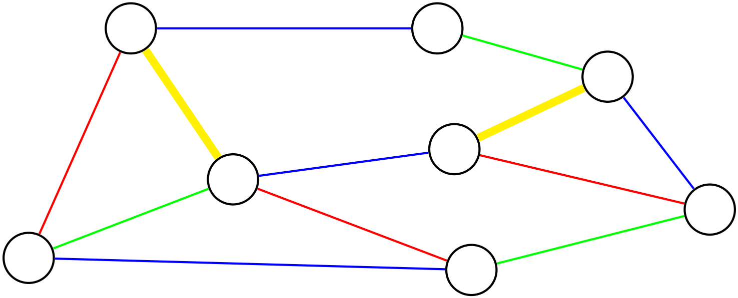
Generating Matching Using Edge Coloring



Balancing Circuit Model

1. Take an edge coloring with $c \leqslant \text{maxdeg} + 1$ colors.
2. In round r use matching induced by color class $r \bmod c$.

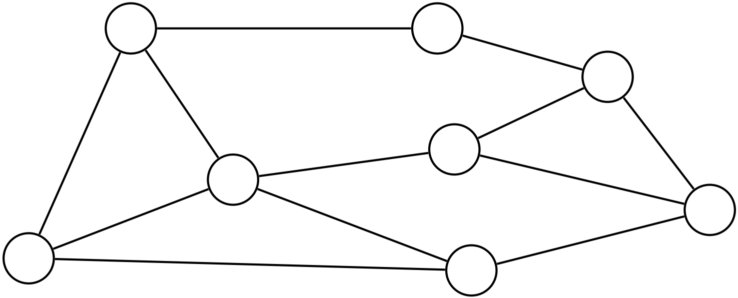
Generating Matching Using Edge Coloring



Balancing Circuit Model

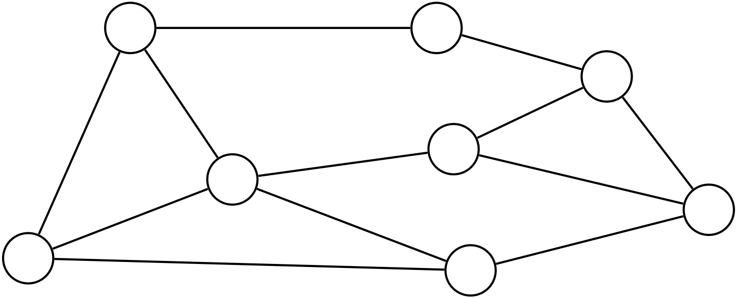
1. Take an edge coloring with $c \leqslant \text{maxdeg} + 1$ colors.
2. In round r use matching induced by color class $r \bmod c$.

Generating Matching Using Randomization



Random Matching (Boyd et al., 2006)

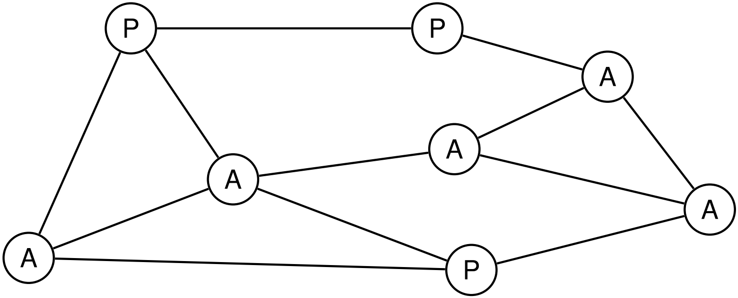
Generating Matching Using Randomization



Random Matching (Boyd et al., 2006)

1. First, every node becomes active (or passive) with prob. $1/2$.

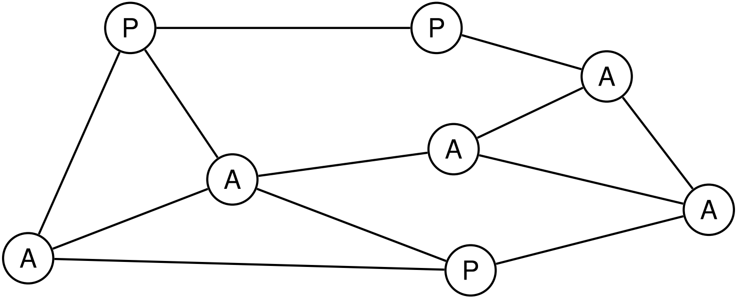
Generating Matching Using Randomization



Random Matching (Boyd et al., 2006)

1. First, every node becomes active (or passive) with prob. $1/2$.

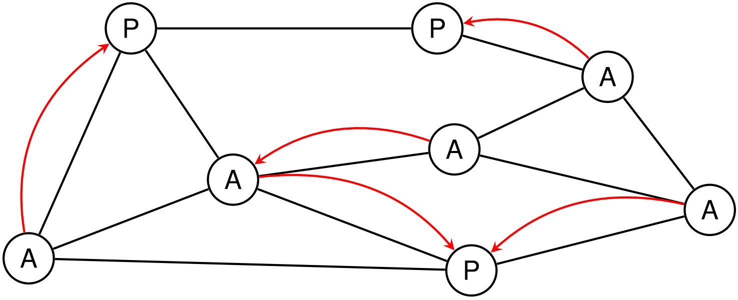
Generating Matching Using Randomization



Random Matching (Boyd et al., 2006)

1. First, every node becomes active (or passive) with prob. $1/2$.
2. Every active node u contacts $v \in N(u)$ with prob. $\frac{1}{\text{maxdeg}}$

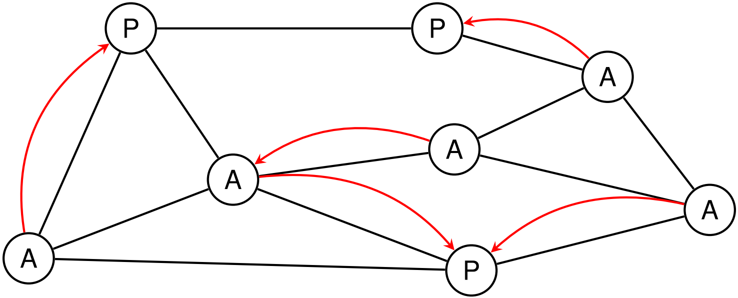
Generating Matching Using Randomization



Random Matching (Boyd et al., 2006)

1. First, every node becomes active (or passive) with prob. $1/2$.
2. Every active node u contacts $v \in N(u)$ with prob. $\frac{1}{\text{maxdeg}}$

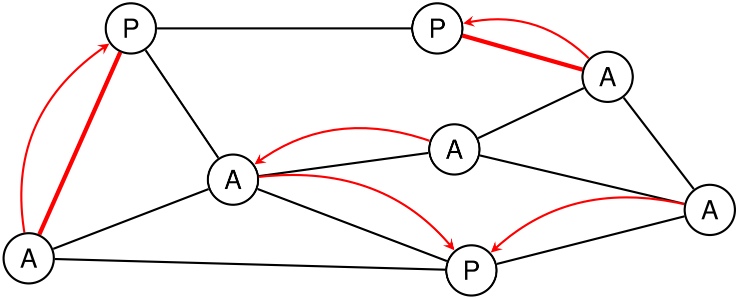
Generating Matching Using Randomization



Random Matching (Boyd et al., 2006)

1. First, every node becomes active (or passive) with prob. $1/2$.
2. Every active node u contacts $v \in N(u)$ with prob. $\frac{1}{\text{maxdeg}}$
3. An active node contacting a passive node which is not contacted by any other node form a pair in the matching

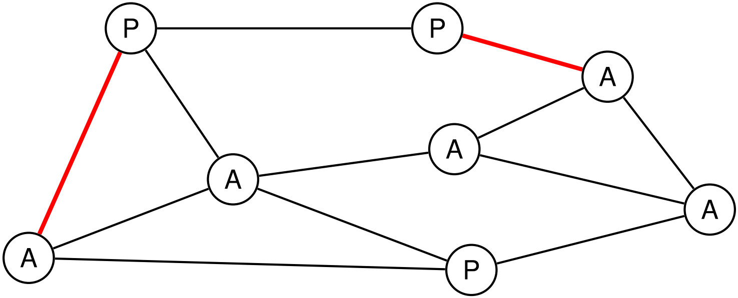
Generating Matching Using Randomization



Random Matching (Boyd et al., 2006)

1. First, every node becomes active (or passive) with prob. $1/2$.
2. Every active node u contacts $v \in N(u)$ with prob. $\frac{1}{\text{maxdeg}}$
3. An active node contacting a passive node which is not contacted by any other node form a pair in the matching

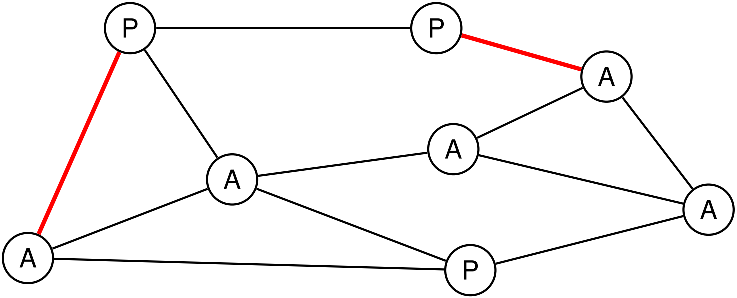
Generating Matching Using Randomization



Random Matching (Boyd et al., 2006)

1. First, every node becomes active (or passive) with prob. $1/2$.
2. Every active node u contacts $v \in N(u)$ with prob. $\frac{1}{\text{maxdeg}}$
3. An active node contacting a passive node which is not contacted by any other node form a pair in the matching

Generating Matching Using Randomization



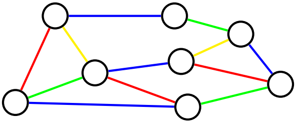
Crucial Properties:

- An edge $\{u, v\} \in E$ is included with prob. $\approx \frac{1}{\text{maxdeg}}$
- Matchings in different rounds are generated independently

Balancing Circuit vs. Random Matching

Balancing Circuit (Dimension Exchange)

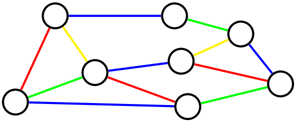
- graphs with structure (grids and hypercubes)
- edge-coloring and order may affect convergence (dense graphs)



Balancing Circuit vs. Random Matching

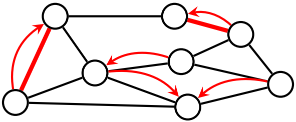
Balancing Circuit (Dimension Exchange)

- graphs with structure (grids and hypercubes)
- edge-coloring and order may affect convergence (dense graphs)



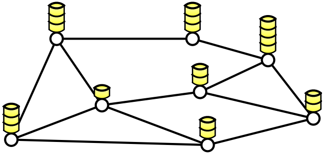
Random Matching

- applicable to any graph
- convergence captured by the spectral gap of the graph



Smoothness of the Load Distribution

- let $x \in \mathbb{R}^n$ be a load vector
- $\overline{x}$ denotes the average load

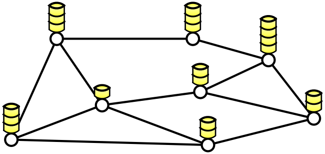


Smoothness of the Load Distribution

- let $x \in \mathbb{R}^n$ be a load vector
- $\overline{x}$ denotes the average load

Metrics

- ℓ_2-norm: $\|x - \overline{x}\|_2 = \sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}$
- makespan: $\max_{i=1}^{n} x_i$
- discrepancy: $\max_{i=1}^{n} x_i - \min_{i=1}^{n} x_i$.

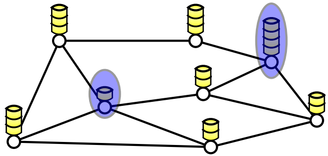


Smoothness of the Load Distribution

- let $x \in \mathbb{R}^n$ be a load vector
- $\overline{x}$ denotes the average load

Metrics

- ℓ_2-norm: $\|x - \overline{x}\|_2 = \sqrt{\sum_{i=1}^n (x_i - \overline{x})^2}$
- makespan: $\max_{i=1}^n x_i$
- discrepancy: $\max_{i=1}^n x_i - \min_{i=1}^n x_i$.



Random Matching in Continuous Case

Ghosh, Muthukrishnan, 1994

- Let $\Phi^t = \sum_{i=1}^{n} \left(x_i^t - \overline{x}\right)^2$. Then,

$$\mathbf{E}\left[\Phi^t - \Phi^{t+1}\right] \geqslant$$

Random Matching in Continuous Case

Ghosh, Muthukrishnan, 1994

- Let $\Phi^t = \sum_{i=1}^{n} \left(x_i^t - \overline{x}\right)^2$. Then,

$$\mathbf{E}\left[\Phi^t - \Phi^{t+1}\right] \geqslant \frac{1-\lambda}{8} \cdot \Phi^t,$$

where $\lambda \in (0, 1]$ is the spectral expansion.

Random Matching in Continuous Case

Ghosh, Muthukrishnan, 1994

- Let $\Phi^t = \sum_{i=1}^n \left(x_i^t - \overline{x}\right)^2$. Then,

$$\mathbf{E}\left[\Phi^t - \Phi^{t+1}\right] \geqslant \frac{1-\lambda}{8} \cdot \Phi^t,$$

where $\lambda \in (0,1]$ is the spectral expansion.

$\Rightarrow$ For any initial load vector with discrepancy K, the discrepancy is at most ϵ w.p. $1 - n^{-1}$ after $\mathcal{O}\left(\frac{\log(n \cdot \frac{K}{\epsilon})}{1-\lambda}\right)$ rounds.

Random Matching in Continuous Case

Ghosh, Muthukrishnan, 1994

- Let $\Phi^t = \sum_{i=1}^{n} \left(x_i^t - \overline{x}\right)^2$. Then,

$$\mathbf{E}\left[\Phi^t - \Phi^{t+1}\right] \geqslant \frac{1-\lambda}{8} \cdot \Phi^t,$$

where $\lambda \in (0,1]$ is the spectral expansion.

$\Rightarrow$ For any initial load vector with discrepancy K, the discrepancy is at most ϵ w.p. $1 - n^{-1}$ after $\mathcal{O}\left(\frac{\log(n \cdot \frac{K}{\epsilon})}{1-\lambda}\right)$ rounds.

- Speed of convergence essentially the same as for FOS diffusion
- Even though load is moved only along a subset of edges, the convergence is in terms of the global properties

Random Matching in Continuous Case

Ghosh, Muthukrishnan, 1994

- Let $\Phi^t = \sum_{i=1}^{n} \left(x_i^t - \overline{x}\right)^2$. Then,

$$\mathbf{E}\left[\Phi^t - \Phi^{t+1}\right] \geqslant \frac{1-\lambda}{8} \cdot \Phi^t,$$

$\Rightarrow$ For any initial load vector with discrepancy K, the discrepancy is at most ϵ w.p. $1 - n^{-1}$ after $\mathcal{O}\left(\frac{\log(n \cdot \frac{K}{\epsilon})}{1-\lambda}\right)$ rounds.

Continuous case:

- Well understood and rapid convergence
- less realistic as tokens can be divided arbitrarily often

- Speed of convergence essentially the same as for FOS diffusion
- Even though load is moved only along a subset of edges, the convergence is in terms of the global properties

Discrete vs. Continuous Load Balancing

What is the relation between the discrete and continuous case?

Subramannian and Scherson 1994, Ghosh, Leighton, Maggs, Muthukrishnan, Plaxton, Rajaraman, Richa, Tarjan and Zuckerman 1995, Lovasz and Winkler 1995, Muthukrishnan, Ghosh and Schultz 1996, Rabani, Sinclair and Wanka 1998.

Outline

Introduction

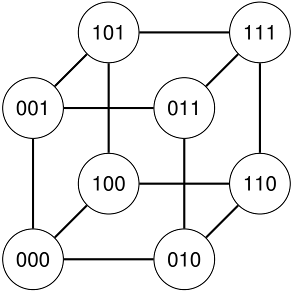
Load Balancing on Hypercubes

Load Balancing on Arbitrary Graphs

Conclusions

Hypercube

- d-dimensional hypercube
  - $V = \{0,1\}^d$, $n = 2^d$
  - $E = \{\{u, v\} \colon u \text{ and } v \text{ differ in one bit}\}$

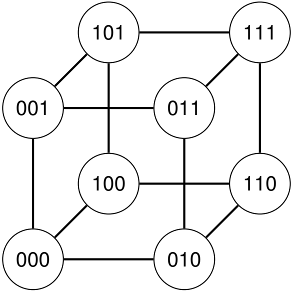


Hypercube

- d-dimensional hypercube
  - $V = \{0,1\}^d$, $n = 2^d$
  - $E = \{\{u,v\} : u \text{ and } v \text{ differ in one bit}\}$

Dimension Exchange (Balancing Circuit)

- round i: every node communicates along dimension i
- load of communicating nodes is averaged

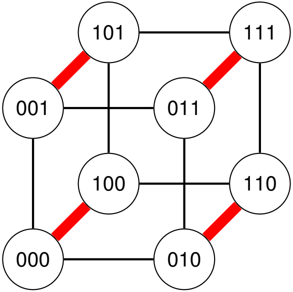


Hypercube

- d-dimensional hypercube
  - $V = \{0,1\}^d$, $n = 2^d$
  - $E = \{\{u,v\} : u$ and v differ in one bit$\}$

Dimension Exchange (Balancing Circuit)

- round i: every node communicates along dimension i
- load of communicating nodes is averaged

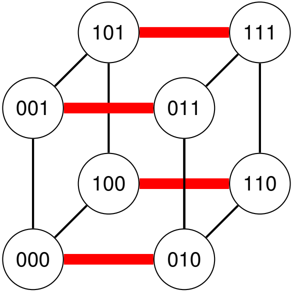


Hypercube

- d-dimensional hypercube
  - $V = \{0,1\}^d$, $n = 2^d$
  - $E = \{\{u,v\}: u \text{ and } v \text{ differ in one bit}\}$

Dimension Exchange (Balancing Circuit)

- round i: every node communicates along dimension i
- load of communicating nodes is averaged

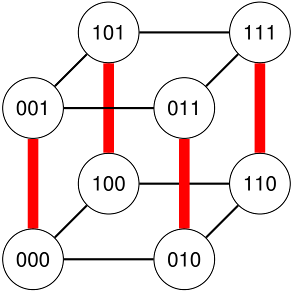


Hypercube

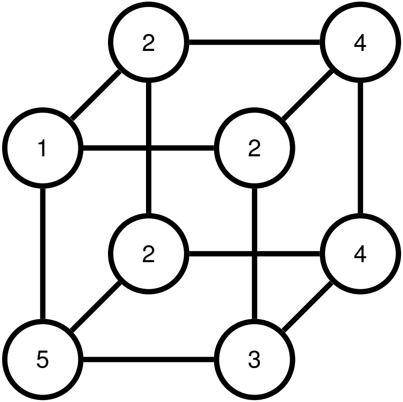
- d-dimensional hypercube
  - $V = \{0,1\}^d$, $n = 2^d$
  - $E = \{\{u,v\} \colon u$ and v differ in one bit$\}$

Dimension Exchange (Balancing Circuit)

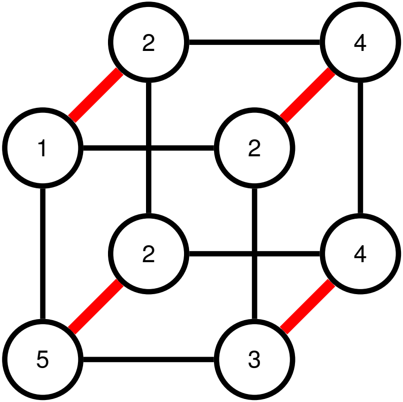
- round i: every node communicates along dimension i
- load of communicating nodes is averaged



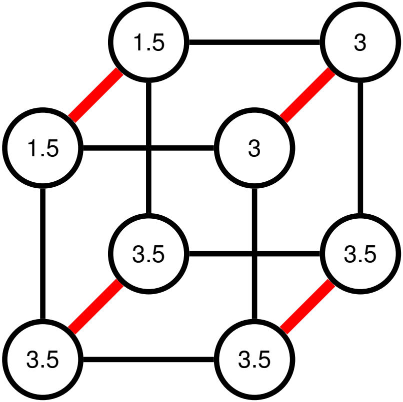
Continuous vs. Discrete Load Balancing



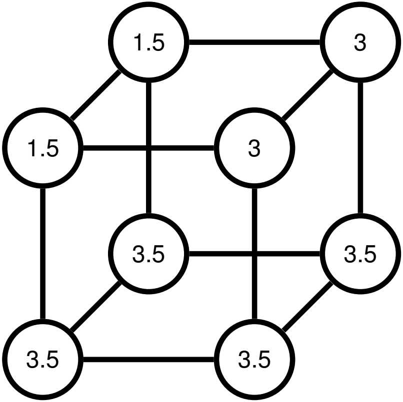
Continuous vs. Discrete Load Balancing



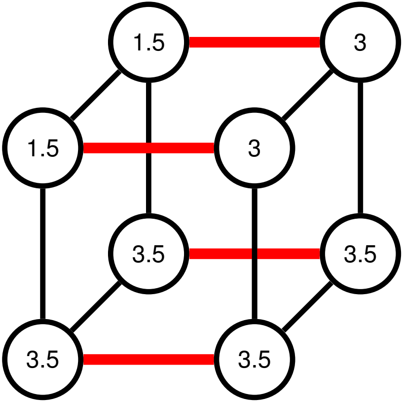
Continuous vs. Discrete Load Balancing



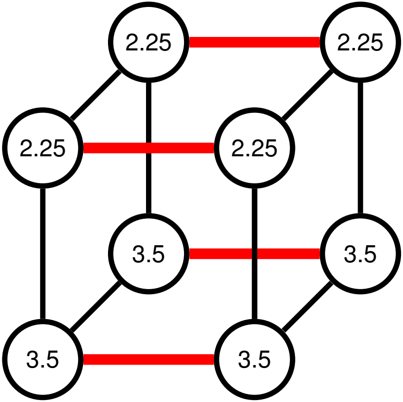
Continuous vs. Discrete Load Balancing



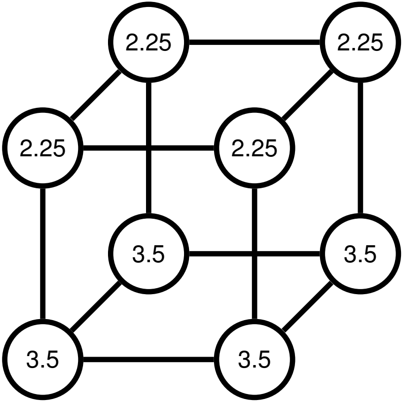
Continuous vs. Discrete Load Balancing



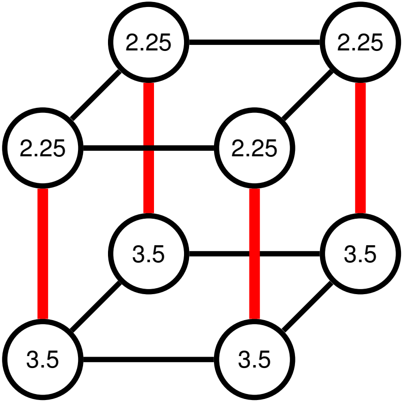
Continuous vs. Discrete Load Balancing



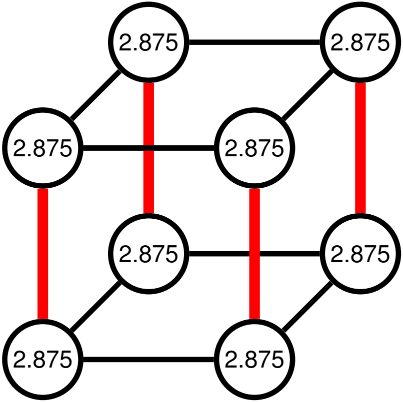
Continuous vs. Discrete Load Balancing



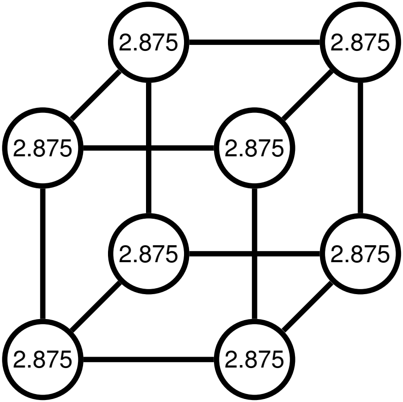
Continuous vs. Discrete Load Balancing



Continuous vs. Discrete Load Balancing

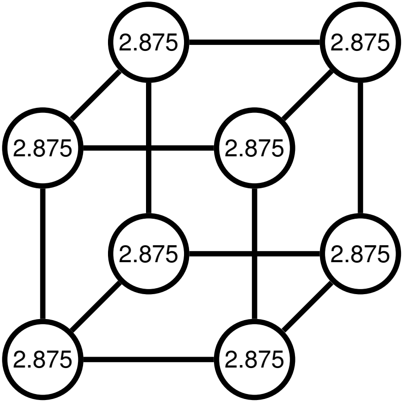


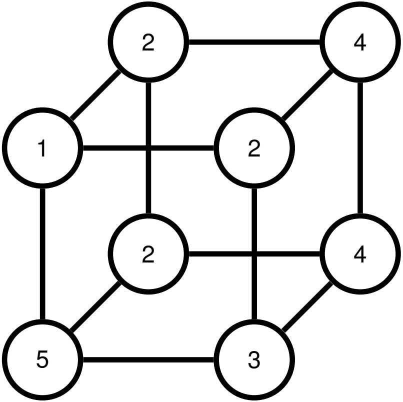
Continuous vs. Discrete Load Balancing



perfectly balanced!

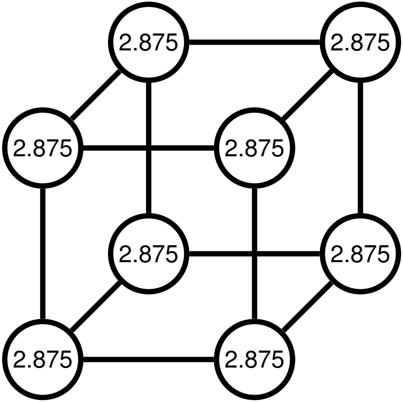
Continuous vs. Discrete Load Balancing



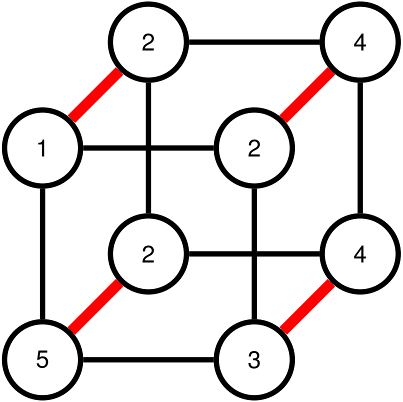


perfectly balanced!

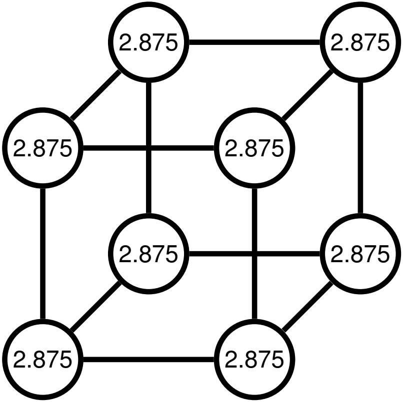
Continuous vs. Discrete Load Balancing

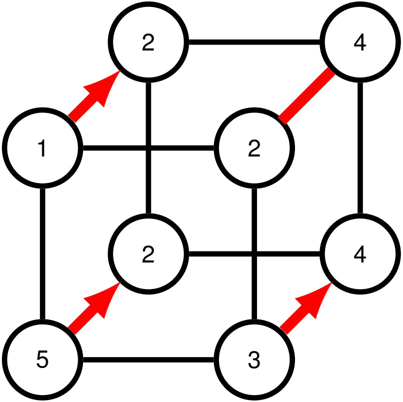


perfectly balanced!



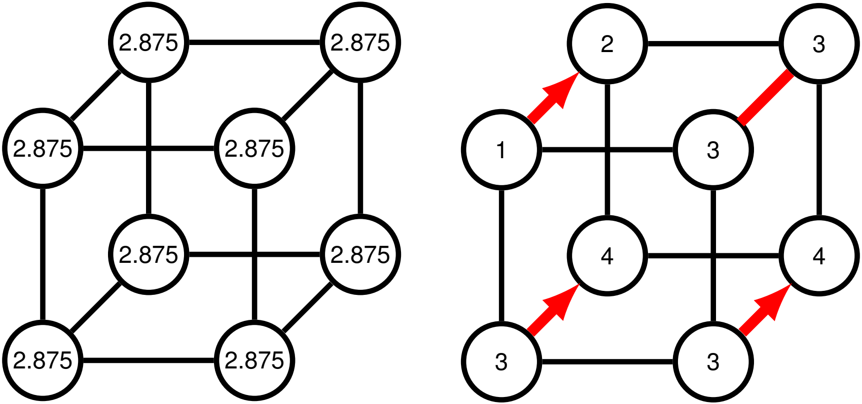
Continuous vs. Discrete Load Balancing





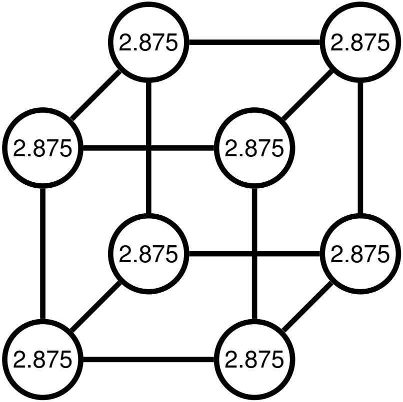
perfectly balanced!

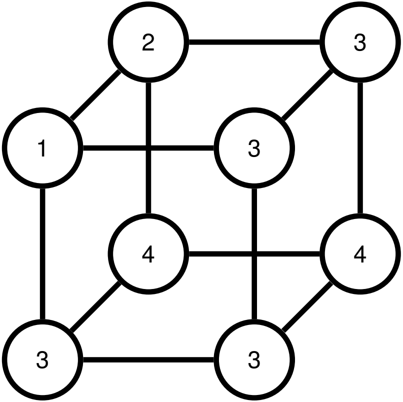
Continuous vs. Discrete Load Balancing



perfectly balanced!

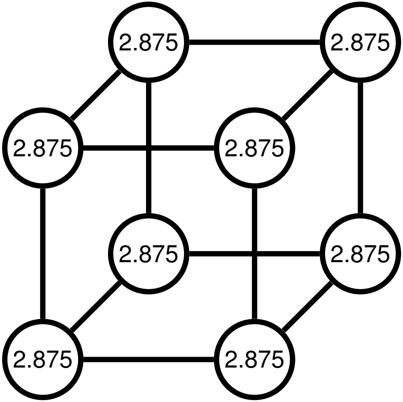
Continuous vs. Discrete Load Balancing

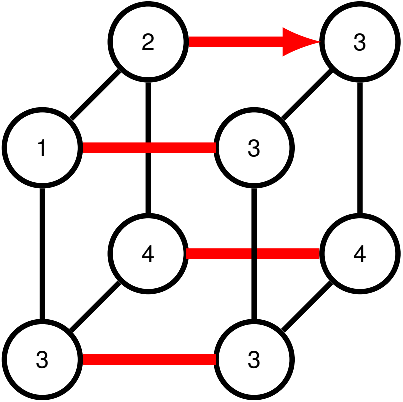




perfectly balanced!

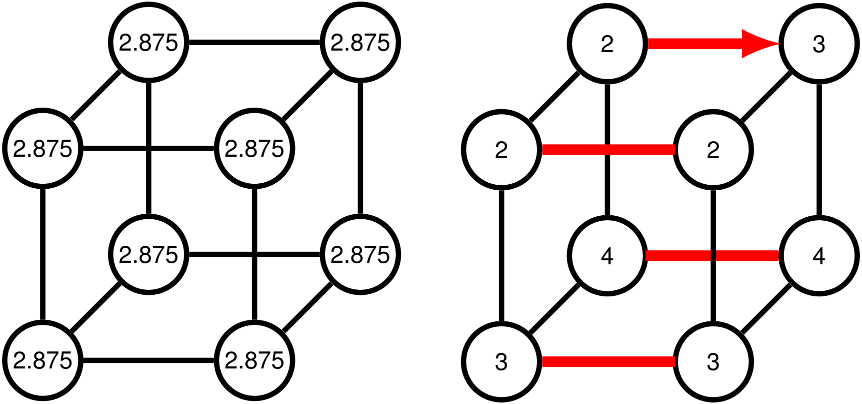
Continuous vs. Discrete Load Balancing





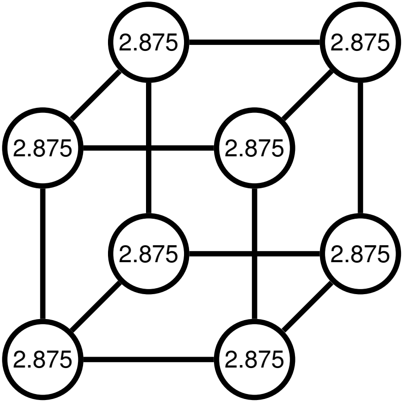
perfectly balanced!

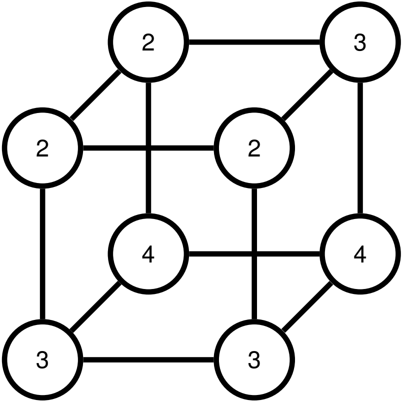
Continuous vs. Discrete Load Balancing



perfectly balanced!

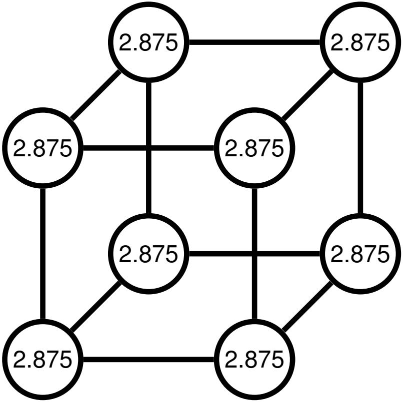
Continuous vs. Discrete Load Balancing

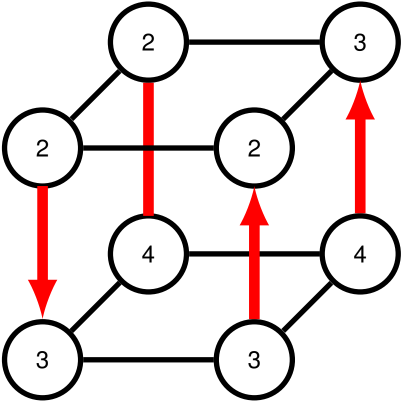




perfectly balanced!

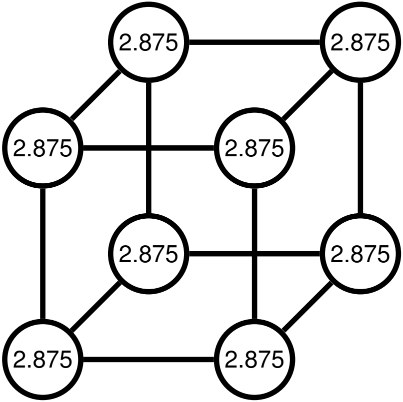
Continuous vs. Discrete Load Balancing

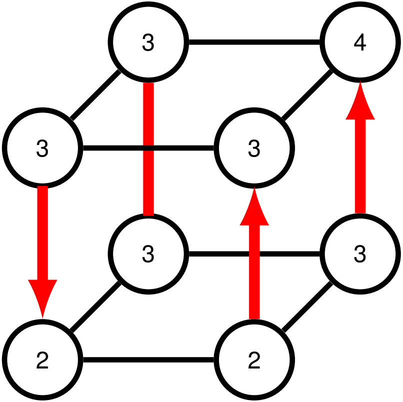




perfectly balanced!

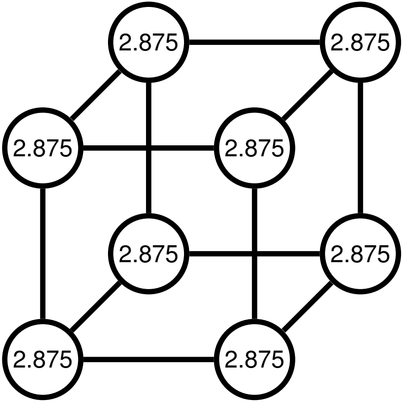
Continuous vs. Discrete Load Balancing

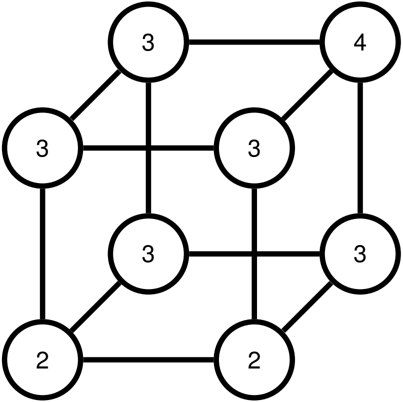




perfectly balanced!

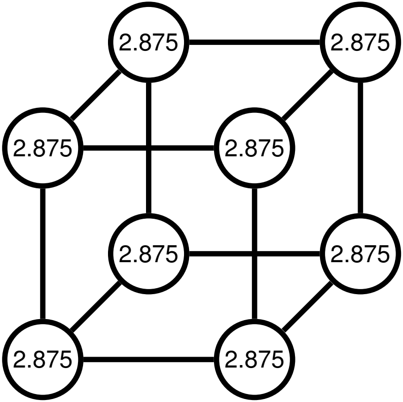
Continuous vs. Discrete Load Balancing



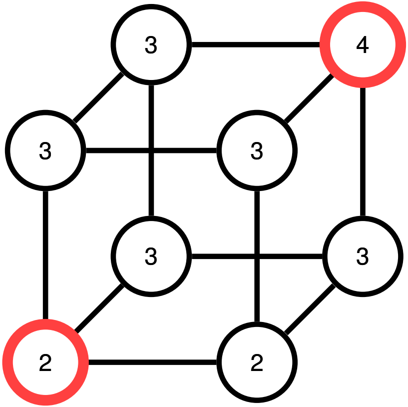


perfectly balanced!

Continuous vs. Discrete Load Balancing

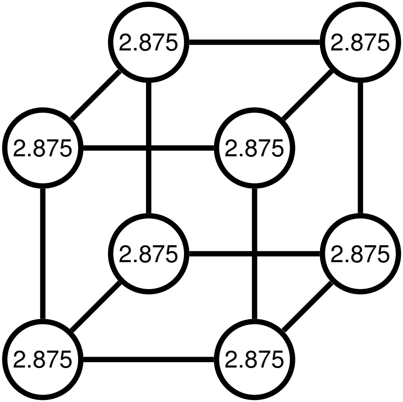


perfectly balanced!

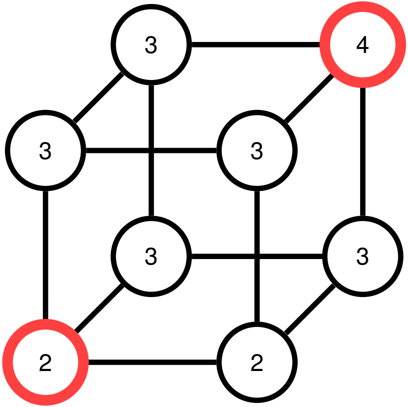


discrepancy of 2

Continuous vs. Discrete Load Balancing



perfectly balanced!



discrepancy of 2

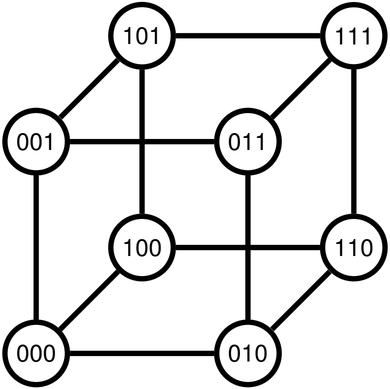
Question

How to minimize the gap between the discrete and continuous case?

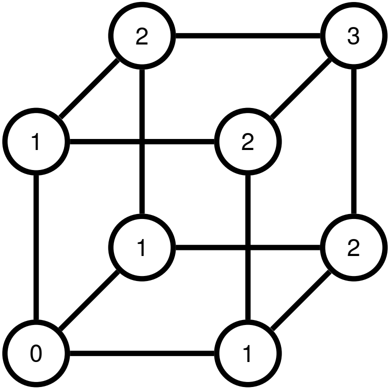
Asynchronous Execution (Smoothing Networks)

Stop

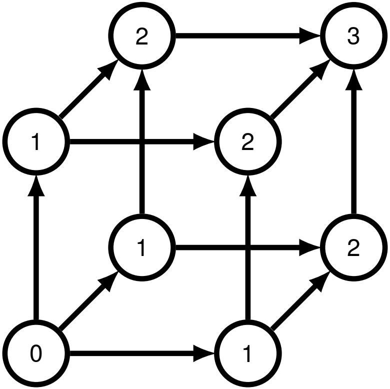
Maximum Discrepancy



Maximum Discrepancy

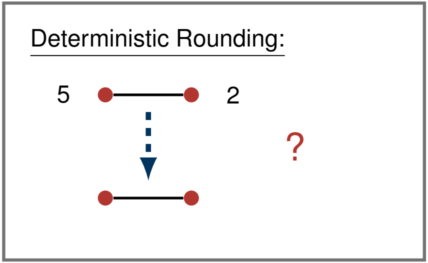


Maximum Discrepancy



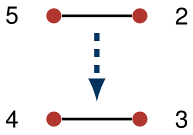
- load vector is never changed
- discrepancy remains 3 (or more generally, $d = \log_2 n$)

Deterministic vs. Randomized Rounding



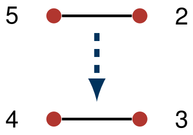
Deterministic vs. Randomized Rounding

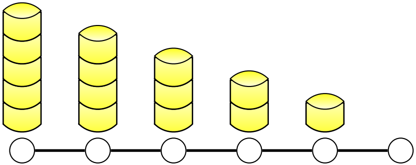
Deterministic Rounding:



Deterministic vs. Randomized Rounding

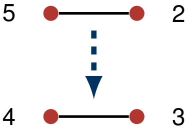
Deterministic Rounding:

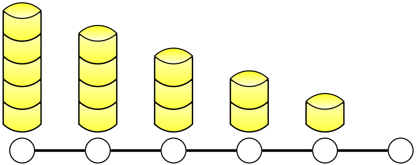




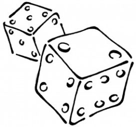
Deterministic vs. Randomized Rounding

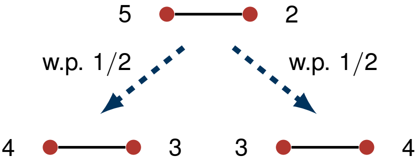
Deterministic Rounding:





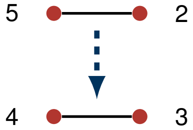
Randomized Rounding:

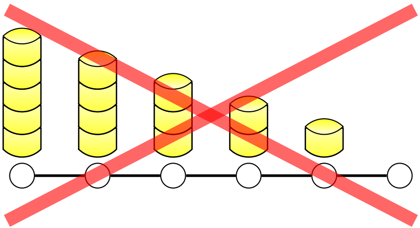




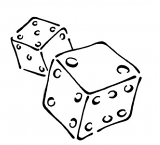
Deterministic vs. Randomized Rounding

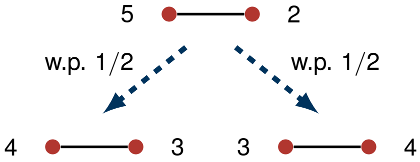
Deterministic Rounding:





Randomized Rounding:





Upper Bounds for the Hypercube

Arbitrary Rounding (Herlihy, Tirthapura, 2006)

For any initial load vector, the disc. is at most $\log_2 n$ after $\log_2 n$ rounds.

Upper Bounds for the Hypercube

Arbitrary Rounding (Herlihy, Tirthapura, 2006)

For any initial load vector, the disc. is at most $\log_2 n$ after $\log_2 n$ rounds.

Randomized Rounding (Herlihy, Tirthapura, 2006)

For any initial load vector, the discrepancy is at most $\mathcal{O}(\sqrt{\log n})$.

Upper Bounds for the Hypercube

Arbitrary Rounding (Herlihy, Tirthapura, 2006)

For any initial load vector, the disc. is at most $\log_2 n$ after $\log_2 n$ rounds.

Randomized Rounding (Herlihy, Tirthapura, 2006)

For any initial load vector, the discrepancy is at most $\mathcal{O}(\sqrt{\log n})$.

Randomized Rounding (Mavronicolas, S., 2010)

For any initial load vector, the discrepancy is at most $\log_2 \log_2 n + 4$.

Upper Bounds for the Hypercube

Arbitrary Rounding (Herlihy, Tirthapura, 2006)

For any initial load vector, the disc. is at most $\log_2 n$ after $\log_2 n$ rounds.

Randomized Rounding (Herlihy, Tirthapura, 2006)

For any initial load vector, the discrepancy is at most $\mathcal{O}(\sqrt{\log n})$.

Randomized Rounding (Mavronicolas, S., 2010)

For any initial load vector, the discrepancy is at most $\log_2 \log_2 n + 4$.

- initial load distribution completely arbitrary (but chosen oblivious to the randomized rounding)
- results hold with probability at least $1 - n^{-1}$

Step 1: Expressing the Rounding Error

$$x_u^t = \frac{x_u^{t-1} + x_v^{t-1}}{2}$$

$$x_v^t = \frac{x_u^{t-1} + x_v^{t-1}}{2}$$

Step 1: Expressing the Rounding Error

$$x_u^t = \frac{x_u^{t-1} + x_v^{t-1}}{2} + e_{u,v}^t$$

$$x_v^t = \frac{x_u^{t-1} + x_v^{t-1}}{2} - e_{u,v}^t,$$

with $e_{u,v}^t$ being the rounding error,

Step 1: Expressing the Rounding Error

$$x_u^t = \frac{x_u^{t-1} + x_v^{t-1}}{2} + e_{u,v}^t$$
$$x_v^t = \frac{x_u^{t-1} + x_v^{t-1}}{2} - e_{u,v}^t,$$

with $e_{u,v}^t$ being the rounding error,

$$e_{u,v}^t = \mathrm{Odd}(x_u^{t-1} + x_v^{t-1}) \cdot \Phi_{u,v}^t,$$

where the $\Phi_{u,v}^t \in \{-1/2, +1/2\}$ is the (random) orientation.

Step 1: Expressing the Rounding Error

$$x_u^t = \frac{x_u^{t-1} + x_v^{t-1}}{2} + e_{u,v}^t$$

$$x_v^t = \frac{x_u^{t-1} + x_v^{t-1}}{2} - e_{u,v}^t,$$

with $e_{u,v}^t$ being the rounding error,

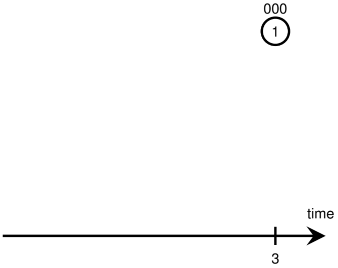
$$e_{u,v}^t = \mathrm{Odd}(x_u^{t-1} + x_v^{t-1}) \cdot \Phi_{u,v}^t,$$

where the $\Phi_{u,v}^t \in \{-1/2, +1/2\}$ is the (random) orientation.

$$e_{u,v}^t \in \{-1/2, 0, 1/2\} \text{ and } \mathbf{E}\left[\, e_{u,v}^t \,\right] = 0.$$

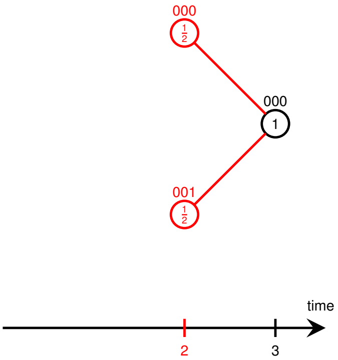
Step 2: Solving and Analyzing the Recursion

x_{000}^{3}



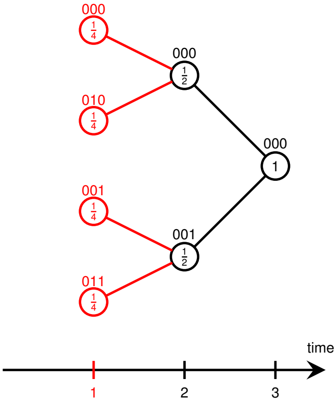
Step 2: Solving and Analyzing the Recursion

$$x^3_{000} = \frac{1}{2}x^2_{000} + \frac{1}{2}x^2_{001} + e^3_{000}$$



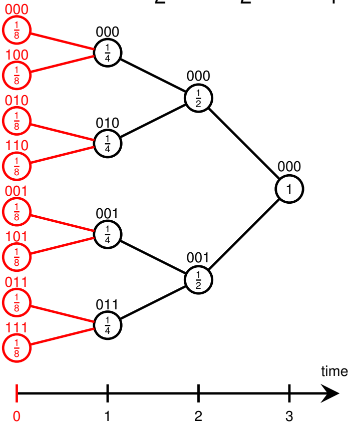
Step 2: Solving and Analyzing the Recursion

$$
\begin{aligned}
x_{000}^{3} &= \frac{1}{2}x_{000}^{2} + \frac{1}{2}x_{001}^{2} + e_{000}^{3} \\
&= \frac{1}{4}x_{000}^{1} + \frac{1}{4}x_{001}^{1} + \frac{1}{4}x_{001}^{1} + \frac{1}{4}x_{011}^{1} + e_{000}^{3} + \frac{1}{2}e_{000}^{2} + \frac{1}{2}e_{001}^{2}
\end{aligned}
$$



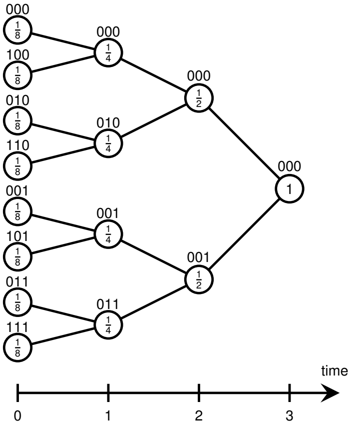
Step 2: Solving and Analyzing the Recursion

$$
\begin{aligned}
x^3_{000} &= \frac{1}{4}x^1_{000} + \frac{1}{4}x^1_{001} + \frac{1}{4}x^1_{001} + \frac{1}{4}x^1_{011} + e^3_{000} + \frac{1}{2}e^2_{000} + \frac{1}{2}e^2_{001} \\
&= \frac{1}{8}x^0_{000} + \frac{1}{8}x^0_{100} + \frac{1}{8}x^0_{010} + \frac{1}{8}x^0_{110} + \frac{1}{8}x^0_{001} + \frac{1}{8}x^0_{101} + \frac{1}{8}x^0_{011} + \frac{1}{8}x^0_{111} \\
&\quad + e^3_{000} + \frac{1}{2}e^2_{000} + \frac{1}{2}e^2_{001} + \frac{1}{4}e^1_{000} + \frac{1}{4}e^1_{010} + \frac{1}{4}e^1_{001} + \frac{1}{4}e^1_{011}
\end{aligned}
$$



Step 2: Solving and Analyzing the Recursion

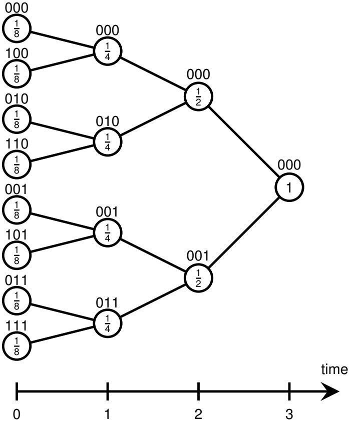
$$x^3_{000} = \frac{1}{8}x^0_{000} + \frac{1}{8}x^0_{100} + \frac{1}{8}x^0_{010} + \frac{1}{8}x^0_{110} + \frac{1}{8}x^0_{001} + \frac{1}{8}x^0_{101} + \frac{1}{8}x^0_{011} + \frac{1}{8}x^0_{111}$$
$$+ e^3_{000} + \frac{1}{2}e^2_{000} + \frac{1}{2}e^2_{001} + \frac{1}{4}e^1_{000} + \frac{1}{4}e^1_{010} + \frac{1}{4}e^1_{001} + \frac{1}{4}e^1_{011}$$



- continuous part and discrete part

Step 2: Solving and Analyzing the Recursion

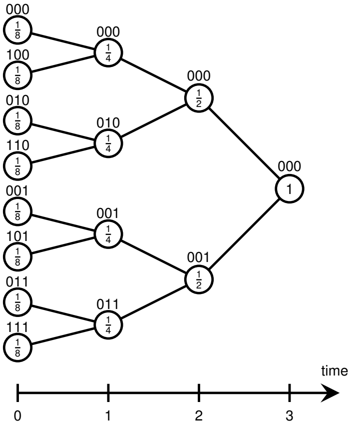
$$x^3_{000} = \frac{1}{8}x^0_{000} + \frac{1}{8}x^0_{100} + \frac{1}{8}x^0_{010} + \frac{1}{8}x^0_{110} + \frac{1}{8}x^0_{001} + \frac{1}{8}x^0_{101} + \frac{1}{8}x^0_{011} + \frac{1}{8}x^0_{111}$$
$$+ e^3_{000} + \frac{1}{2}e^2_{000} + \frac{1}{2}e^2_{001} + \frac{1}{4}e^1_{000} + \frac{1}{4}e^1_{010} + \frac{1}{4}e^1_{001} + \frac{1}{4}e^1_{011}$$



- continuous part and discrete part
- continuous part equals the average load

Step 2: Solving and Analyzing the Recursion

$$x_{000}^3 = \frac{1}{8}x_{000}^0 + \frac{1}{8}x_{100}^0 + \frac{1}{8}x_{010}^0 + \frac{1}{8}x_{110}^0 + \frac{1}{8}x_{001}^0 + \frac{1}{8}x_{101}^0 + \frac{1}{8}x_{011}^0 + \frac{1}{8}x_{111}^0$$
$$+e_{000}^3 + \frac{1}{2}e_{000}^2 + \frac{1}{2}e_{001}^2 + \frac{1}{4}e_{000}^1 + \frac{1}{4}e_{010}^1 + \frac{1}{4}e_{001}^1 + \frac{1}{4}e_{011}^1$$

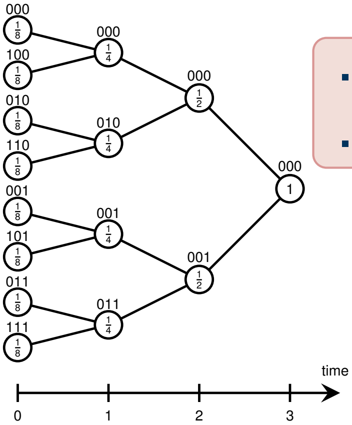


- continuous part and discrete part
- continuous part equals the average load

⇒ loads are divisible, then perfectly balanced

Step 2: Solving and Analyzing the Recursion

$$x^3_{000} = \frac{1}{8}x^0_{000} + \frac{1}{8}x^0_{100} + \frac{1}{8}x^0_{010} + \frac{1}{8}x^0_{110} + \frac{1}{8}x^0_{001} + \frac{1}{8}x^0_{101} + \frac{1}{8}x^0_{011} + \frac{1}{8}x^0_{111}$$
$$+ e^3_{000} + \frac{1}{2}e^2_{000} + \frac{1}{2}e^2_{001} + \frac{1}{4}e^1_{000} + \frac{1}{4}e^1_{010} + \frac{1}{4}e^1_{001} + \frac{1}{4}e^1_{011}$$



- blue part essentially sum of independent random variables
- ranges decrease exponentially!

- continuous part and discrete part
- continuous part equals the average load

$\Rightarrow$ loads are divisible, then perfectly balanced

Lower Bound for the Hypercube

Upper Bound (Mavronicolas, S., 2010)

For any initial load vector, the discrepancy is at most $\log_2 \log_2 n + 4$.

Lower Bound for the Hypercube

Upper Bound (Mavronicolas, S., 2010)

For any initial load vector, the discrepancy is at most $\log_2 \log_2 n + 4$.

Lower Bound (Mavronicolas, S., 2010)

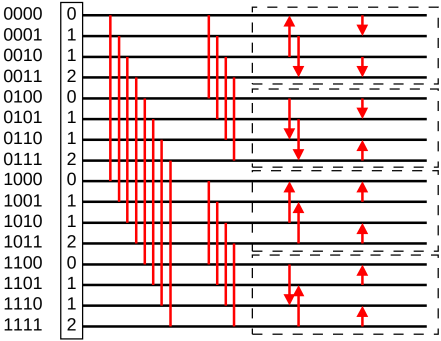
There are initial load vectors so that the discrepancy is at least $\log_2 \log_2 n - 2$ w.p. $1 - n^{-1}$.

Proof Idea of the Lower Bound

Initial load at i is the number of ones in the $\log_2 \log_2 n + 1$ lowest bits.

Proof Idea of the Lower Bound

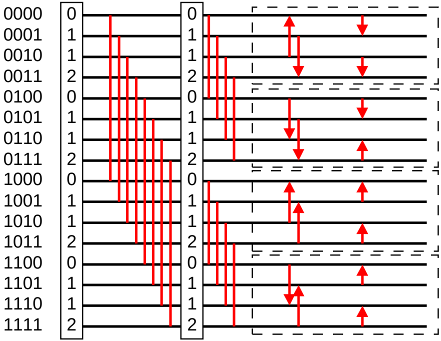
Initial load at i is the number of ones in the $\log_2 \log_2 n + 1$ lowest bits.



Proof Idea of the Lower Bound

Initial load at i is the number of ones in the $\log_2 \log_2 n + 1$ lowest bits.

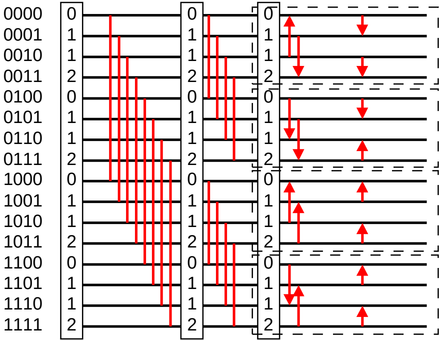
- no balancing in the first $\log_2 n - \log_2 \log_2 n + 1$ rounds



Proof Idea of the Lower Bound

Initial load at i is the number of ones in the $\log_2 \log_2 n + 1$ lowest bits.

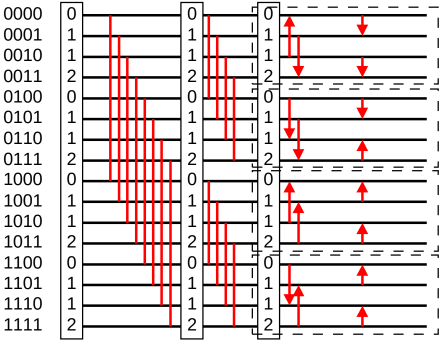
- no balancing in the first $\log_2 n - \log_2 \log_2 n + 1$ rounds



Proof Idea of the Lower Bound

Initial load at i is the number of ones in the $\log_2 \log_2 n + 1$ lowest bits.

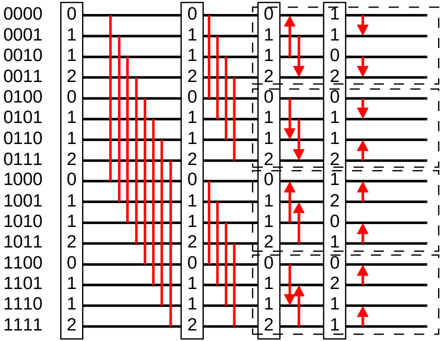
- no balancing in the first $\log_2 n - \log_2 \log_2 n + 1$ rounds
- last $\log_2 \log_2 n - 1$ rounds: $\approx \frac{n}{\log_2 n}$ parallel subcubes



Proof Idea of the Lower Bound

Initial load at i is the number of ones in the $\log_2 \log_2 n + 1$ lowest bits.

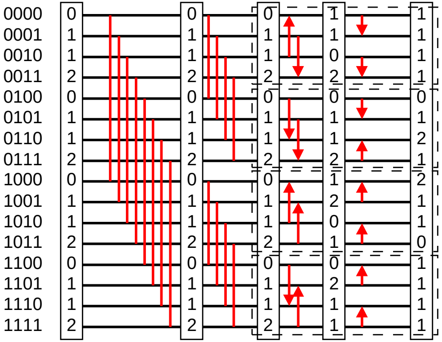
- no balancing in the first $\log_2 n - \log_2 \log_2 n + 1$ rounds
- last $\log_2 \log_2 n - 1$ rounds: $\approx \frac{n}{\log_2 n}$ parallel subcubes
- each w.p. $\approx \frac{1}{\sqrt{n}}$ discrepancy at least $\log_2 \log_2 n - 2$



Proof Idea of the Lower Bound

Initial load at i is the number of ones in the $\log_2 \log_2 n + 1$ lowest bits.

- no balancing in the first $\log_2 n - \log_2 \log_2 n + 1$ rounds
- last $\log_2 \log_2 n - 1$ rounds: $\approx \frac{n}{\log_2 n}$ parallel subcubes
- each w.p. $\approx \frac{1}{\sqrt{n}}$ discrepancy at least $\log_2 \log_2 n - 2$



Improving the Lower Bound

Upper Bound (Mavronicolas, S., 2010)

For any initial load vector, the discrepancy is at most $\log_2 \log_2 n + 4$.

Lower Bound (Mavronicolas, S., 2010)

There are initial load vectors so that the discrepancy is at least $\log_2 \log_2 n - 2$ w.p. $1 - n^{-1}$.

Improving the Lower Bound

Upper Bound (Mavronicolas, S., 2010)

For any initial load vector, the discrepancy is at most $\log_2 \log_2 n + 4$.

Lower Bound (Mavronicolas, S., 2010)

There are initial load vectors so that the discrepancy is at least $\log_2 \log_2 n - 2$ w.p. $1 - n^{-1}$.

How can we reduce the discrepancy further?

Improving the Lower Bound

Upper Bound (Mavronicolas, S., 2010)

For any initial load vector, the discrepancy is at most $\log_2 \log_2 n + 4$.

Lower Bound (Mavronicolas, S., 2010)

There are initial load vectors so that the discrepancy is at least $\log_2 \log_2 n - 2$ w.p. $1 - n^{-1}$.

How can we reduce the discrepancy further?

Average Case Input (Friedrich, S., Vilenchik, 2011)

Even if the initial load at a node is chosen i.u.r. in $\{0, 1, \ldots, n-1\}$, then the discrepancy is at least $\frac{1}{2} \cdot \log_2 \log_2 n - 2$ w.p. $1 - n^{-1}$.

New Attempt

Old Protocol (Mavronicolas, S., 2010)

- rounds $0, 1, \ldots, \log_2 n - 1$
- in round i communicate along dimension i
- discrepancy $\leqslant \log_2 \log_2 n + 4$

New Attempt

Old Protocol (Mavronicolas, S., 2010)

- rounds $0, 1, \dots, \log_2 n - 1$
- in round i communicate along dimension i
- discrepancy $\leqslant \log_2 \log_2 n + 4$

New Protocol (Mavronicolas, S., 2010)

- rounds $0, 1, \dots, 3 \log_2 n - 1$
- in round i communicate along dimension $i \mod \log n$

New Attempt

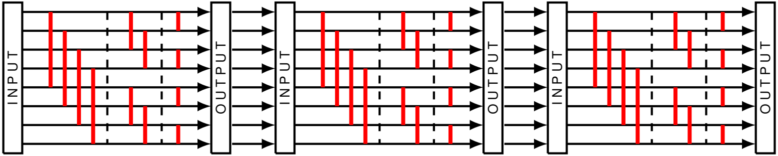
Old Protocol (Mavronicolas, S., 2010)

- rounds $0, 1, \ldots, \log_2 n - 1$
- in round i communicate along dimension i
- discrepancy $\leqslant \log_2 \log_2 n + 4$

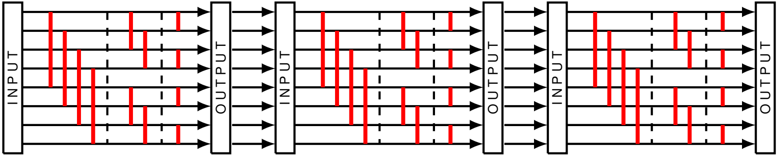
New Protocol (Mavronicolas, S., 2010)

- rounds $0, 1, \ldots, 3 \log_2 n - 1$
- in round i communicate along dimension $i \mod \log n$
- discrepancy $\leqslant 2$

Analysis

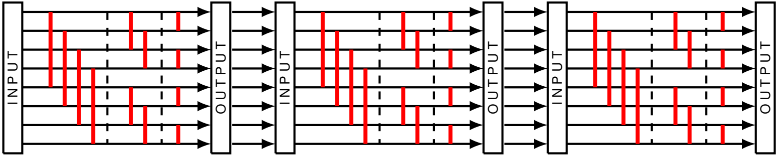


Analysis



- after the first $\log_2 n$ rounds, discrepancy is at most $\log\log_2 n + 4$

Analysis



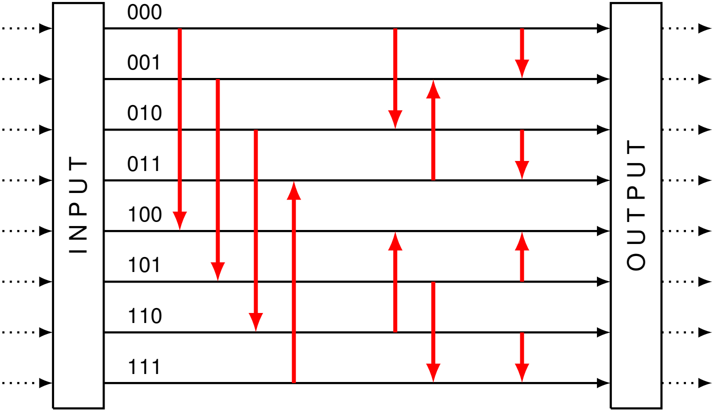
- after the first $\log_2 n$ rounds, discrepancy is at most $\log\log_2 n + 4$
- analysis consists of $\log_2 \log_2 n + 2$ phases: each phase reduces D by 1
- focus on the last phase (D drops from 3 to 2)

Maximum-Paths

- idea: keep track of the maxima in the load vector

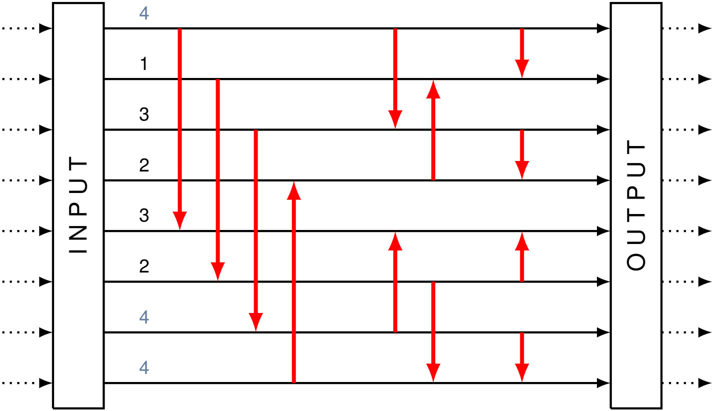
Maximum-Paths

- idea: keep track of the maxima in the load vector



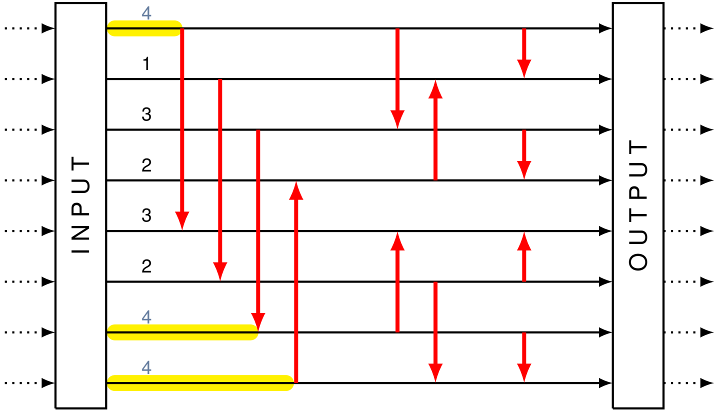
Maximum-Paths

- idea: keep track of the maxima in the load vector



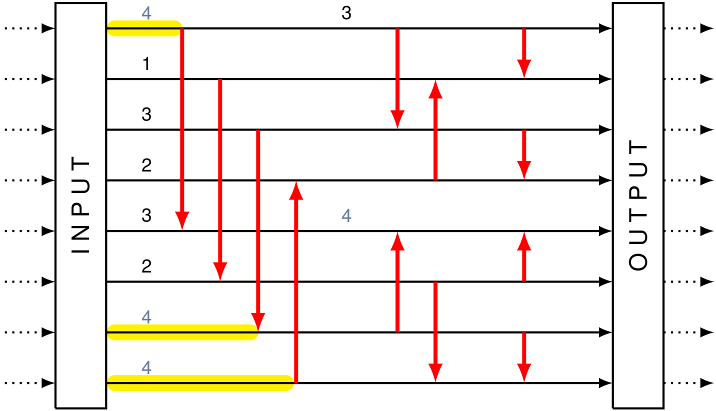
Maximum-Paths

- idea: keep track of the maxima in the load vector



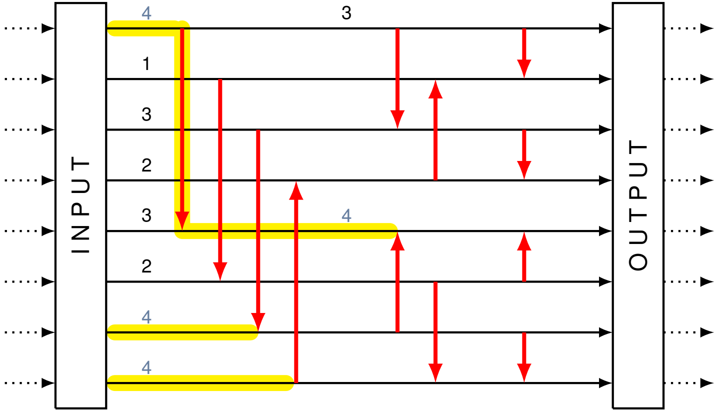
Maximum-Paths

- idea: keep track of the maxima in the load vector



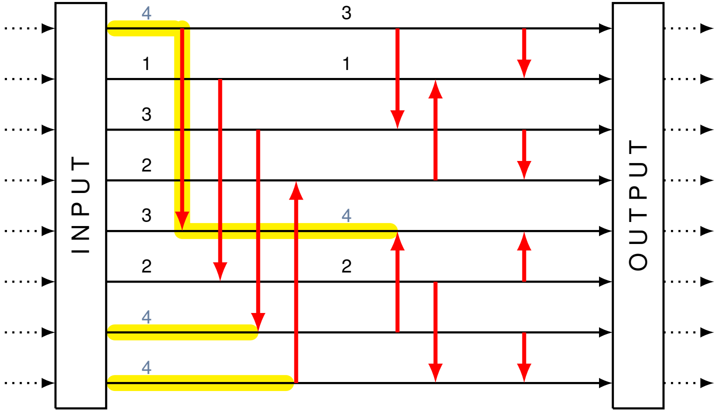
Maximum-Paths

- idea: keep track of the maxima in the load vector



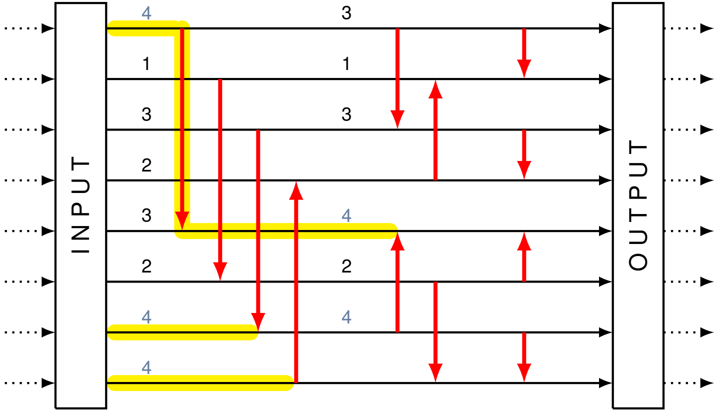
Maximum-Paths

- idea: keep track of the maxima in the load vector



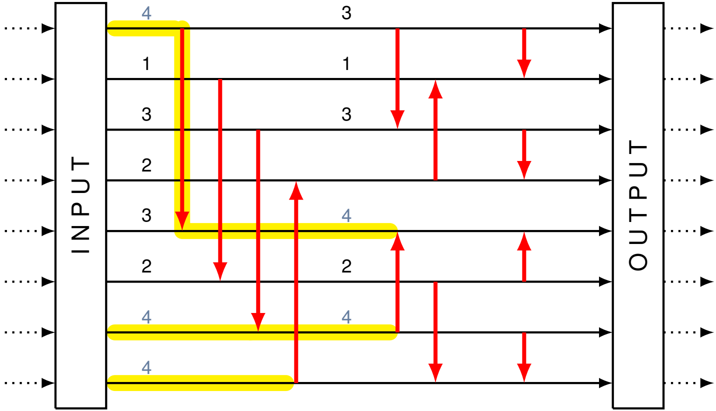
Maximum-Paths

- idea: keep track of the maxima in the load vector



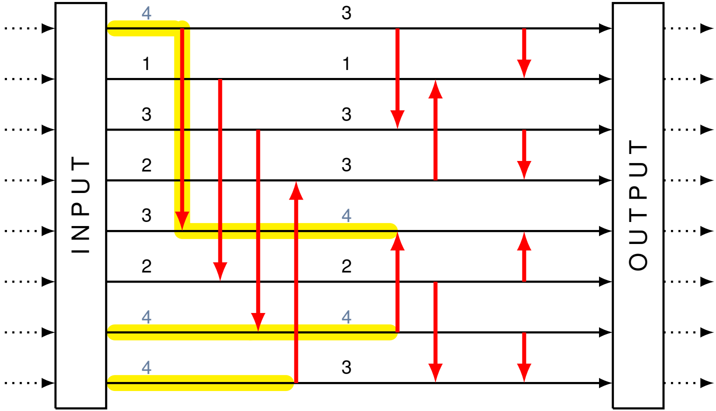
Maximum-Paths

- idea: keep track of the maxima in the load vector



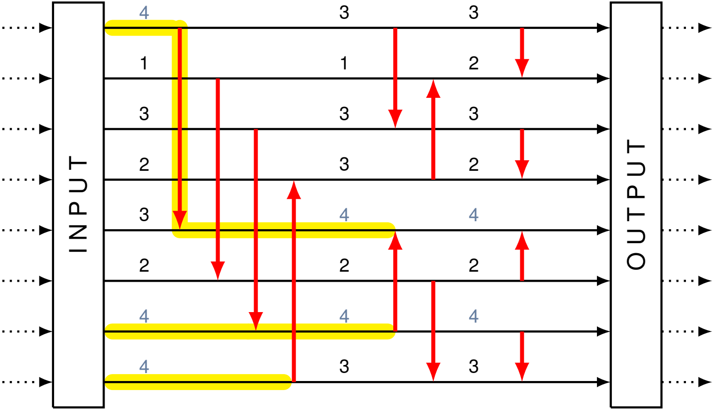
Maximum-Paths

- idea: keep track of the maxima in the load vector



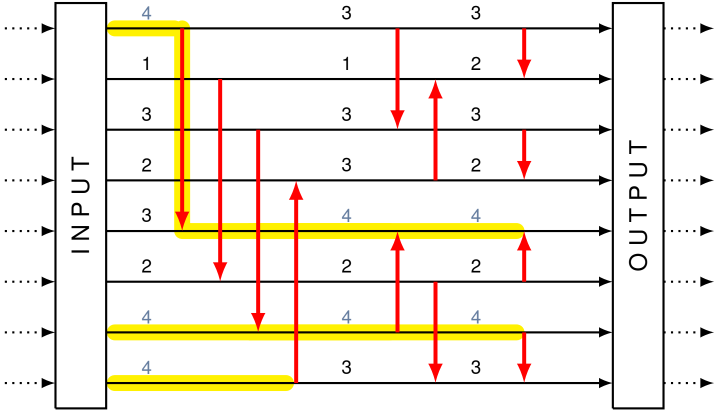
Maximum-Paths

- idea: keep track of the maxima in the load vector



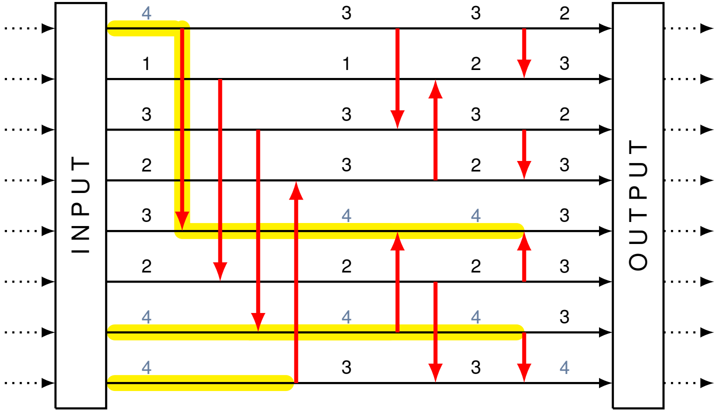
Maximum-Paths

- idea: keep track of the maxima in the load vector



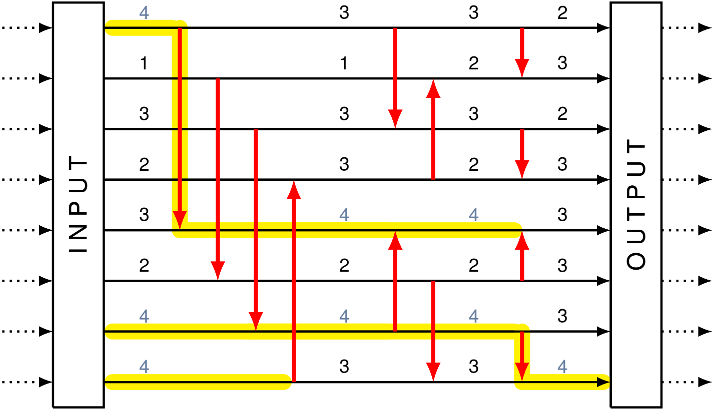
Maximum-Paths

- idea: keep track of the maxima in the load vector

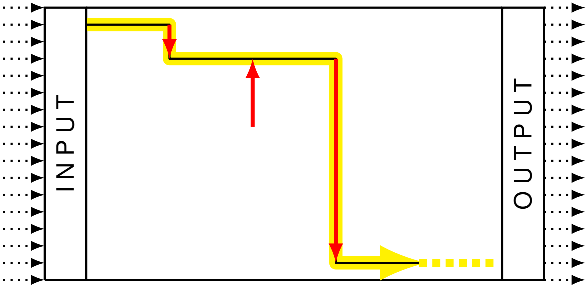


Maximum-Paths

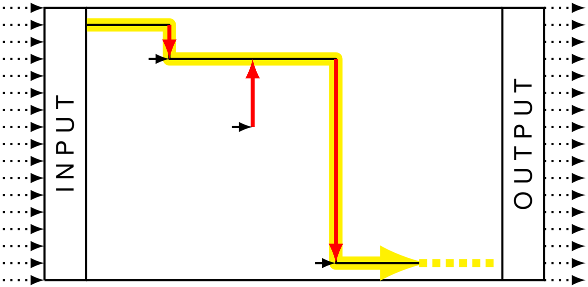
- idea: keep track of the maxima in the load vector



Reducing Discrepancy to 2

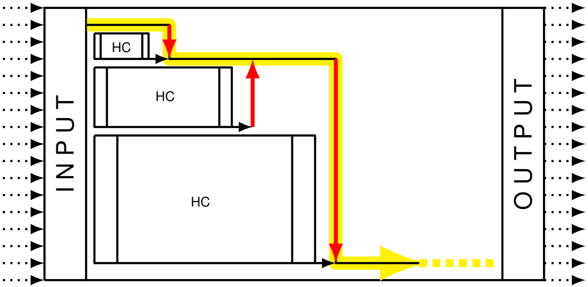


Reducing Discrepancy to 2



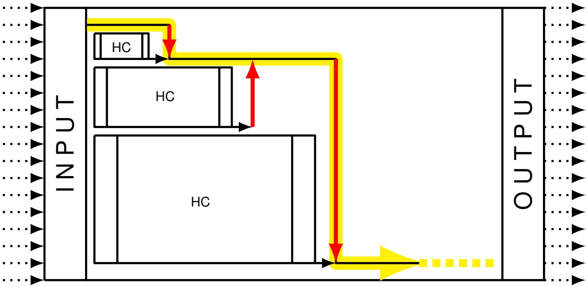
- maximum path survives $\Leftrightarrow$ all inputs $\geqslant$ max -1

Reducing Discrepancy to 2



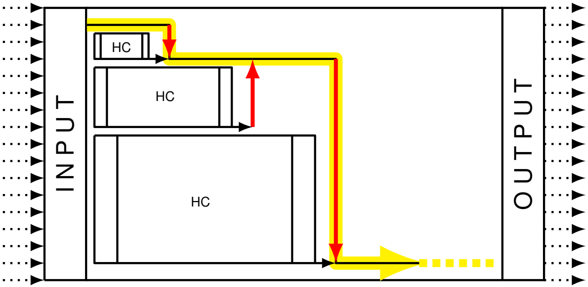
- maximum path survives $\Leftrightarrow$ all inputs $\geqslant$ max -1
- these inputs are outputs of disjoint subcubes,
  and: the averages of inputs to the subcubes are "good"

Reducing Discrepancy to 2



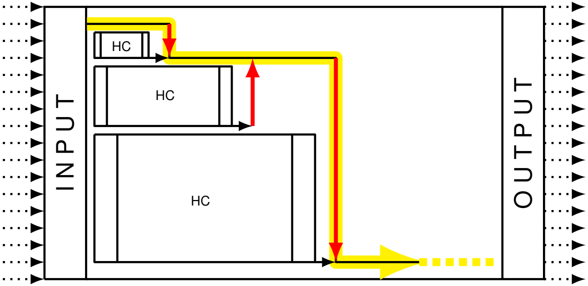
- maximum path survives $\Leftrightarrow$ all inputs $\geqslant$ max -1
- these inputs are outputs of disjoint subcubes,
  and: the averages of inputs to the subcubes are "good"
- $\mathbf{Pr}\,[\,\text{output} \geqslant \max -1\,] \leqslant \frac{1}{2}$

Reducing Discrepancy to 2



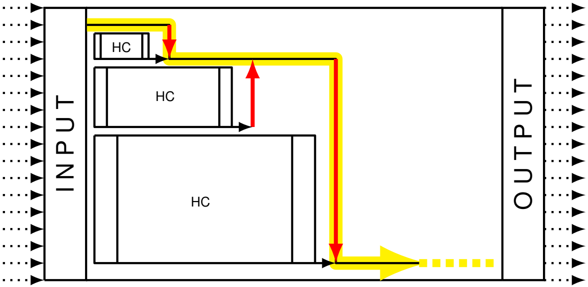
- maximum path survives $\Leftrightarrow$ all inputs $\geqslant$ max -1
- these inputs are outputs of disjoint subcubes,
  and: the averages of inputs to the subcubes are "good"
- $\mathbf{Pr}\,[\,\text{output} \geqslant \text{max} - 1\,] \leqslant \frac{1}{2}$
- $\mathbf{Pr}\,[\,\text{path survives } \log n \text{ rounds}\,] \leqslant \left(\frac{1}{2}\right)^{\log n}$

Reducing Discrepancy to 2



- maximum path survives $\Leftrightarrow$ all inputs $\geqslant$ max -1
- these inputs are outputs of disjoint subcubes,
  and: the averages of inputs to the subcubes are "good"
- $\mathbf{Pr}\,[\,\text{output} \geqslant \max - 1\,] \leqslant \frac{1}{2}$
- $\mathbf{Pr}\,[\,\text{path survives } \log n \text{ rounds}\,] \leqslant \left(\frac{1}{2}\right)^{\log n}$
- $\mathbf{Pr}\,[\,\text{path survives } 2\log n \text{ rounds}\,] \leqslant \left(\frac{1}{2}\right)^{2\log n}$

Reducing Discrepancy to 2



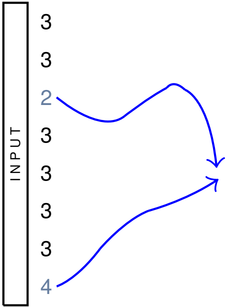
- maximum path survives $\Leftrightarrow$ all inputs $\geqslant$ max -1
- these inputs are outputs of disjoint subcubes,
  and: the averages of inputs to the subcubes are “good”
- $\mathbf{Pr}\,[\,\text{output} \geqslant \max - 1\,] \leqslant \frac{1}{2}$
- $\mathbf{Pr}\,[\,\text{path survives } \log n \text{ rounds}\,] \leqslant \left(\frac{1}{2}\right)^{\log n}$
- $\mathbf{Pr}\,[\,\text{path survives } 2\log n \text{ rounds}\,] \leqslant \left(\frac{1}{2}\right)^{2\log n}$

$\Rightarrow$ with prob. $1 - n^{-1}$, discrepancy is 2 $\square$

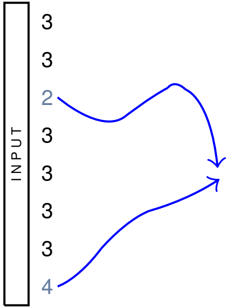
Discrepancy of 1?

INPUT	3
	3
	2
	3
	3
	3
	3
	4

Discrepancy of 1?

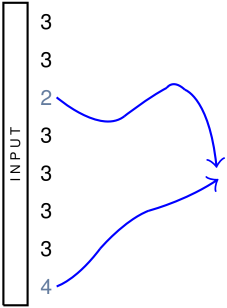


Discrepancy of 1?



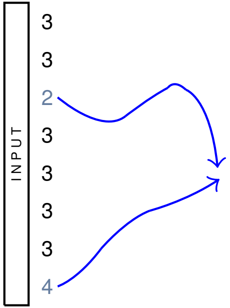
- "2" and "4" do independent random walks

Discrepancy of 1?



- "2" and "4" do independent random walks
- for proper start vertices, meeting time is $n - 1$
  $\Rightarrow$ achieving discrepancy of 1 needs $n - 1$ steps

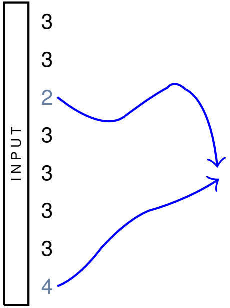
Discrepancy of 1?



- "2" and "4" do independent random walks
- for proper start vertices, meeting time is $n-1$
  $\Rightarrow$ achieving discrepancy of 1 needs $n-1$ steps
- Formally,

$$\mathbf{Pr}\left[\,\text{discrepancy is 1 after } t \text{ rounds}\,\right] \leqslant \frac{t}{n-1}.$$

Discrepancy of 1?



- "2" and "4" do independent random walks
- for proper start vertices, meeting time is $n-1$
  $\Rightarrow$ achieving discrepancy of 1 needs $n-1$ steps
- Formally,

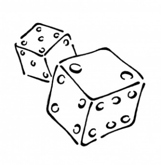
$$\mathbf{Pr}\left[\,\text{discrepancy is 1 after } t \text{ rounds}\,\right] \leqslant \frac{t}{n-1}.$$

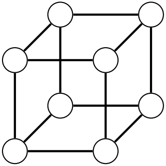
(holds for any network and any matchings)

Hypercube Summary

Randomized Rounding

- $D = \log_2 \log_2 n \pm \Theta(1)$ after $\log_2 n$ rounds
- $D = 3$ after $\log_2 n + o(\log n)$ rounds
- $D = 2$ after $3 \log_2 n$ rounds
- $D = 1$ not possible in $o(n)$ rounds



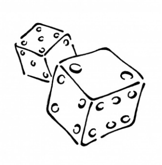


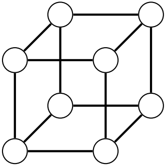
Hypercube Summary

Randomized Rounding

- $D = \log_2 \log_2 n \pm \Theta(1)$ after $\log_2 n$ rounds
- $D = 3$ after $\log_2 n + o(\log n)$ rounds
- $D = 2$ after $3 \log_2 n$ rounds
- $D = 1$ not possible in $o(n)$ rounds

$\Rightarrow$ since $\log_2 n$ rounds are necessary, hypercube is very close to the optimal network



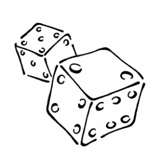


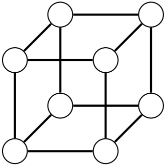
Hypercube Summary

Randomized Rounding

- $D = \log_2 \log_2 n \pm \Theta(1)$ after $\log_2 n$ rounds
- $D = 3$ after $\log_2 n + o(\log n)$ rounds
- $D = 2$ after $3 \log_2 n$ rounds
- $D = 1$ not possible in $o(n)$ rounds

$\Rightarrow$ since $\log_2 n$ rounds are necessary, hypercube is very close to the optimal network





What about general graphs?

Outline

Introduction

Load Balancing on Hypercubes

Load Balancing on Arbitrary Graphs

Conclusions

Load Balancing: Results

- $1-\lambda$: spectral expansion of the n-vertex regular graph

Load Balancing: Results

- $1-\lambda$: spectral expansion of the n-vertex regular graph
- K: discrepancy of the initial load vector

Load Balancing: Results

- $1 - \lambda$: spectral expansion of the n-vertex regular graph
- K: discrepancy of the initial load vector
- all results hold with probability $1 - o(1)$ as $n \to \infty$

Load Balancing: Results

- $1 - \lambda$: spectral expansion of the n-vertex regular graph
- K: discrepancy of the initial load vector
- all results hold with probability $1 - o(1)$ as $n \to \infty$

Deterministic Rounding (Rabani, Sinclair, Wanka, 1998)

For any graph, the discrepancy is $\mathcal{O}\left(\frac{\log n}{1-\lambda}\right)$ after $\mathcal{O}\left(\frac{\log(Kn)}{1-\lambda}\right)$ rounds.

Load Balancing: Results

- $1-\lambda$: spectral expansion of the n-vertex regular graph
- K: discrepancy of the initial load vector
- all results hold w

In the continuous case, the time to reach discrepancy $\mathcal{O}(1)$ is $\Theta\left(\frac{\log(Kn)}{1-\lambda}\right)$.

Deterministic Rounding (Rabani, Sinclair, Wanka,

For any graph, the discrepancy is $\mathcal{O}\left(\frac{\log n}{1-\lambda}\right)$ after $\mathcal{O}\left(\frac{\log(Kn)}{1-\lambda}\right)$ rounds.

Load Balancing: Results

- $1-\lambda$: spectral expansion of the n-vertex regular graph
- K: discrepancy of the initial load vector
- all results hold with probability $1-o(1)$ as $n \to \infty$

This is at least the diameter of the graph!

Deterministic Rounding (Raba… air, Wanka, 1998)

For any graph, the discrepancy is $\mathcal{O}\left(\frac{\log n}{1-\lambda}\right)$ after $\mathcal{O}\left(\frac{\log(Kn)}{1-\lambda}\right)$ rounds.

Load Balancing: Results

- $1-\lambda$: spectral expansion of the n-vertex regular graph
- K: discrepancy of the initial load vector
- all results hold with probability $1-o(1)$ as $n \to \infty$

Deterministic Rounding (Rabani, Sinclair, Wanka, 1998)

For any graph, the discrepancy is $\mathcal{O}\left(\frac{\log n}{1-\lambda}\right)$ after $\mathcal{O}\left(\frac{\log(Kn)}{1-\lambda}\right)$ rounds.

Load Balancing: Results

- $1 - \lambda$: spectral expansion of the n-vertex regular graph
- K: discrepancy of the initial load vector
- all results hold with probability $1 - o(1)$ as $n \to \infty$

Deterministic Rounding (Rabani, Sinclair, Wanka, 1998)

For any graph, the discrepancy is $\mathcal{O}\left(\frac{\log n}{1-\lambda}\right)$ after $\mathcal{O}\left(\frac{\log(Kn)}{1-\lambda}\right)$ rounds.

Randomized Rounding (Friedrich, S., 2009)

For any graph, the discrepancy is $\mathcal{O}\left(\sqrt{\frac{\log n}{1-\lambda}}\right)$ after $\mathcal{O}\left(\frac{\log(Kn)}{1-\lambda}\right)$ rounds.

Load Balancing: Results

- $1-\lambda$: spectral expansion of the n-vertex regular graph
- K: discrepancy of the initial load vector
- all results hold with probability $1-o(1)$ as $n \to \infty$

Deterministic Rounding (Rabani, Sinclair, Wanka, 1998)

For any graph, the discrepancy is $\mathcal{O}\left(\frac{\log n}{1-\lambda}\right)$ after $\mathcal{O}\left(\frac{\log(Kn)}{1-\lambda}\right)$ rounds.

Randomized Rounding (Friedrich, S., 2009)

For any graph, the discrepancy is $\mathcal{O}\left(\sqrt{\frac{\log n}{1-\lambda}}\right)$ after $\mathcal{O}\left(\frac{\log(Kn)}{1-\lambda}\right)$ rounds.

Randomized Rounding (S., Sun, 2012)

For any graph, the discrepancy is $\mathcal{O}(1)$ after $\mathcal{O}\left(\frac{\log(Kn)}{1-\lambda}\right)$ rounds.

Load Balancing: Results

- $1-\lambda$: spectral expansion of the n-vertex regular graph
- K: discrepancy of the initial load vector
- all results hold w

In the continuous case, the time to reach discrepancy $\mathcal{O}(1)$ is $\Theta\left(\frac{\log(Kn)}{1-\lambda}\right)$.

Deterministic Rounding (Rabani, Sinclair, Wanka,

For any graph, the discrepancy is $\mathcal{O}\left(\frac{\log n}{1-\lambda}\right)$ after $\mathcal{O}\left(\frac{\log(Kn)}{1-\lambda}\right)$ rounds.

Randomized Rounding (Friedrich, S., 2009)

For any graph, the discrepancy is $\mathcal{O}\left(\sqrt{\frac{\log n}{1-\lambda}}\right)$ after $\mathcal{O}\left(\frac{\log(Kn)}{1-\lambda}\right)$ rounds.

Randomized Rounding (S., Sun, 2012)

For any graph, the discrepancy is $\mathcal{O}(1)$ after $\mathcal{O}\left(\frac{\log(Kn)}{1-\lambda}\right)$ rounds.

No diff. between continuous and discrete case

Load Balancing: Results

- $1-\lambda$: spectral expansion of the n-vertex regular graph
- K: discrepancy of the initial load vector
- all results hold with probability $1-o(1)$ as $n \to \infty$

Deterministic Rounding (Rabani, Sinclair, Wanka, 1998)

For any graph, the discrepancy is $\mathcal{O}\left(\frac{\log n}{1-\lambda}\right)$ after $\mathcal{O}\left(\frac{\log(Kn)}{1-\lambda}\right)$ rounds.

Randomized Rounding (Friedrich, S., 2009)

For any graph, the discrepancy is $\mathcal{O}\left(\sqrt{\frac{\log n}{1-\lambda}}\right)$ after $\mathcal{O}\left(\frac{\log(Kn)}{1-\lambda}\right)$ rounds.

Randomized Rounding (S., Sun, 2012)

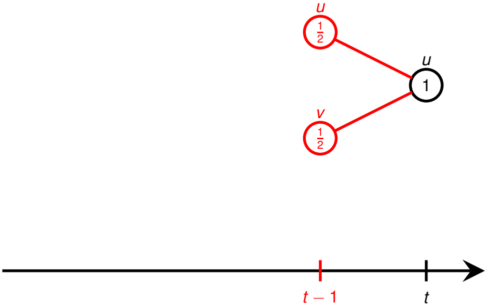
For any graph, the discrepancy is $\mathcal{O}(1)$ after $\mathcal{O}\left(\frac{\log(Kn)}{1-\lambda}\right)$ rounds.

Analyzing Recursion

x_u^t

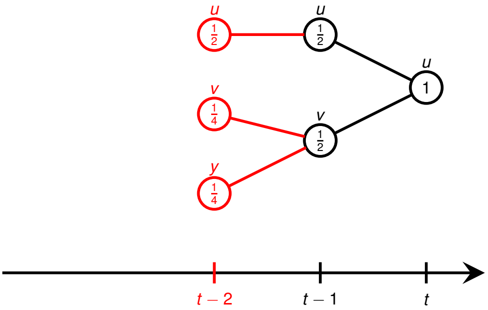
Analyzing Recursion

$$x_u^t = \frac{1}{2}x_u^{t-1} + \frac{1}{2}x_v^{t-1} + e_{u,v}^{t-1}$$



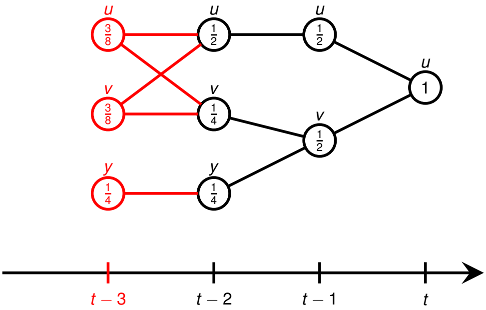
Analyzing Recursion

$$
\begin{aligned}
x_u^t &= \frac{1}{2}x_u^{t-1} + \frac{1}{2}x_v^{t-1} + e_{u,v}^{t-1} \\
&= \frac{1}{2}x_u^{t-2} + \frac{1}{4}x_v^{t-2} + \frac{1}{4}x_y^{t-2} + e_{u,v}^{t-1} + \frac{1}{2}e_{v,y}^{t-2}
\end{aligned}
$$



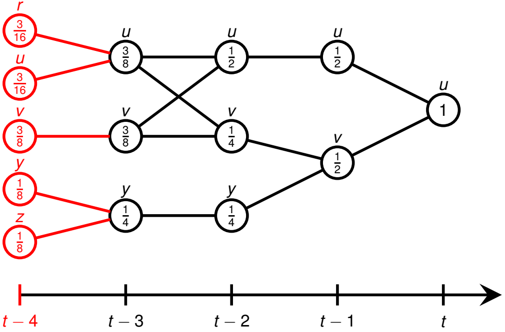
Analyzing Recursion

$$
\begin{aligned}
x_u^t &= \frac{1}{2}x_u^{t-2} + \frac{1}{4}x_v^{t-2} + \frac{1}{4}x_y^{t-2} + e_{u,v}^{t-1} + \frac{1}{2}e_{v,y}^{t-2} \\
&= \frac{3}{8}x_u^{t-3} + \frac{3}{8}x_v^{t-3} + \frac{1}{4}x_y^{t-3} + e_{u,v}^{t-1} + \frac{1}{2}e_{v,y}^{t-2} + \frac{1}{2}e_{u,v}^{t-3} - \frac{1}{4}e_{u,v}^{t-3}
\end{aligned}
$$



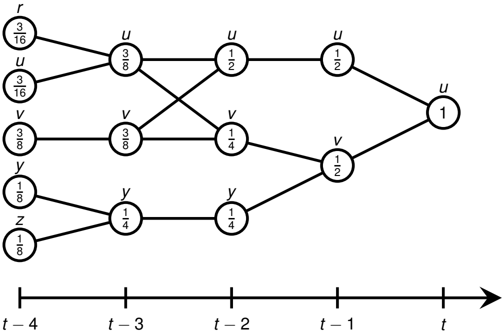
Analyzing Recursion

$$
\begin{aligned}
x_u^t &= \frac{3}{8}x_u^{t-3} + \frac{3}{8}x_v^{t-3} + \frac{1}{4}x_y^{t-3} + e_{u,v}^{t-1} + \frac{1}{2}e_{v,y}^{t-2} + \frac{1}{2}e_{u,v}^{t-3} - \frac{1}{4}e_{u,v}^{t-3} \\
&= \frac{3}{16}x_r^{t-4} + \frac{3}{16}x_u^{t-4} + \frac{3}{8}x_v^{t-4} + \frac{1}{8}x_y^{t-4} + \frac{1}{8}x_z^{t-4} \\
&\quad + e_{u,v}^{t-1} + \frac{1}{2}e_{v,y}^{t-2} + \frac{1}{2}e_{u,v}^{t-3} - \frac{1}{4}e_{u,v}^{t-3} + \frac{3}{8}e_{r,u}^{t-4} + \frac{1}{4}e_{y,z}^{t-4}
\end{aligned}
$$



Analyzing Recursion

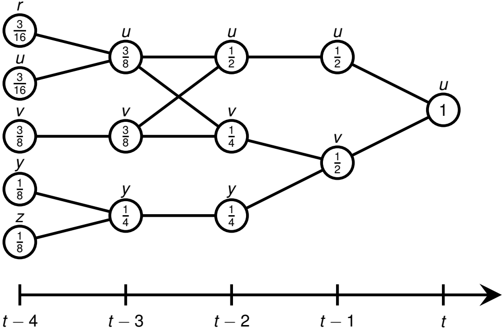
$$x_u^t = \frac{3}{16} x_r^{t-4} + \frac{3}{16} x_u^{t-4} + \frac{3}{8} x_v^{t-4} + \frac{1}{8} x_y^{t-4} + \frac{1}{8} x_z^{t-4}$$
$$+ e_{u,v}^{t-1} + \frac{1}{2} e_{v,y}^{t-2} + \frac{1}{2} e_{u,v}^{t-3} - \frac{1}{4} e_{u,v}^{t-3} + \frac{3}{8} e_{r,u}^{t-4} + \frac{1}{4} e_{y,z}^{t-4}$$



- continuous part and deviation

Analyzing Recursion

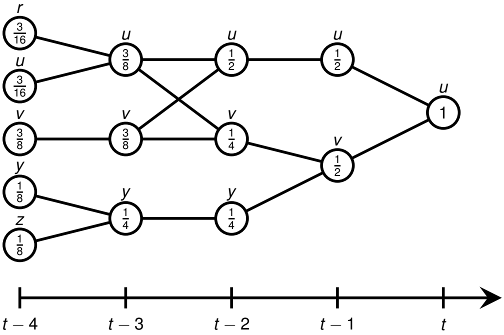
$$x_u^t = \frac{3}{16} x_r^{t-4} + \frac{3}{16} x_u^{t-4} + \frac{3}{8} x_v^{t-4} + \frac{1}{8} x_y^{t-4} + \frac{1}{8} x_z^{t-4}$$
$$+ e_{u,v}^{t-1} + \frac{1}{2} e_{v,y}^{t-2} + \frac{1}{2} e_{u,v}^{t-3} - \frac{1}{4} e_{u,v}^{t-3} + \frac{3}{8} e_{r,u}^{t-4} + \frac{1}{4} e_{y,z}^{t-4}$$



- continuous part and deviation
- coefficients of x^t converge

Analyzing Recursion

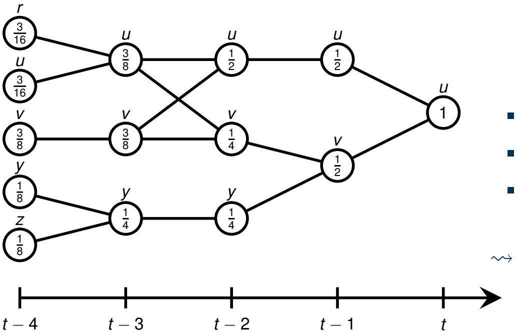
$$x_u^t = \frac{3}{16}x_r^{t-4} + \frac{3}{16}x_u^{t-4} + \frac{3}{8}x_v^{t-4} + \frac{1}{8}x_y^{t-4} + \frac{1}{8}x_z^{t-4}$$
$$+ 1e_{u,v}^{t-1} + \frac{1}{2}e_{v,y}^{t-2} + \frac{1}{2}e_{u,v}^{t-3} - \frac{1}{4}e_{u,v}^{t-3} + \frac{3}{8}e_{r,u}^{t-4} + \frac{1}{4}e_{y,z}^{t-4}$$



- continuous part and deviation
- coefficients of x^t converge
- coefficients of e's keep track of this convergence

Analyzing Recursion

$$x_u^t = \frac{3}{16}x_r^{t-4} + \frac{3}{16}x_u^{t-4} + \frac{3}{8}x_v^{t-4} + \frac{1}{8}x_y^{t-4} + \frac{1}{8}x_z^{t-4}$$
$$+ 1e_{u,v}^{t-1} + \frac{1}{2}e_{v,y}^{t-2} + \frac{1}{2}e_{u,v}^{t-3} - \frac{1}{4}e_{u,v}^{t-3} + \frac{3}{8}e_{r,u}^{t-4} + \frac{1}{4}e_{y,z}^{t-4}$$

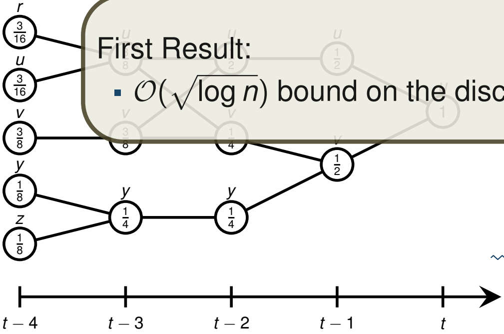


- continuous part and deviation
- coefficients of x^t converge
- coefficients of e's keep track of this convergence

⇝ deviation part dominated by $\mathcal{N}(0,2)$

Analyzing Recursion

$$x_u^t = \frac{3}{16}x_r^{t-4} + \frac{3}{16}x_u^{t-4} + \frac{3}{8}x_v^{t-4} + \frac{1}{8}x_y^{t-4} + \frac{1}{8}x_z^{t-4}$$
$$+ 1e_{u,v}^{t-1} + \frac{1}{2}e_{v,y}^{t-2} + \frac{1}{2}e_{u,v}^{t-3} - \frac{1}{4}e_{u,v}^{t-3} + \frac{3}{8}e_{r,u}^{t-4} + \frac{1}{4}e_{y,z}^{t-4}$$



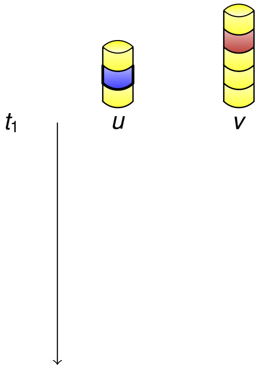
First Result:

- $\mathcal{O}(\sqrt{\log n})$ bound on the discrepancy

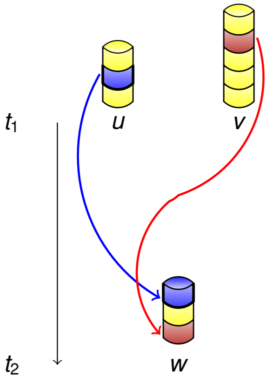
- continuous part and deviation
- coefficients of x^t converge
- coefficients of e's keep track of this convergence

$\rightsquigarrow$ deviation part dominated by $\mathcal{N}(0, 2)$

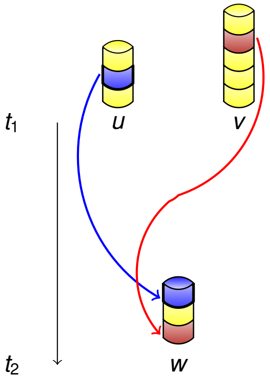
Step 1: Token Movements as Random Walks



Step 1: Token Movements as Random Walks



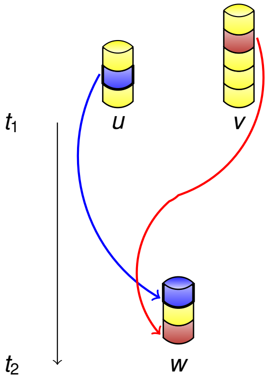
Step 1: Token Movements as Random Walks



Negative Correlation

- **Pr** [blue token on w] $= \mathbf{M}_{u,w}^{[t_1,t_2]}$, **Pr** [red token on w] $= \mathbf{M}_{v,w}^{[t_1,t_2]}$

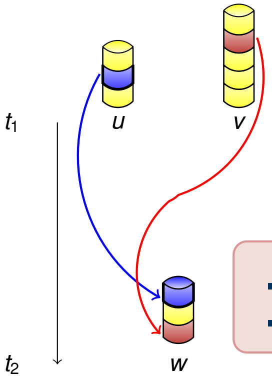
Step 1: Token Movements as Random Walks



Negative Correlation

- $\mathbf{Pr}\left[\text{blue token on } w\right] = \mathbf{M}_{u,w}^{[t_1,t_2]}$, $\mathbf{Pr}\left[\text{red token on } w\right] = \mathbf{M}_{v,w}^{[t_1,t_2]}$
- $\mathbf{Pr}\left[\text{blue and red token are on } w\right] \leqslant \mathbf{M}_{u,w}^{[t_1,t_2]} \cdot \mathbf{M}_{v,w}^{[t_1,t_2]}$

Step 1: Token Movements as Random Walks

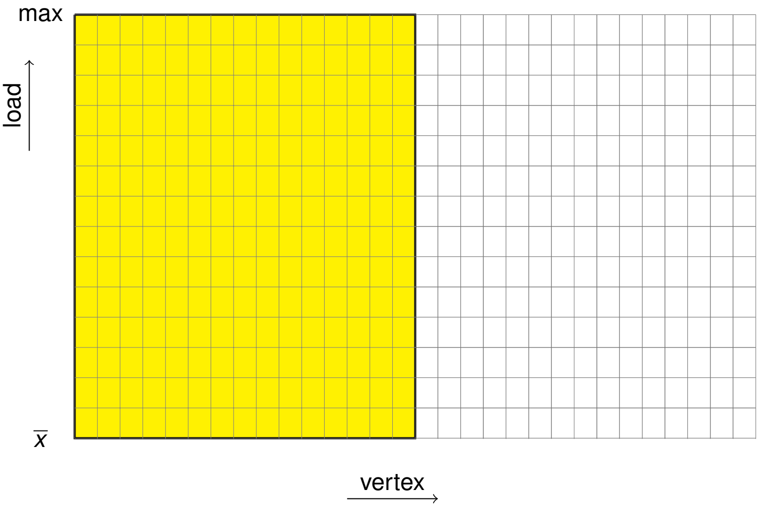


- Allows use of Chernoff bounds for independent r.v.'s
- Fewer tokens yields better concentration

Negative Correlation

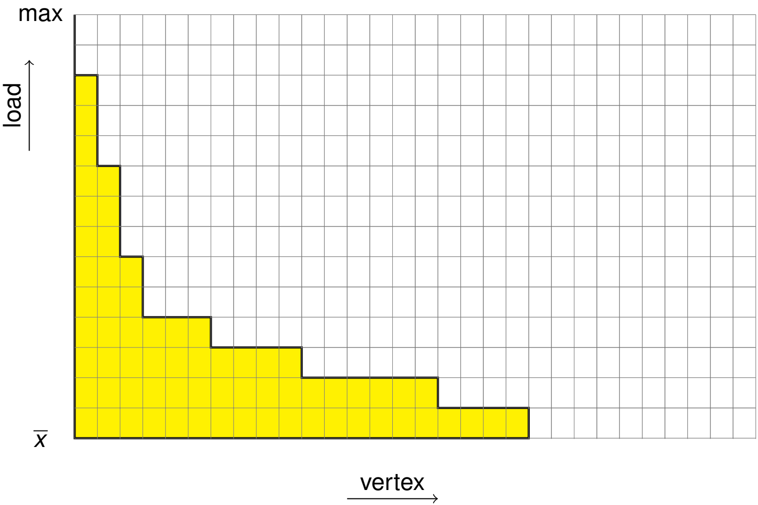
- $\mathbf{Pr}\,[\,\text{blue token on } w\,] = \mathbf{M}_{u,w}^{[t_1,t_2]}$, $\quad \mathbf{Pr}\,[\,\text{red token on } w\,] = \mathbf{M}_{v,w}^{[t_1,t_2]}$
- $\mathbf{Pr}\,[\,\text{blue and red token are on } w\,] \leqslant \mathbf{M}_{u,w}^{[t_1,t_2]} \cdot \mathbf{M}_{v,w}^{[t_1,t_2]}$

Step 2: Sparsification of the Load Vector

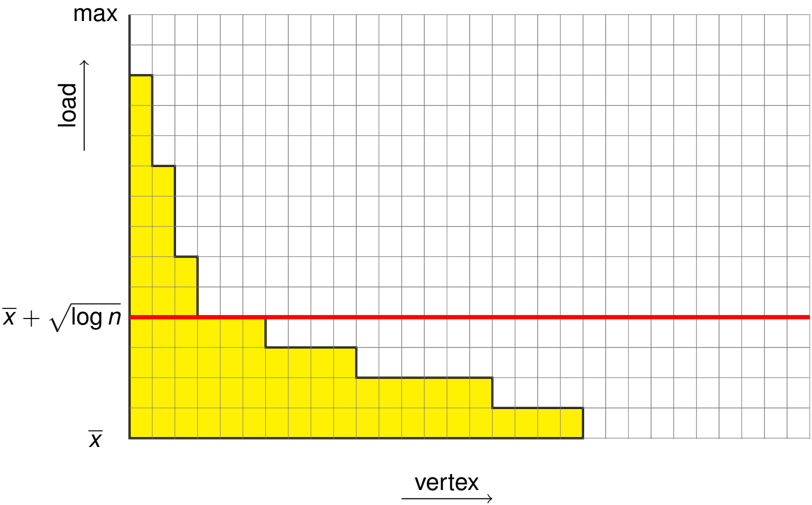


Step 2: Sparsification of the Load Vector

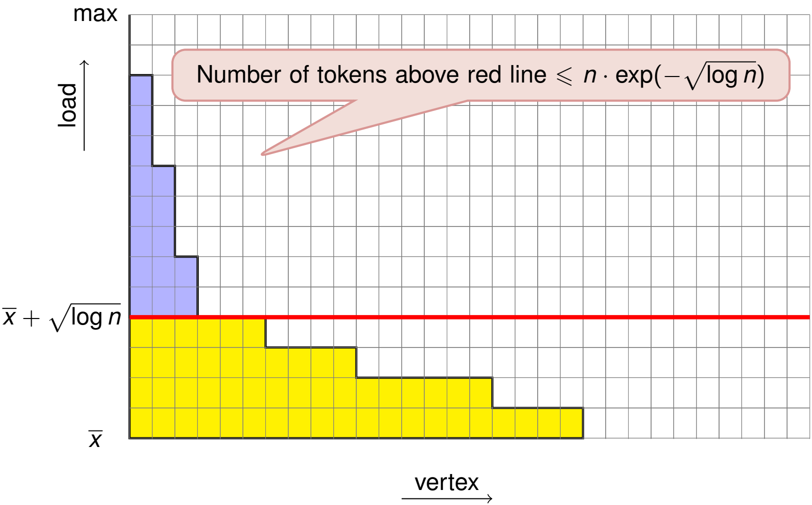
Step 2: Sparsification of the Load Vector



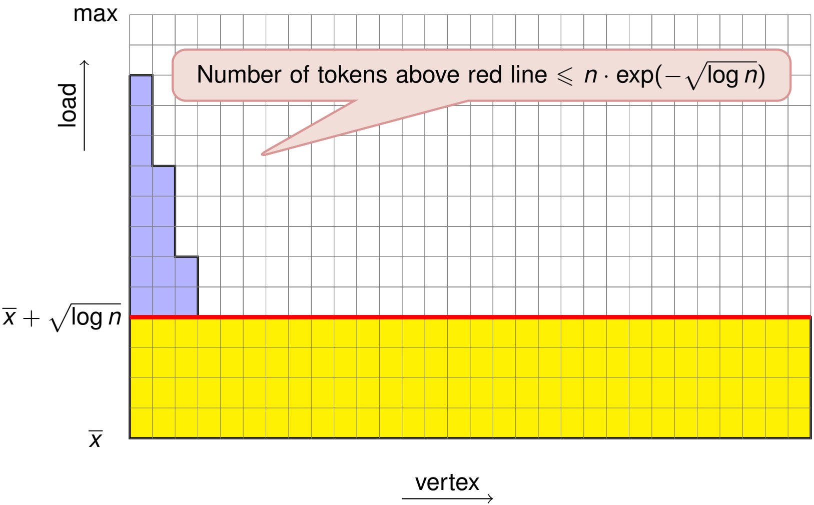
Step 2: Sparsification of the Load Vector



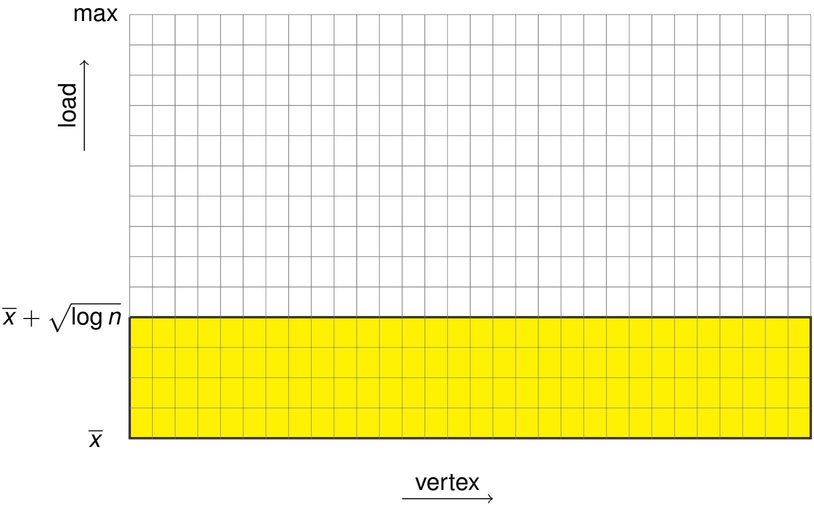
Step 2: Sparsification of the Load Vector



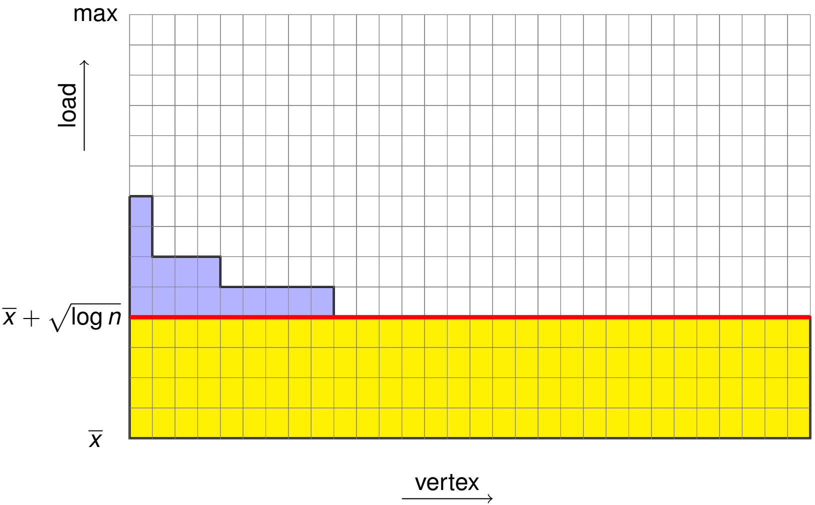
Step 2: Sparsification of the Load Vector



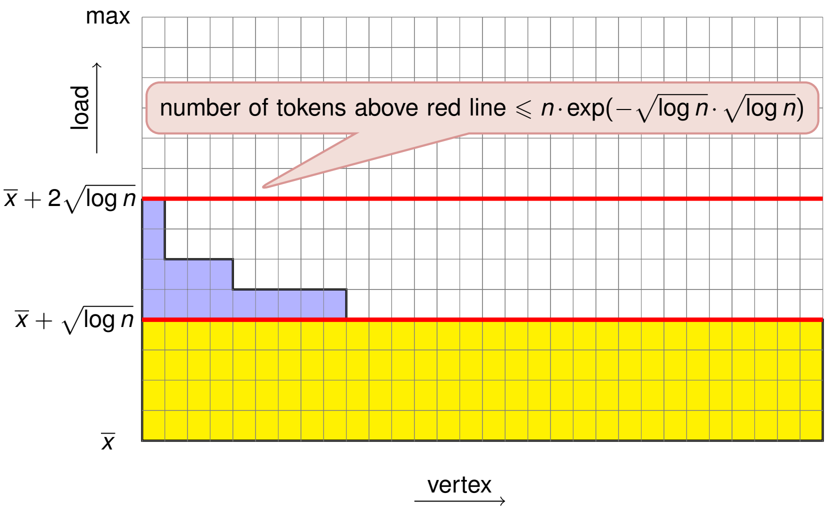
Step 2: Sparsification of the Load Vector



Step 2: Sparsification of the Load Vector

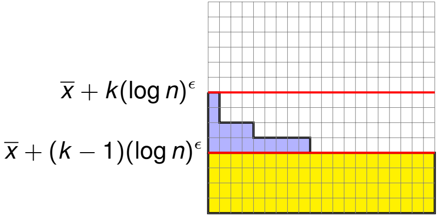


Step 2: Sparsification of the Load Vector



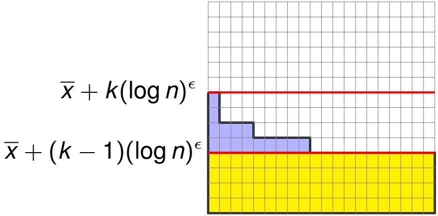
Step 2: Sparsification of the Load Vector

- Let $\epsilon > 0$ be any value
- After k iterations, number of blue tokens is $\leqslant n \cdot \exp(-(\log n)^{\epsilon \cdot k})$.



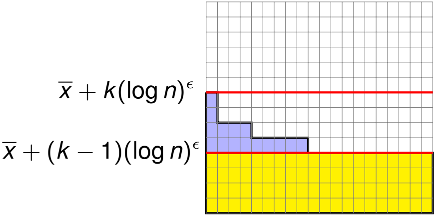
Step 2: Sparsification of the Load Vector

- Let $\epsilon > 0$ be any value
- After k iterations, number of blue tokens is $\leqslant n \cdot \exp(-(\log n)^{\epsilon \cdot k})$.
- Choosing $k = \frac{1}{\epsilon}$ yields a maximum load of $\overline{x} + k \cdot (\log n)^{\epsilon}$



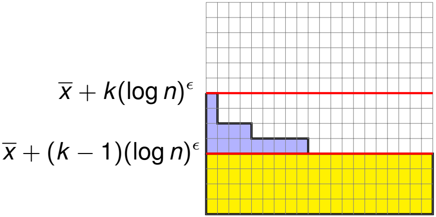
Step 2: Sparsification of the Load Vector

- Let $\epsilon > 0$ be any value
- After k iterations, number of blue tokens is $\leqslant n \cdot \exp(-(\log n)^{\epsilon \cdot k})$.
- Choosing $k = \frac{1}{\epsilon}$ yields a maximum load of $\overline{x} + k \cdot (\log n)^{\epsilon}$
- lower bound on minimum load by symmetry



Step 2: Sparsification of the Load Vector

- Let $\epsilon > 0$ be any value
- After k iterations, number of blue tokens is $\leqslant n \cdot \exp(-(\log n)^{\epsilon \cdot k})$.
- Choosing $k = \frac{1}{\epsilon}$ yields a maximum load of $\overline{x} + k \cdot (\log n)^{\epsilon}$
- lower bound on minimum load
  by symmetry

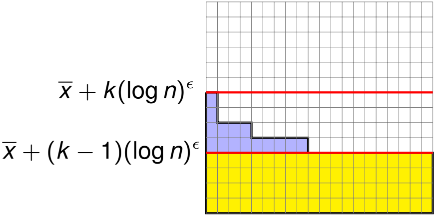


Preliminary Results (also for non-regular graphs)

- After $\mathcal{O}\left(\frac{1}{\epsilon} \cdot \frac{\log(Kn)}{1-\lambda}\right)$ rounds, discrepancy is $\mathcal{O}((\log n)^{\epsilon})$.

Step 2: Sparsification of the Load Vector

- Let $\epsilon > 0$ be any value
- After k iterations, number of blue tokens is $\leqslant n \cdot \exp(-(\log n)^{\epsilon \cdot k})$.
- Choosing $k = \frac{1}{\epsilon}$ yields a maximum load of $\overline{x} + k \cdot (\log n)^{\epsilon}$
- lower bound on minimum load by symmetry

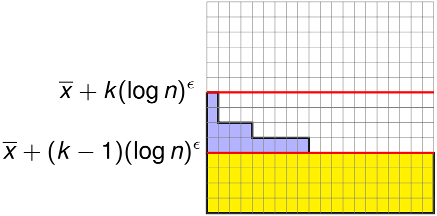


Preliminary Results (also for non-regular graphs)

- After $\mathcal{O}\left(\frac{1}{\epsilon} \cdot \frac{\log(Kn)}{1-\lambda}\right)$ rounds, discrepancy is $\mathcal{O}((\log n)^{\epsilon})$.
- After $\mathcal{O}\left(\log\log n \cdot \frac{\log(Kn)}{1-\lambda}\right)$ rounds, discrepancy is $\mathcal{O}(\log\log n)$.

Step 2: Sparsification of the Load Vector

- Let $\epsilon > 0$ be any value
- After k iterations, number of blue tokens is $\leqslant n \cdot \exp(-(\log n)^{\epsilon \cdot k})$.
- Choosing $k = \frac{1}{\epsilon}$ yields a maximum load of $\overline{x} + k \cdot (\log n)^{\epsilon}$
- lower bound on minimum load by symmetry



Preliminary Results (also for non-regular graphs)

- After $\mathcal{O}\left(\frac{1}{\epsilon} \cdot \frac{\log(Kn)}{1-\lambda}\right)$ rounds, discrepancy is $\mathcal{O}((\log n)^{\epsilon})$.
- After $\mathcal{O}\left(\log\log n \cdot \frac{\log(Kn)}{1-\lambda}\right)$ rounds, discrepancy is $\mathcal{O}(\log\log n)$.

(Much) more work required to get constant discrepancy...

Outline

Introduction

Load Balancing on Hypercubes

Load Balancing on Arbitrary Graphs

Conclusions

Hypercube

Results

- Discrepancy of $\log\log n + \Theta(1)$ after $\log_2 n$ rounds
- Discrepancy of 3 after $\log_2 n + o(\log n)$ rounds
- Discrepancy of 2 after $3\log_2 n$ rounds

Conclusion

- Very good understanding
- Since $\log_2 n$ rounds are necessary, hypercube is "optimal network"
- Proofs: Chernoff bounds using independence

Hypercube

Results

- Discrepancy of $\log\log n + \Theta(1)$ after $\log_2 n$ rounds
- Discrepancy of 3 after $\log_2 n + o(\log n)$ rounds
- Discrepancy of 2 after $3\log_2 n$ rounds

Conclusion

- Very good understanding
- Since $\log_2 n$ rounds are necessary, hypercube is "optimal network"
- Proofs: Chernoff bounds using independence

Arbitrary Graphs

Results for Random Matchings

- Constant Discrepancy in $\mathcal{O}\left(\frac{\log(Kn)}{1-\lambda}\right)$ rounds for any regular graph

Techniques

- Movements of Tokens instead of rounding errors
- Sparsification: Reduce general problem to sparse vectors

Future Work

- Derandomization
  - Random Matchings $\rightsquigarrow$ Balancing Circuit Model
  - Randomized Rounding $\rightsquigarrow$ (Deterministic) Rounding with Constraints

Future Work

- Derandomization
  - Random Matchings $\rightsquigarrow$ Balancing Circuit Model
  - Randomized Rounding $\rightsquigarrow$ (Deterministic) Rounding with Constraints
- Dynamic Settings
  - Edges of the graphs change
  - Jobs are processed or created during execution

Future Work

- Derandomization
  - Random Matchings $\rightsquigarrow$ Balancing Circuit Model
  - Randomized Rounding $\rightsquigarrow$ (Deterministic) Rounding with Constraints
- Dynamic Settings
  - Edges of the graphs change
  - Jobs are processed or created during execution
- Heterogenous Settings
  - Non-Regular Graphs
  - Different Weights, Different Speeds

Future Work

- Derandomization
  - Random Matchings $\rightsquigarrow$ Balancing Circuit Model
  - Randomized Rounding $\rightsquigarrow$ (Deterministic) Rounding with Constraints
- Dynamic Settings
  - Edges of the graphs change
  - Jobs are processed or created during execution
- Heterogenous Settings
  - Non-Regular Graphs
  - Different Weights, Different Speeds

Thank you for your attention!