Usklajevanje mnenj s pomočjo matematike

Matjaž Krnc

Izleti v matematično vesolje
Koper, 17.1.2018

Oris vsebine

UNIVERSITÄT SALZBURG

Consensus doesn't happen by magic... You have to drive to it.

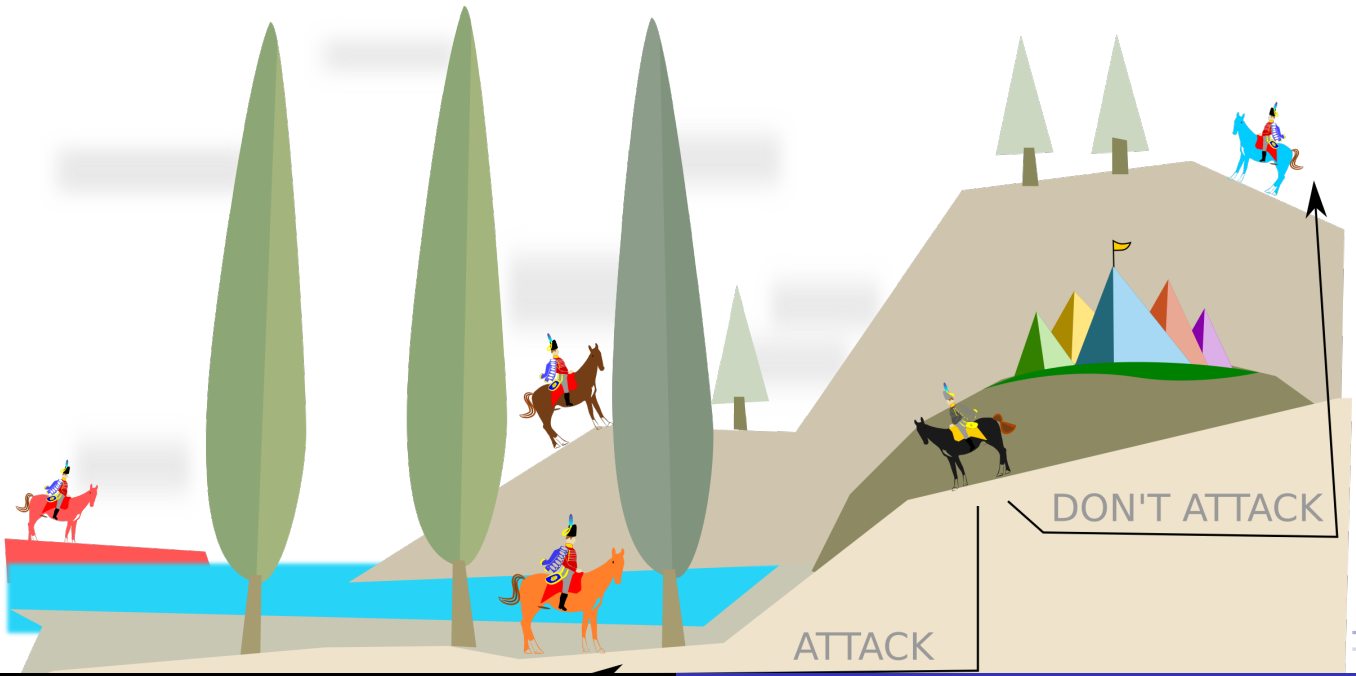
Christine Quinn Read

Oris vsebine

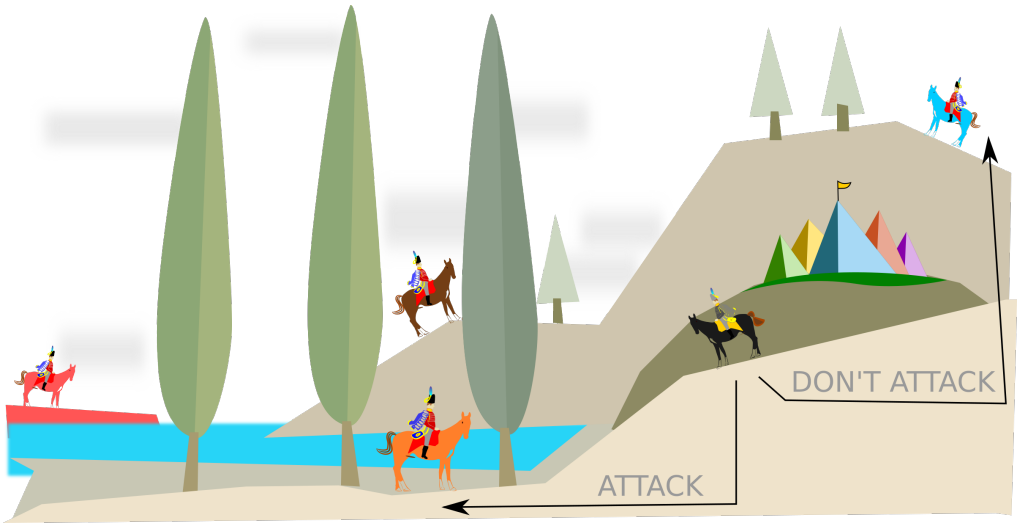
UNIVERSITÄT SALZBURG

Problem bizantinskih generalov

- Vsaki enoti bizantinske vojske poveljuje drug general; nekateri od njih so podkupljeni.
- Različne enote so razporejene okrog napadenega mesta.
- Omejena komunikacija! Generali komunicirajo s pomočjo kurirjev.



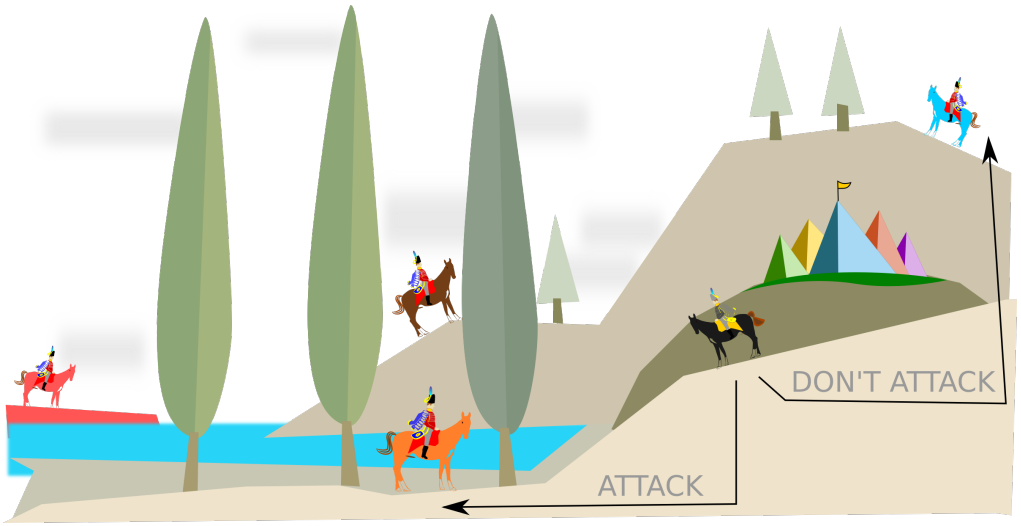
Problem bizantinskih generalov



Cilji:

- Vsi pošteni generali se odločijo enako.
- Odločitev je sprejeta v doglednem času.
- Majhno število izdajalcev ne more prisiliti poštenih generalov k slabemu načrtu.

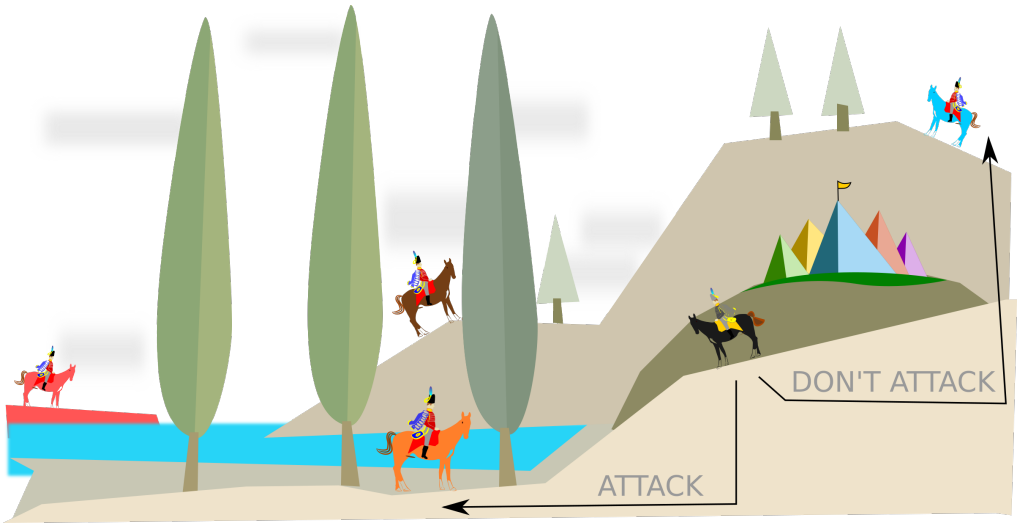
Problem bizantinskih generalov



Cilji:

- Vsi pošteni generali se odločijo enako.
- Odločitev je sprejeta v doglednem času.
- Majhno število izdajalcev ne more prisiliti poštenih generalov k slabemu načrtu.

Problem bizantinskih generalov



Cilji:

- Vsi pošteni generali se odločijo enako.
- Odločitev je sprejeta v doglednem času.
- Majhno število izdajalcev ne more prisiliti poštenih generalov k slabemu načrtu.

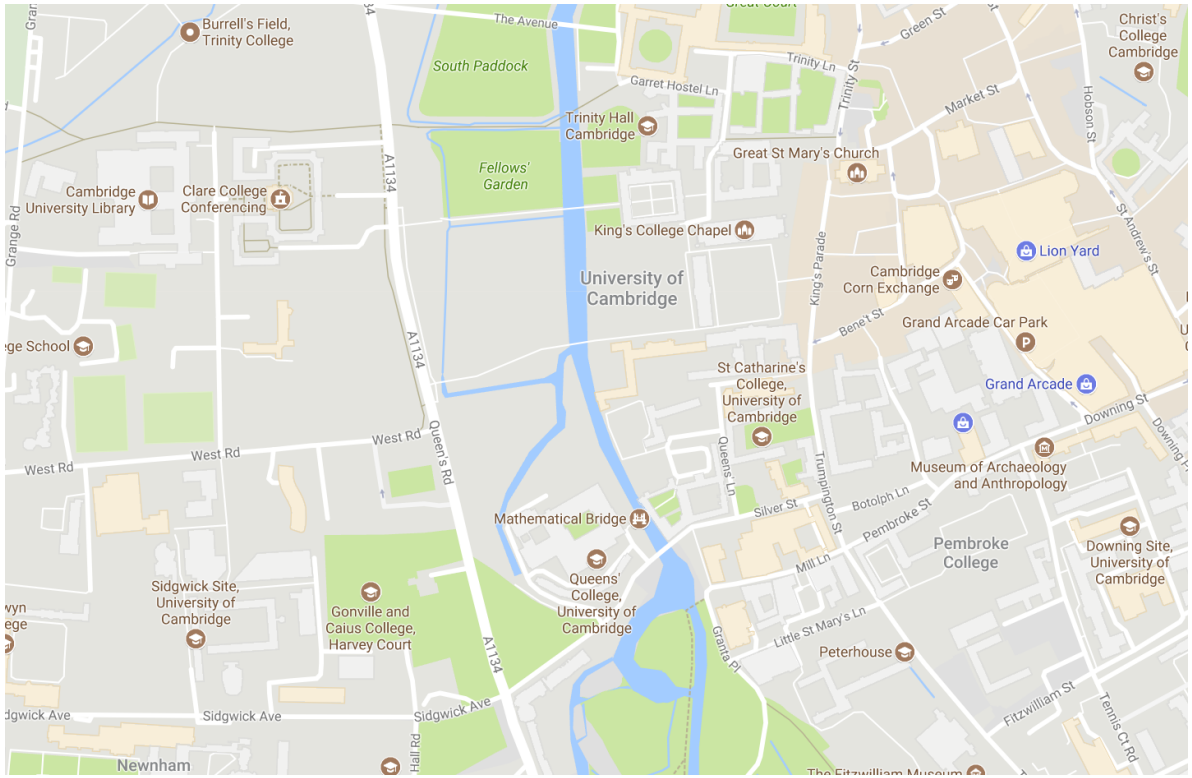
UNIVERSITÄT SALZBURG

Oris vsebine

UNIVERSITÄT SALZBURG

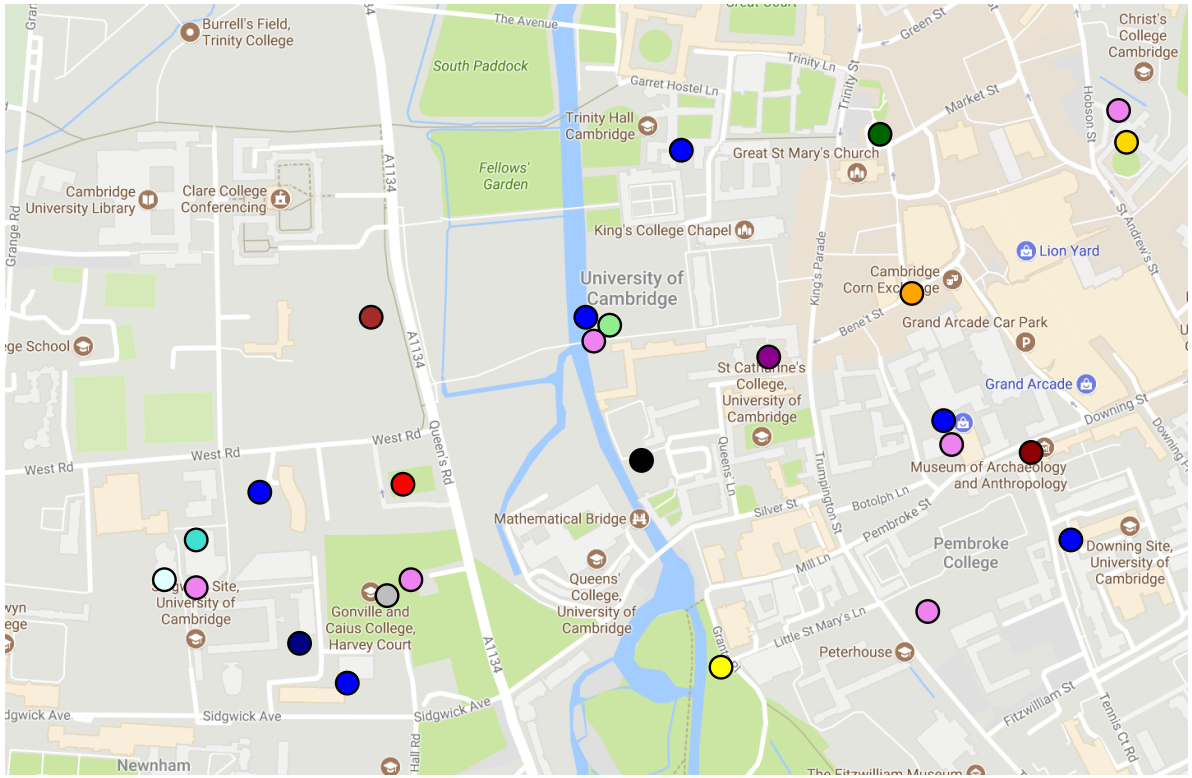
Problem naključnega bara

- Skupina prijateljev rada skupaj pohajkuje ob petkih zvečer.
- Vsak se je vsak pripravljen prilagodit skupini, a ima tudi neko začetno preferenco, ter telefon.
- Ko v petek pade mrak, začne vsak od prijateljev z naključnim kontaktiranjem ostalih, ter poizveduje o mnenju.



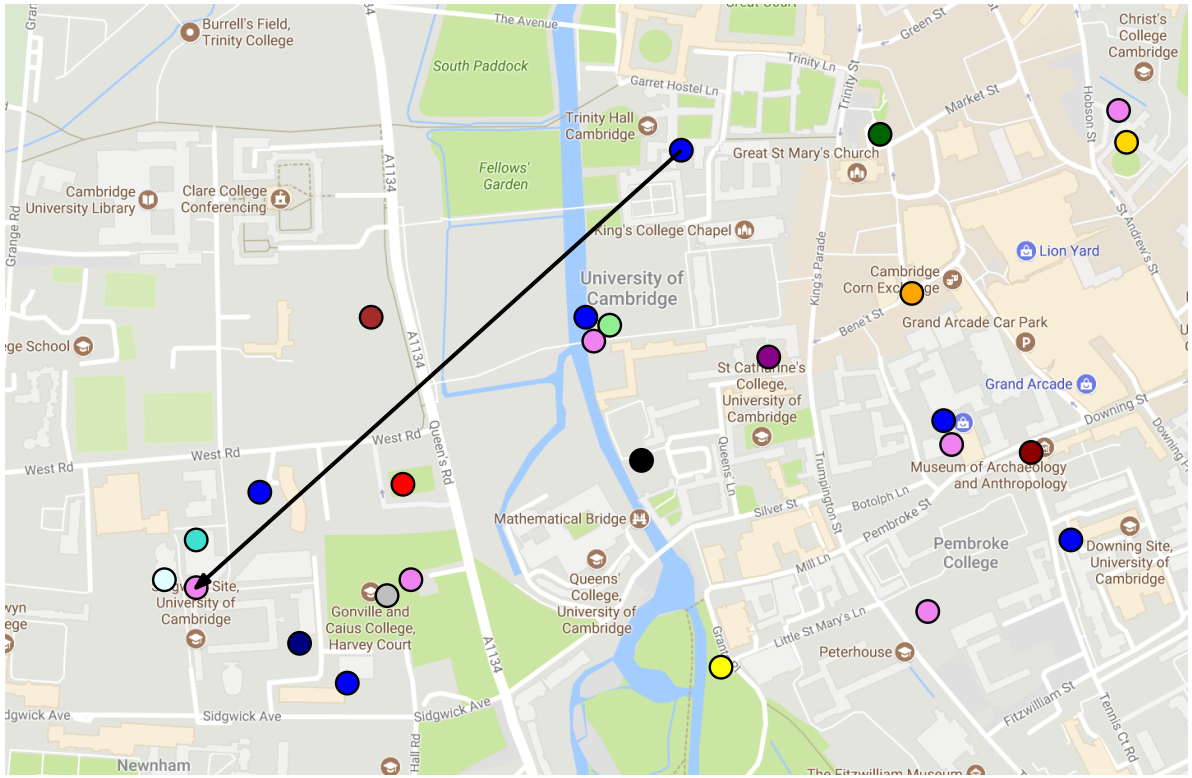
Problem naključnega bara

- Skupina prijateljev rada skupaj pohajkuje ob petkih zvečer.
- Vsak se je vsak pripravljen prilagodit skupini, a ima tudi neko začetno preferenco, ter telefon.
- Ko v petek pade mrak, začne vsak od prijateljev z naključnim kontaktiranjem ostalih, ter poizveduje o mnenju.



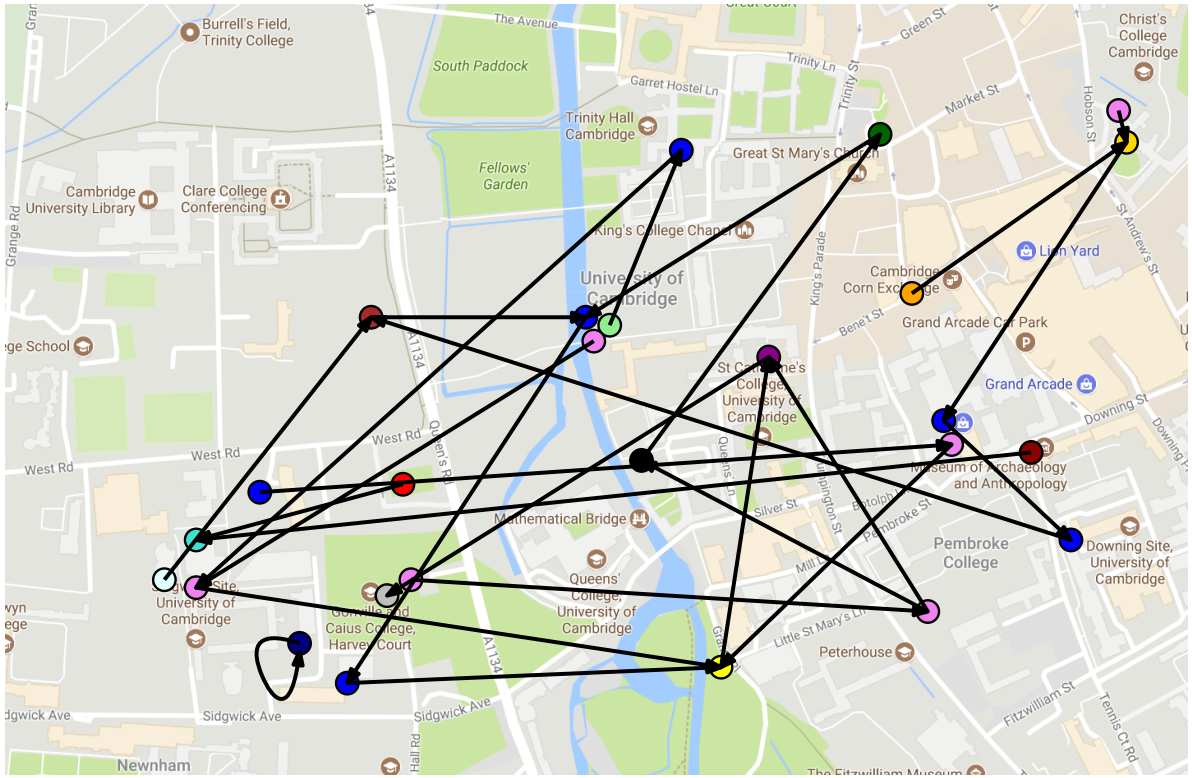
Problem naključnega bara

- Skupina prijateljev rada skupaj pohajkuje ob petkih zvečer.
- Vsak se je vsak pripravljen prilagodit skupini, a ima tudi neko začetno preferenco, ter telefon.
- Ko v petek pade mrak, začne vsak od prijateljev z naključnim kontaktiranjem ostalih, ter poizveduje o mnenju.



Problem naključnega bara

- Skupina prijateljev rada skupaj pohajkuje ob petkih zvečer.
- Vsak se je vsak pripravljen prilagodit skupini, a ima tudi neko začetno preferenco, ter telefon.
- Ko v petek pade mrak, začne vsak od prijateljev z naključnim kontaktiranjem ostalih, ter poizveduje o mnenju.



Oris vsebine

UNIVERSITÄT SALZBURG

Kakšna naj bo rešitev?

Cilji:

- Robusten postopek ki zagotovi konsenz!
- Hitrost usklajevanja mora biti hitra, predvidljiva, merljiva.
- Algoritem deluje učinkovito tudi, če ga uporabimo na velikih omrežjih.
- Končna aktivnost mora sovpadati z začetno prevladujočo.
- Osebe, ki morebiti niso dovolj prisebne, da bi upoštevale dogovorjen postopek, sistema ne zavedejo.

Uporaba:

- Sinhronizacija ur
- Volitve
- PageRank
- Uravnoteženje obremenitev

UNIVERSITÄT SALZBURG

Kakšna naj bo rešitev?

Cilji:

- Robusten postopek ki zagotovi konsenz!
- Hitrost usklajevanja mora biti hitra, predvidljiva, merljiva.
- Algoritem deluje učinkovito tudi, če ga uporabimo na velikih omrežjih.
- Končna aktivnost mora sovpadati z začetno prevladujočo.
- Osebe, ki morebiti niso dovolj prisebne, da bi upoštevale dogovorjen postopek, sistema ne zavedejo.

Uporaba:

- Sinhronizacija ur
- Volitve
- PageRank
- Uravnoteženje obremenitev

UNIVERSITÄT SALZBURG

Kakšna naj bo rešitev?

Cilji:

- Robusten postopek ki zagotovi konsenz!
- Hitrost usklajevanja mora biti hitra, predvidljiva, merljiva.
- Algoritem deluje učinkovito tudi, če ga uporabimo na velikih omrežjih.
- Končna aktivnost mora sovpadati z začetno prevladujočo.
- Osebe, ki morebiti niso dovolj prisebne, da bi upoštevale dogovorjen postopek, sistema ne zavedejo.

Uporaba:

- Sinhronizacija ur
- Volitve
- PageRank
- Uravnoteženje obremenitev

UNIVERSITÄT SALZBURG

Kakšna naj bo rešitev?

Cilji:

- Robusten postopek ki zagotovi konsenz!
- Hitrost usklajevanja mora biti hitra, predvidljiva, merljiva.
- Algoritem deluje učinkovito tudi, če ga uporabimo na velikih omrežjih.
- Končna aktivnost mora sovpadati z začetno prevladujočo.
- Osebe, ki morebiti niso dovolj prisebne, da bi upoštevale dogovorjen postopek, sistema ne zavedejo.

Uporaba:

- Sinhronizacija ur
- Volitve
- PageRank
- Uravnoteženje obremenitev

UNIVERSITÄT SALZBURG

Kakšna naj bo rešitev?

Cilji:

- Robusten postopek ki zagotovi konsenz!
- Hitrost usklajevanja mora biti hitra, predvidljiva, merljiva.
- Algoritem deluje učinkovito tudi, če ga uporabimo na velikih omrežjih.
- Končna aktivnost mora sovpadati z začetno prevladujočo.
- Osebe, ki morebiti niso dovolj prisebne, da bi upoštevale dogovorjen postopek, sistema ne zavedejo.

Uporaba:

- Sinhronizacija ur
- Volitve
- PageRank
- Uravnoteženje obremenitev

UNIVERSITÄT SALZBURG

Kakšna naj bo rešitev?

Cilji:

- Robusten postopek ki zagotovi konsenz!
- Hitrost usklajevanja mora biti hitra, predvidljiva, merljiva.
- Algoritem deluje učinkovito tudi, če ga uporabimo na velikih omrežjih.
- Končna aktivnost mora sovpadati z začetno prevladujočo.
- Osebe, ki morebiti niso dovolj prisebne, da bi upoštevale dogovorjen postopek, sistema ne zavedejo.

Uporaba:

- Sinhronizacija ur
- Volitve
- PageRank
- Uravnoteženje obremenitev

UNIVERSITÄT SALZBURG

Kakšna naj bo rešitev?

Cilji:

- Robusten postopek ki zagotovi konsenz!
- Hitrost usklajevanja mora biti hitra, predvidljiva, merljiva.
- Algoritem deluje učinkovito tudi, če ga uporabimo na velikih omrežjih.
- Končna aktivnost mora sovpadati z začetno prevladujočo.
- Osebe, ki morebiti niso dovolj prisebne, da bi upoštevale dogovorjen postopek, sistema ne zavedejo.

Uporaba:

- Sinhronizacija ur
- Volitve
- PageRank
- Uravnoteženje obremenitev

UNIVERSITÄT SALZBURG

Kakšna naj bo rešitev?

Cilji:

- Robusten postopek ki zagotovi konsenz!
- Hitrost usklajevanja mora biti hitra, predvidljiva, merljiva.
- Algoritem deluje učinkovito tudi, če ga uporabimo na velikih omrežjih.
- Končna aktivnost mora sovpadati z začetno prevladujočo.
- Osebe, ki morebiti niso dovolj prisebne, da bi upoštevale dogovorjen postopek, sistema ne zavedejo.

Uporaba:

- Sinhronizacija ur
- Volitve
- PageRank
- Uravnoteženje obremenitev

UNIVERSITÄT SALZBURG

Kakšna naj bo rešitev?

Cilji:

- Robusten postopek ki zagotovi konsenz!
- Hitrost usklajevanja mora biti hitra, predvidljiva, merljiva.
- Algoritem deluje učinkovito tudi, če ga uporabimo na velikih omrežjih.
- Končna aktivnost mora sovpadati z začetno prevladujočo.
- Osebe, ki morebiti niso dovolj prisebne, da bi upoštevale dogovorjen postopek, sistema ne zavedejo.

Uporaba:

- Sinhronizacija ur
- Volitve
- PageRank
- Uravnoteženje obremenitev

UNIVERSITÄT SALZBURG

Kakšna naj bo rešitev?

Cilji:

- Robusten postopek ki zagotovi konsenz!
- Hitrost usklajevanja mora biti hitra, predvidljiva, merljiva.
- Algoritem deluje učinkovito tudi, če ga uporabimo na velikih omrežjih.
- Končna aktivnost mora sovpadati z začetno prevladujočo.
- Osebe, ki morebiti niso dovolj prisebne, da bi upoštevale dogovorjen postopek, sistema ne zavedejo.

Uporaba:

- Sinhronizacija ur
- Volitve
- PageRank
- Uravnoteženje obremenitev


UNIVERSITÄT SALZBURG

Vloga konsenza v Blockchain-u

- Omrežje potrebuje fiksni čas, da se nov blok razširi po omrežju.
- V Blockchain-u lahko imamo v nekem trenutku različne potrjene zadnje bloke – potreba po konsenzu!

Primer.

V omrežju se nahajata dve enako dolgi (najdaljši) verziji Blockchain-a, recimo jima $\mathscr{B}_1$ in $\mathscr{B}_2$, ki se razlikujeta le v zadnjem bloku, pri čemer je delež sistema z verzijo $\mathscr{B}_2$ le 1%. Kakšne so naše prednosti če rudarimo v bloku $\mathscr{B}_1$?

UNIVERSITÄT SALZBURG

Vloga konsenza v Blockchain-u

- Omrežje potrebuje fiksni čas, da se nov blok razširi po omrežju.
- V Blockchain-u lahko imamo v nekem trenutku različne potrjene zadnje bloke – potreba po konsenzu!

Primer.

V omrežju se nahajata dve enako dolgi (najdaljši) verziji Blockchain-a, recimo jima $\mathscr{B}_1$ in $\mathscr{B}_2$, ki se razlikujeta le v zadnjem bloku, pri čemer je delež sistema z verzijo $\mathscr{B}_2$ le 1%. Kakšne so naše prednosti če rudarimo v bloku $\mathscr{B}_1$?

UNIVERSITÄT SALZBURG

Vloga konsenza v Blockchain-u

- Omrežje potrebuje fiksni čas, da se nov blok razširi po omrežju.
- V Blockchain-u lahko imamo v nekem trenutku različne potrjene zadnje bloke – potreba po konsenzu!

Primer.

V omrežju se nahajata dve enako dolgi (najdaljši) verziji Blockchain-a, recimo jima $\mathscr{B}_1$ in $\mathscr{B}_2$, ki se razlikujeta le v zadnjem bloku, pri čemer je delež sistema z verzijo $\mathscr{B}_2$ le 1%. Kakšne so naše prednosti če rudarimo v bloku $\mathscr{B}_1$?

UNIVERSITÄT SALZBURG

Vloga konsenza v Blockchain-u

- Omrežje potrebuje fiksni čas, da se nov blok razširi po omrežju.
- V Blockchain-u lahko imamo v nekem trenutku različne potrjene zadnje bloke – potreba po konsenzu!

Primer.

V omrežju se nahajata dve enako dolgi (najdaljši) verziji Blockchain-a, recimo jima $\mathscr{B}_1$ in $\mathscr{B}_2$, ki se razlikujeta le v zadnjem bloku, pri čemer je delež sistema z verzijo $\mathscr{B}_2$ le 1%. Kakšne so naše prednosti če rudarimo v bloku $\mathscr{B}_1$?

UNIVERSITÄT SALZBURG

Oris vsebine

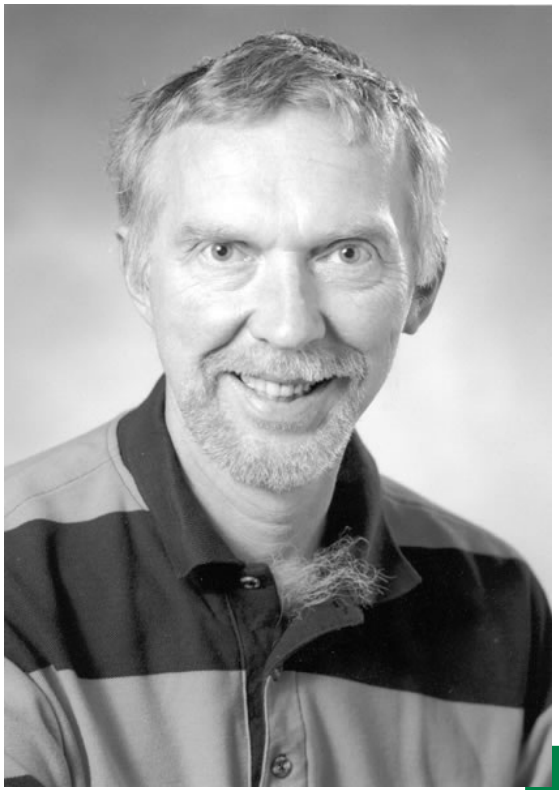
UNIVERSITÄT SALZBURG

Opis matematičnega modela

Sinhronizirani model

Model naključnih telefonskih klicev (Demers et al., 1987; Karp et al., 2000)

- Množica n oseb komunicira med sabo,
- njihov postopek teče vzporedno ter sinhronizirano v časovnih enotah.
- V vsaki čas. enoti se vzpostavi naključna komunikacija s katerokoli osebo.



UNIVERSITÄT SALZBURG

Opis matematičnega modela

Sinhronizirani model

Dodatne sposobnosti vozlišč:

- Poznajo število n?
- Imajo lokalni spomin?
- Možen naključni stik s konstantno mnogo ljudmi?
- Imajo globalni spomin?
- Se zavedajo začetne "razpršenosti"?
- Zmorejo generirati naključno število?

UNIVERSITÄT SALZBURG

Problem konsenza

Problem:

V omrežju imamo množico n vozlišč, ki imajo dodeljeno začetno mnenje/barvo iz množice $\{1, \ldots, k\}$. Najdi enoličen protokol, po katerem naj se vozlišča ravnajo, tako da se mnanja poenotijo k začetno-dominantnemu mnenju/barvi.

Od česa je t.i. *čas poenotenja* odvisen?

Problem konsenza

Problem:

V omrežju imamo množico n vozlišč, ki imajo dodeljeno začetno mnenje/barvo iz množice $\{1, \ldots, k\}$. Najdi enoličen protokol, po katerem naj se vozlišča ravnajo, tako da se mnanja poenotijo k začetno-dominantnemu mnenju/barvi.

Od česa je t.i. *čas poenotenja* odvisen?

Oris vsebine

UNIVERSITÄT SALZBURG

PULL pristop

Algoritem PULL

V vsakem koraku izberemo naključno vozlišče, ter prevzamemo njegovo mneje/barvo.

- Časovno zelo potratno – $\Omega(n)$
- Lepa matematična analiza.
- Slabo zagotovilo, da bo začetno-dominantna barva prevladala tudi na koncu.

Povezane objave:

- Nakata, Imahayashi, and Yamashita (1999),
- Hassin and Peleg (2001),
- Cooper, Elsässer, Ono, and Radzik (2013),
- Berenbrink, Giakkoupis, Kermarrec and Mallmann-Trenn (2016),
- Varun, Mallmann-Trenn and Sauerwald (2017).

UNIVERSITÄT SALZBURG

PULL pristop

Algoritem PULL

V vsakem koraku izberemo naključno vozlišče, ter prevzamemo njegovo mneje/barvo.

- Časovno zelo potratno – $\Omega(n)$
- Lepa matematična analiza.
- Slabo zagotovilo, da bo začetno-dominantna barva prevladala tudi na koncu.

Povezane objave:

- Nakata, Imahayashi, and Yamashita (1999),
- Hassin and Peleg (2001),
- Cooper, Elsässer, Ono, and Radzik (2013),
- Berenbrink, Giakkoupis, Kermarrec and Mallmann-Trenn (2016),
- Varun, Mallmann-Trenn and Sauerwald (2017).

UNIVERSITÄT SALZBURG

PULL pristop

Algoritem PULL

V vsakem koraku izberemo naključno vozlišče, ter prevzamemo njegovo mneje/barvo.

- Časovno zelo potratno – $\Omega(n)$
- Lepa matematična analiza.
- Slabo zagotovilo, da bo začetno-dominantna barva prevladala tudi na koncu.

Povezane objave:

- Nakata, Imahayashi, and Yamashita (1999),
- Hassin and Peleg (2001),
- Cooper, Elsässer, Ono, and Radzik (2013),
- Berenbrink, Giakkoupis, Kermarrec and Mallmann-Trenn (2016),
- Varun, Mallmann-Trenn and Sauerwald (2017).

UNIVERSITÄT SALZBURG

PULL pristop

Algoritem PULL

V vsakem koraku izberemo naključno vozlišče, ter prevzamemo njegovo mneje/barvo.

- Časovno zelo potratno – $\Omega(n)$
- Lepa matematična analiza.
- Slabo zagotovilo, da bo začetno-dominantna barva prevladala tudi na koncu.

Povezane objave:

- Nakata, Imahayashi, and Yamashita (1999),
- Hassin and Peleg (2001),
- Cooper, Elsässer, Ono, and Radzik (2013),
- Berenbrink, Giakkoupis, Kermarrec and Mallmann-Trenn (2016),
- Varun, Mallmann-Trenn and Sauerwald (2017).

UNIVERSITÄT SALZBURG

PULL pristop

Algoritem PULL

V vsakem koraku izberemo naključno vozlišče, ter prevzamemo njegovo mneje/barvo.

- Časovno zelo potratno – $\Omega(n)$
- Lepa matematična analiza.
- Slabo zagotovilo, da bo začetno-dominantna barva prevladala tudi na koncu.

Povezane objave:

- Nakata, Imahayashi, and Yamashita (1999),
- Hassin and Peleg (2001),
- Cooper, Elsässer, Ono, and Radzik (2013),
- Berenbrink, Giakkoupis, Kermarrec and Mallmann-Trenn (2016),
- Varun, Mallmann-Trenn and Sauerwald (2017).

UNIVERSITÄT SALZBURG

PULL pristop

Algoritem PULL

V vsakem koraku izberemo naključno vozlišče, ter prevzamemo njegovo mneje/barvo.

- Časovno zelo potratno – $\Omega(n)$
- Lepa matematična analiza.
- Slabo zagotovilo, da bo začetno-dominantna barva prevladala tudi na koncu.

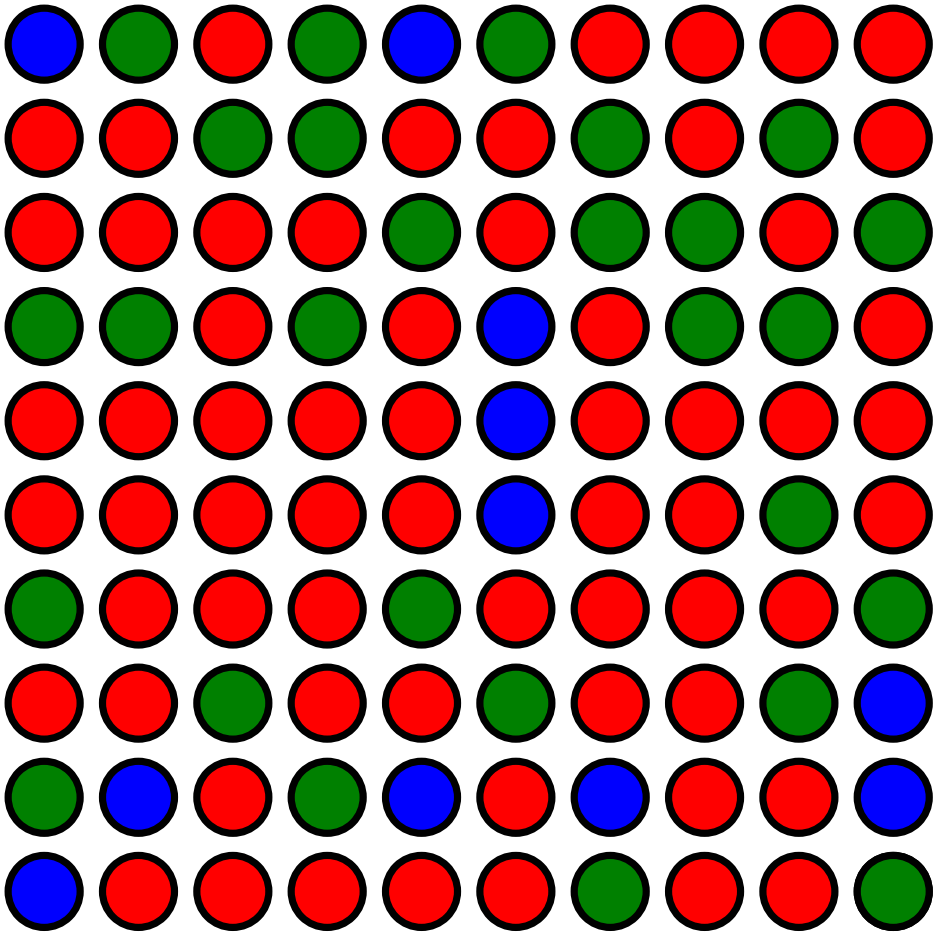
Povezane objave:

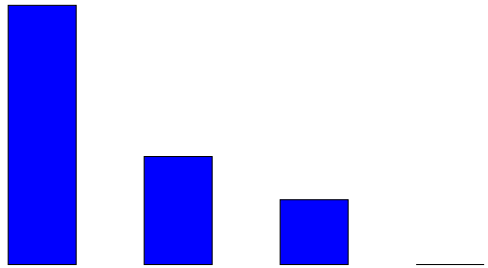
- Nakata, Imahayashi, and Yamashita (1999),
- Hassin and Peleg (2001),
- Cooper, Elsässer, Ono, and Radzik (2013),
- Berenbrink, Giakkoupis, Kermarrec and Mallmann-Trenn (2016),
- Varun, Mallmann-Trenn and Sauerwald (2017).

UNIVERSITÄT SALZBURG

PULL pristop (0)

Kar slišiš, verjameš!



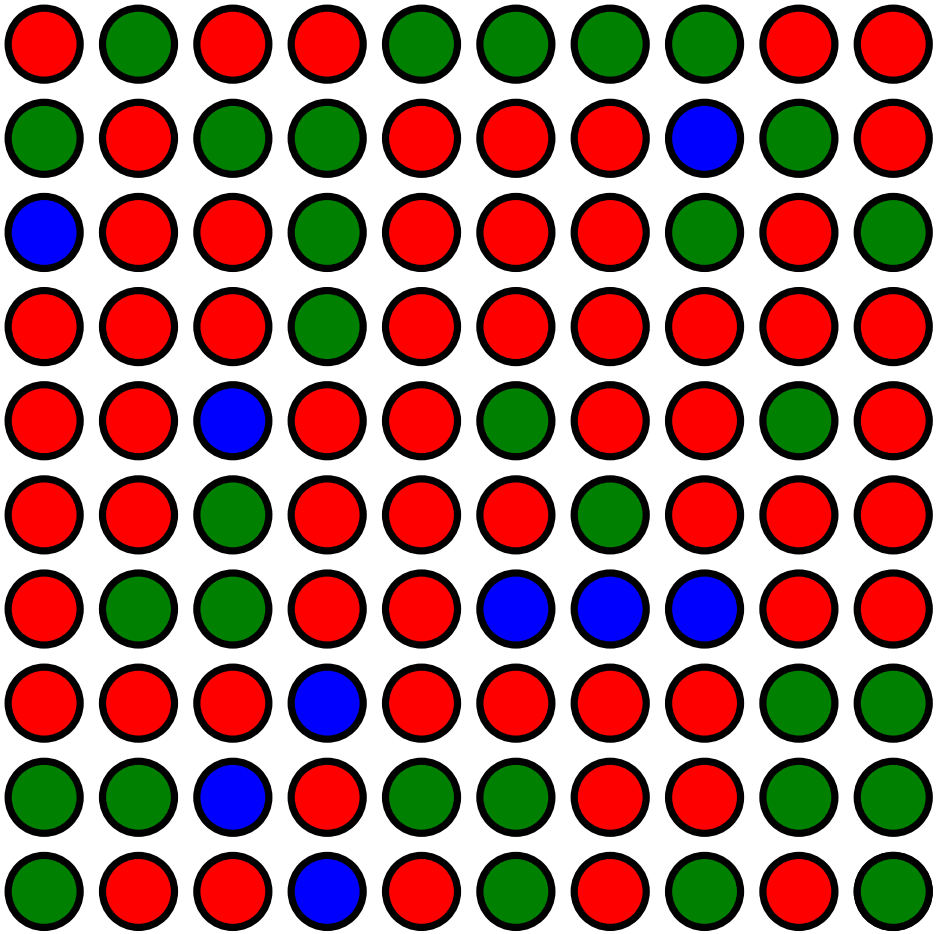


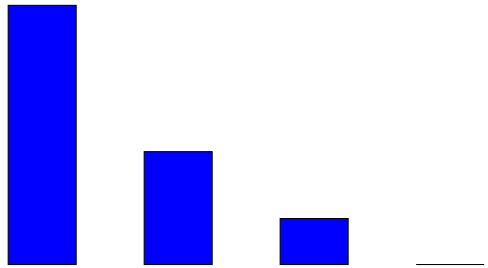
Začetna distribucija:
$(\mathtt{r}, \mathtt{g}, \mathtt{b}) \sim$
$(0.6, 0.25, 0.15)$

UNIVERSITÄT SALZBURG

PULL pristop (1)

Kar slišiš, verjameš!



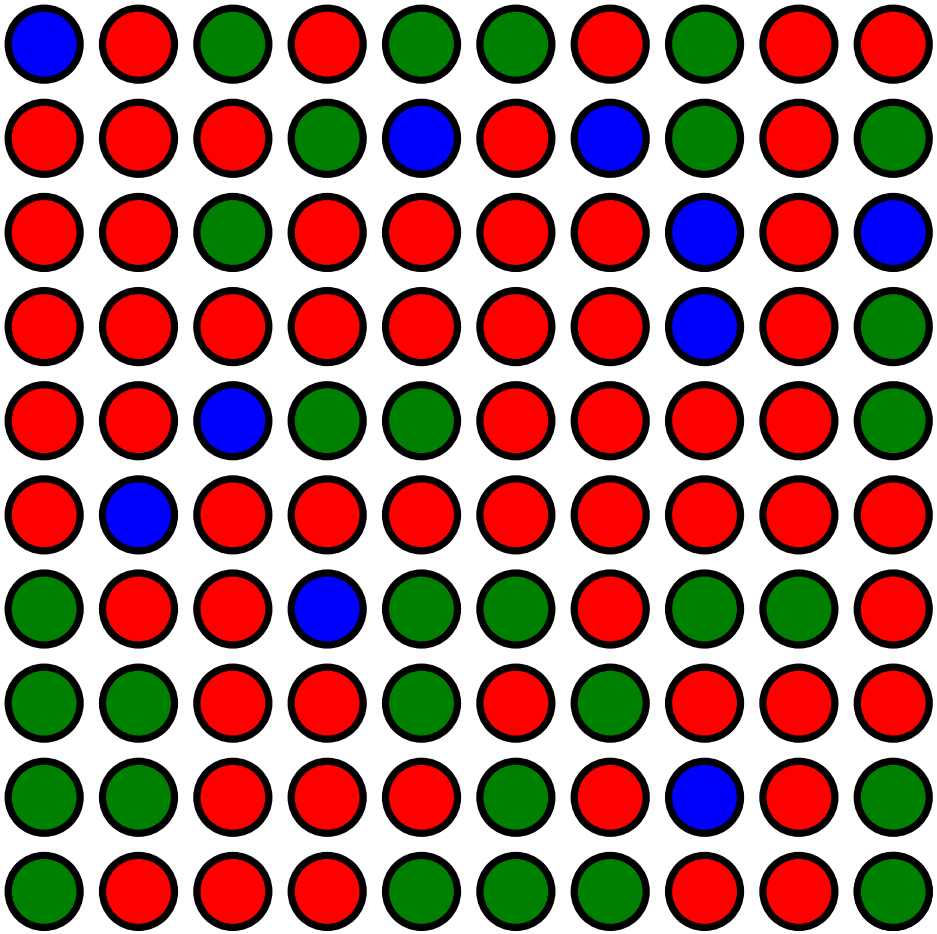


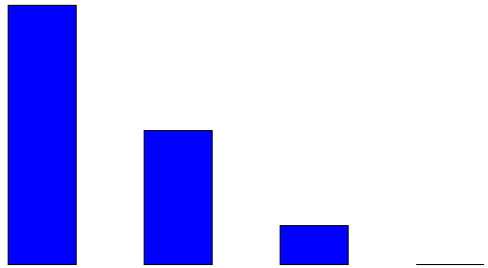
Začetna distribucija:
$(\mathtt{r}, \mathtt{g}, \mathtt{b}) \sim$
$(0.6, 0.25, 0.15)$

UNIVERSITÄT SALZBURG

PULL pristop (2)

Kar slišiš, verjameš!



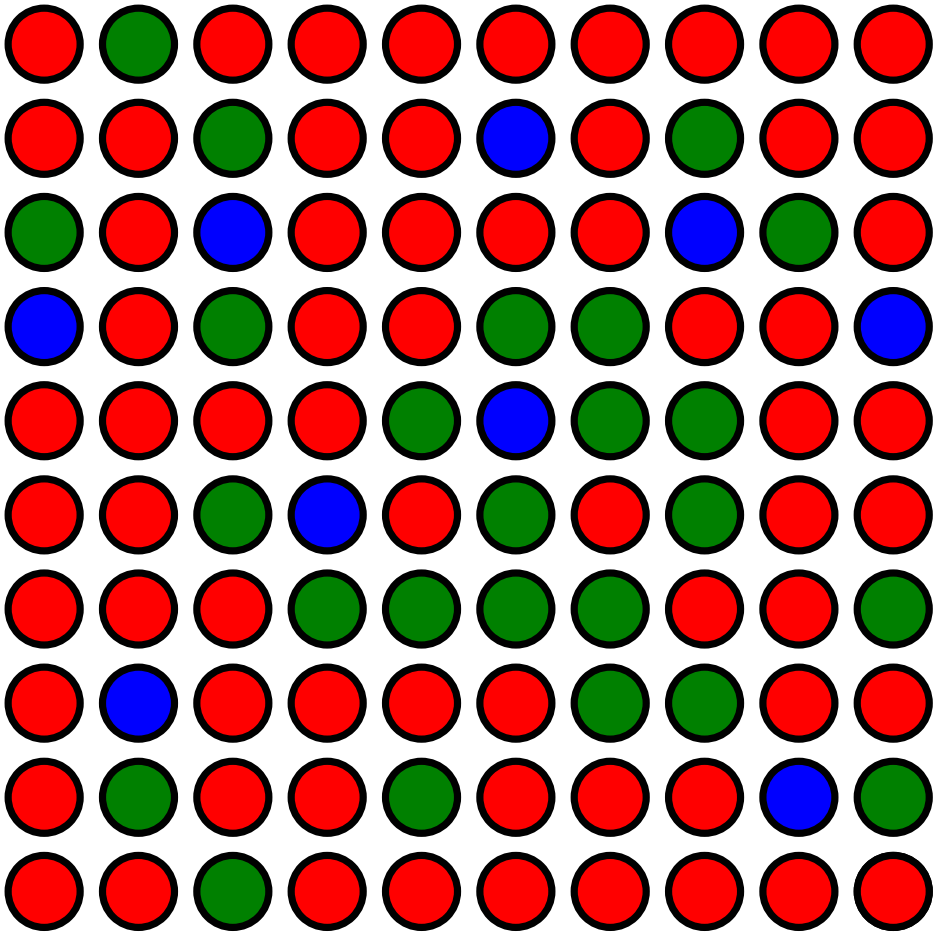


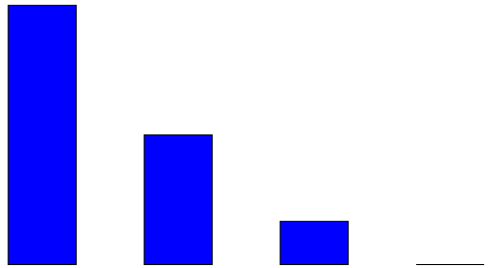
Začetna distribucija:
$(\mathtt{r}, \mathtt{g}, \mathtt{b}) \sim$
$(0.6, 0.25, 0.15)$

UNIVERSITÄT SALZBURG

PULL pristop (3)

Kar slišiš, verjameš!



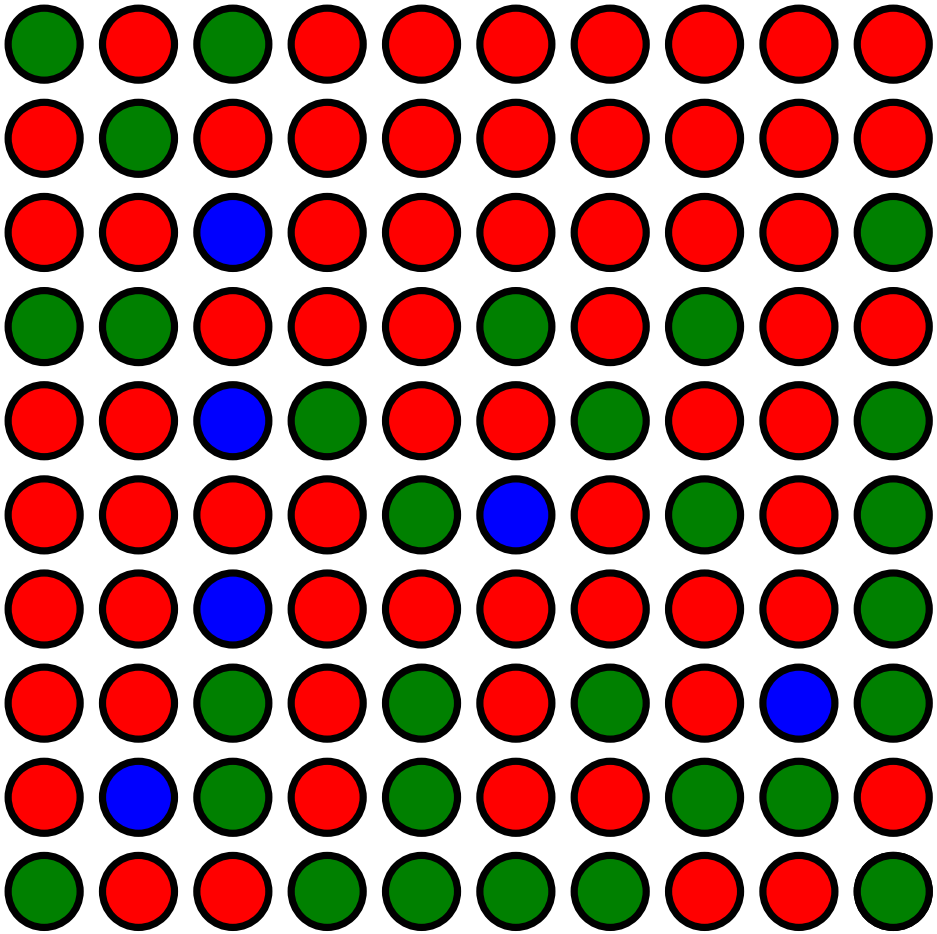


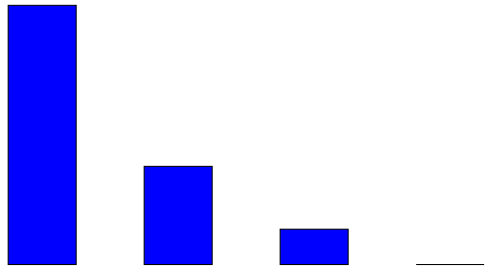
Začetna distribucija:
$(\mathtt{r}, \mathtt{g}, \mathtt{b}) \sim$
$(0.6, 0.25, 0.15)$

UNIVERSITÄT SALZBURG

PULL pristop (4)

Kar slišiš, verjameš!

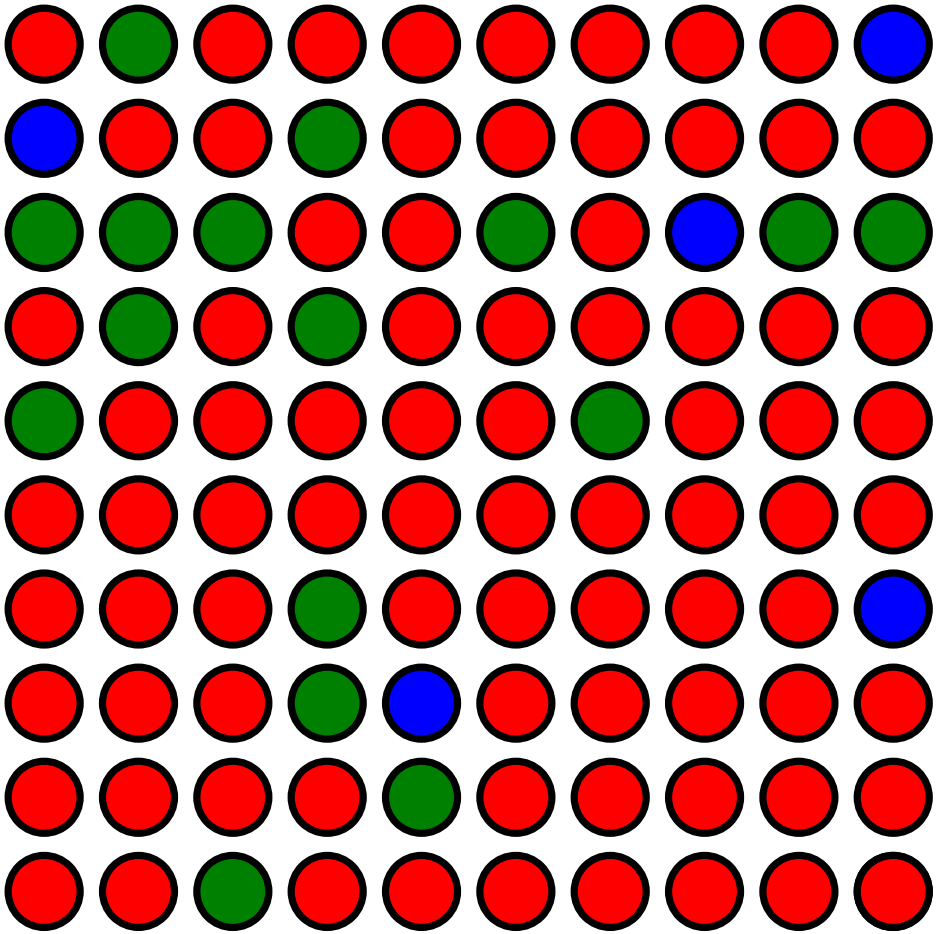


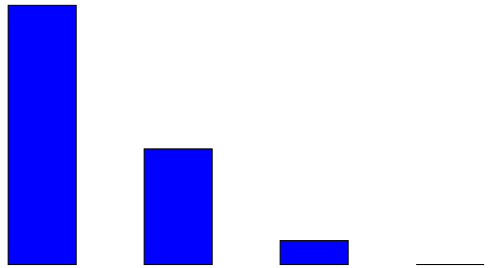


Začetna distribucija:
$(\mathtt{r}, \mathtt{g}, \mathtt{b}) \sim$
$(0.6, 0.25, 0.15)$

PULL pristop (5)

Kar slišiš, verjameš!

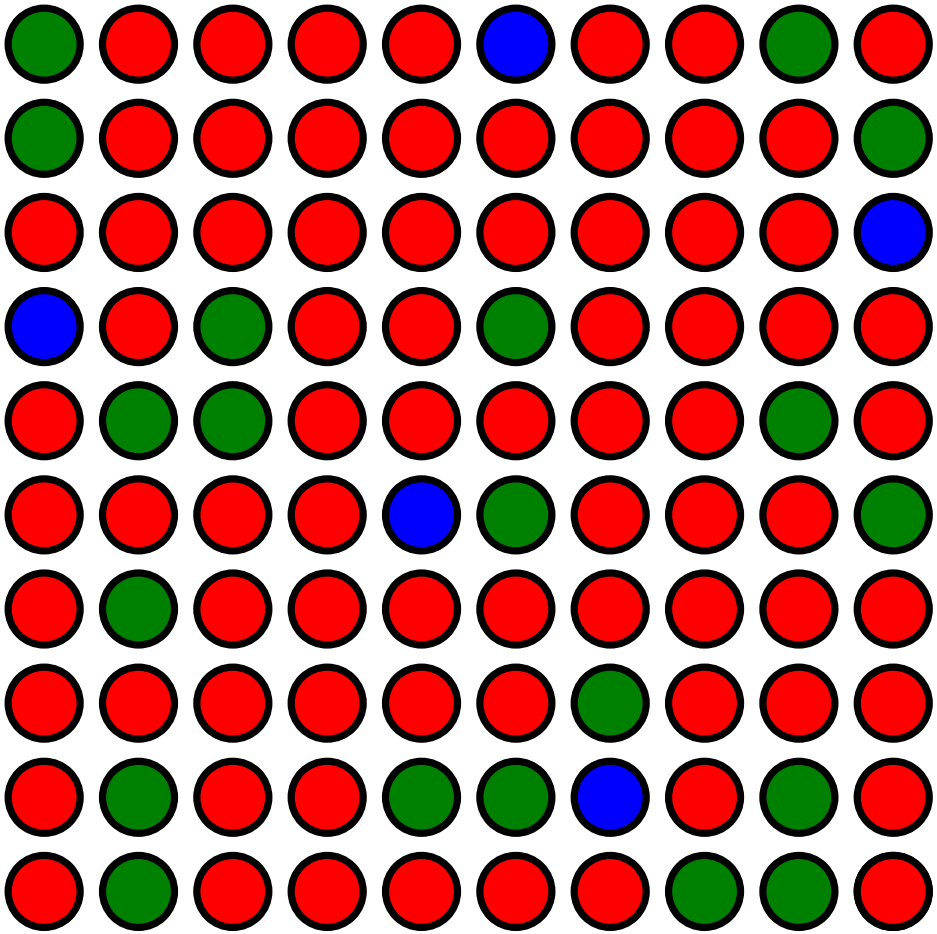


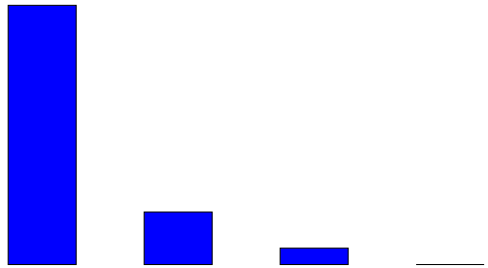


Začetna distribucija:
$(\mathtt{r}, \mathtt{g}, \mathtt{b}) \sim$
$(0.6, 0.25, 0.15)$

PULL pristop (6)

Kar slišiš, verjameš!

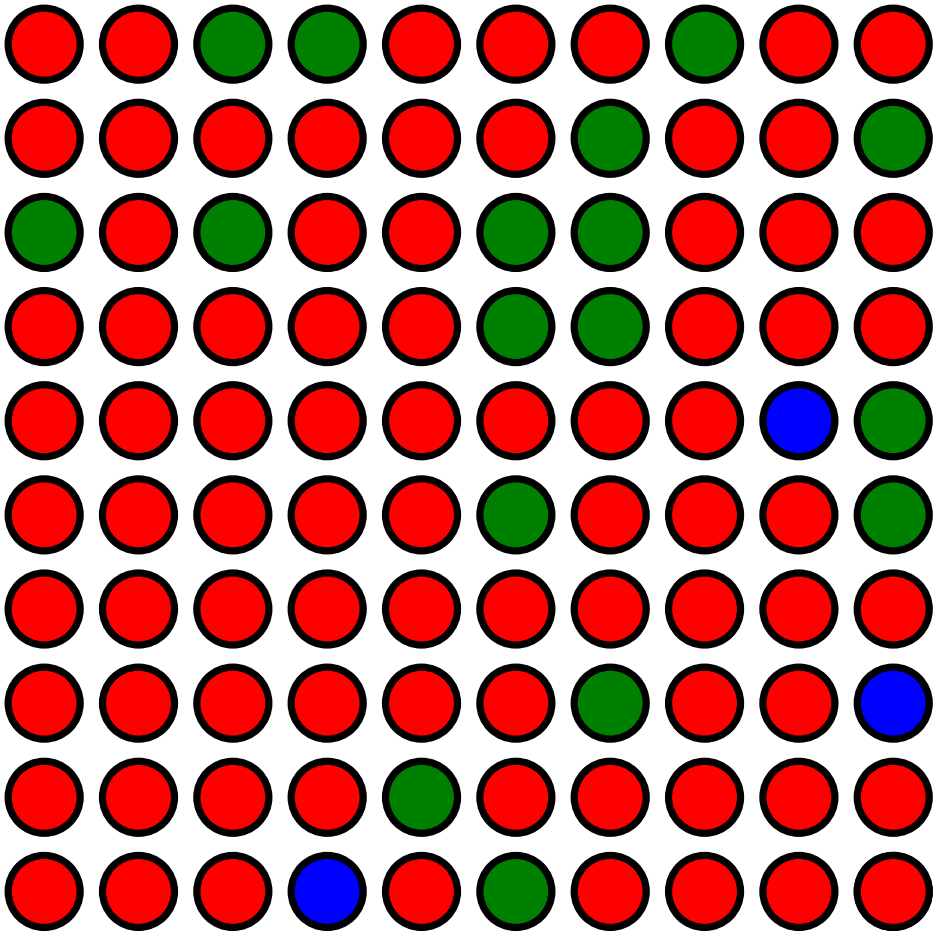


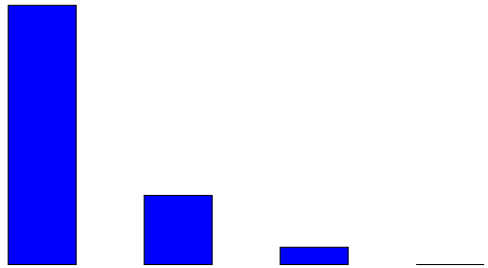


Začetna distribucija:
$(\mathtt{r}, \mathtt{g}, \mathtt{b}) \sim$
$(0.6, 0.25, 0.15)$

PULL pristop (7)

Kar slišiš, verjameš!

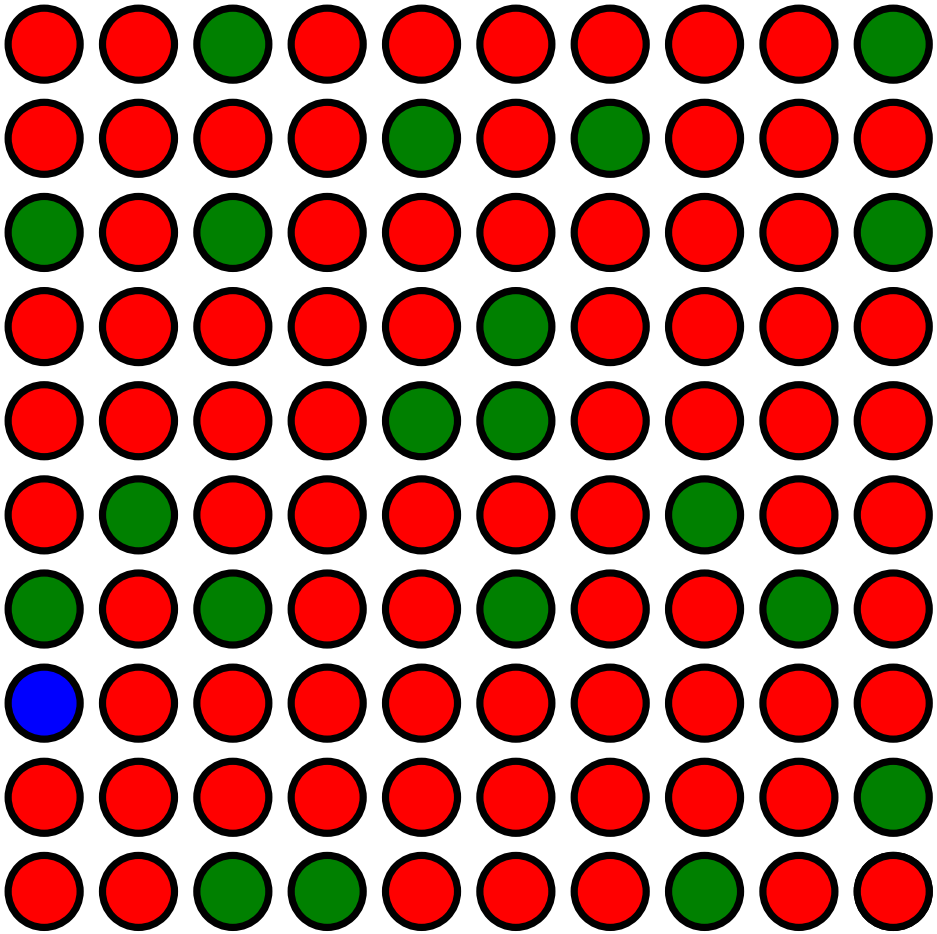


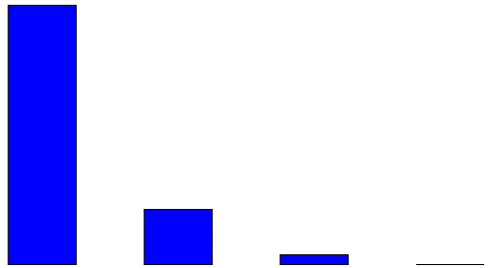


Začetna distribucija:
$(\mathtt{r}, \mathtt{g}, \mathtt{b}) \sim$
$(0.6, 0.25, 0.15)$

PULL pristop (8)

Kar slišiš, verjameš!



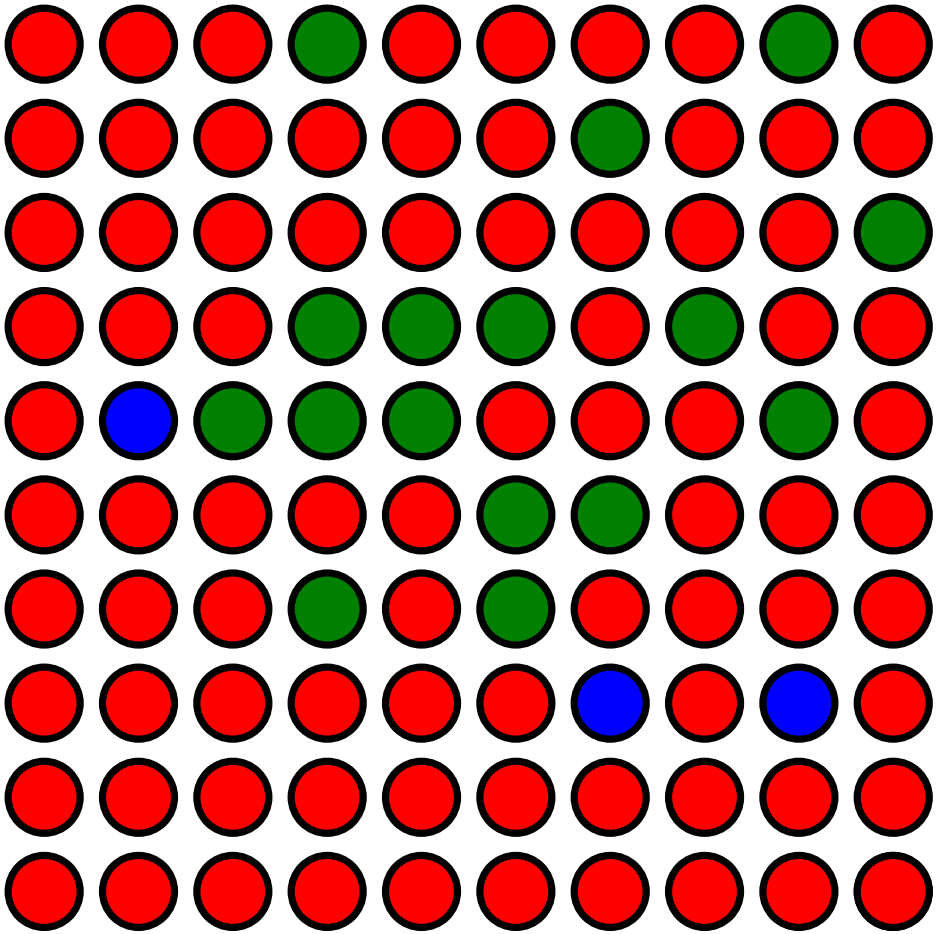


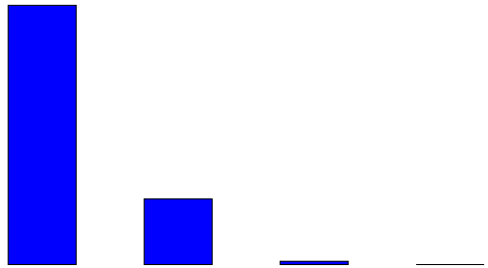
Začetna distribucija:
$(\mathtt{r}, \mathtt{g}, \mathtt{b}) \sim$
$(0.6, 0.25, 0.15)$

UNIVERSITÄT SALZBURG

PULL pristop (9)

Kar slišiš, verjameš!





Začetna distribucija:
$(\mathtt{r}, \mathtt{g}, \mathtt{b}) \sim$
$(0.6, 0.25, 0.15)$

UNIVERSITÄT SALZBURG

Oris vsebine

UNIVERSITÄT SALZBURG

Pristop 2-vzorčenja

"If you hear it twice, it must be true"

Algoritem 2-vzorčenja

Naključno izberi dve vozlišči. Če se njuni barvi ujemata, le-to barvo privzemi.

- Imamo garancijo da ob nežnih začetnih pogojih (odklon $\sim \sqrt{n \log n}$) na koncu prevlada mnenje z največjim začetnim deležem.
- Vzorčenje bo uspešno samo v majhnem deležu vozlišč – potrebujemo $\Omega(k)$ časovnih enot.

UNIVERSITÄT SALZBURG

Pristop 2-vzorčenja

"If you hear it twice, it must be true"

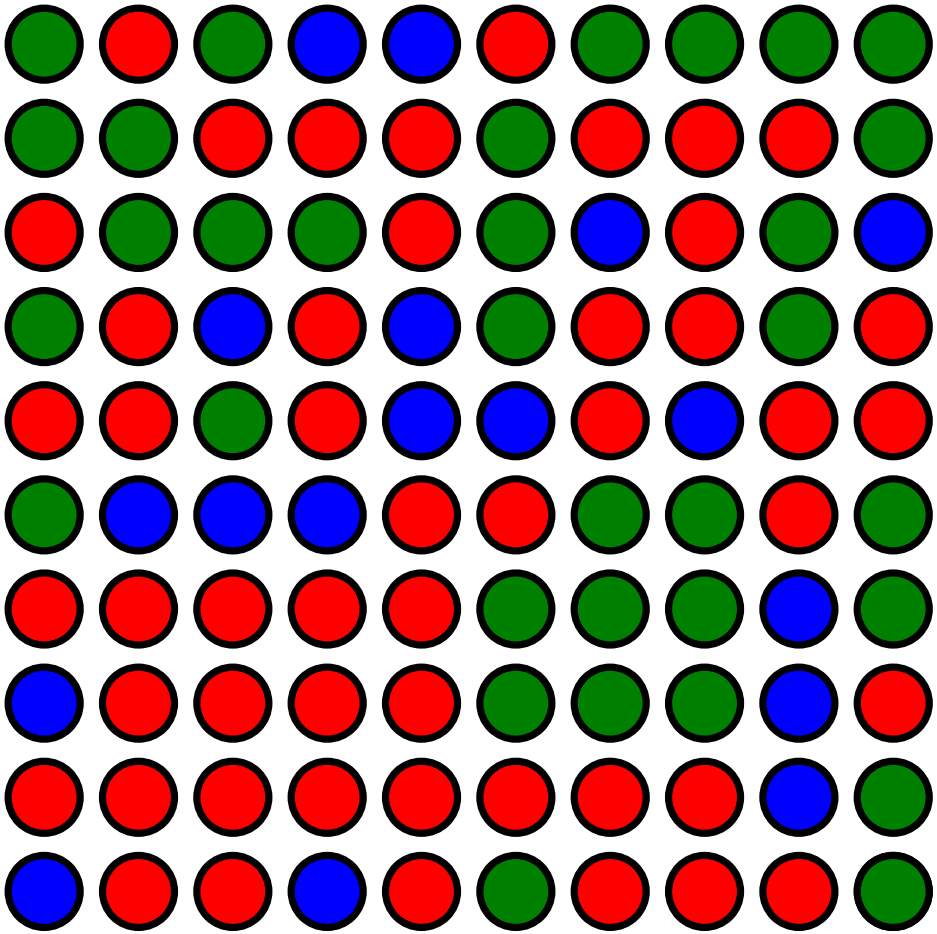
Algoritem 2-vzorčenja

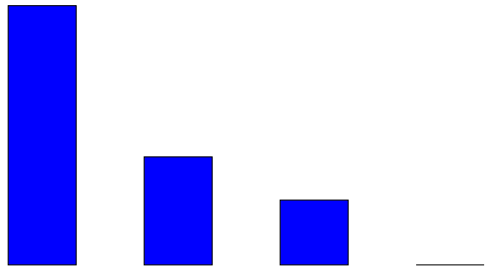
Naključno izberi dve vozlišči. Če se njuni barvi ujemata, le-to barvo privzemi.

- Imamo garancijo da ob nežnih začetnih pogojih (odklon $\sim \sqrt{n \log n}$) na koncu prevlada mnenje z največjim začetnim deležem.
- Vzorčenje bo uspešno samo v majhnem deležu vozlišč – potrebujemo $\Omega(k)$ časovnih enot.

Pristop 2-vzorčenja (0)

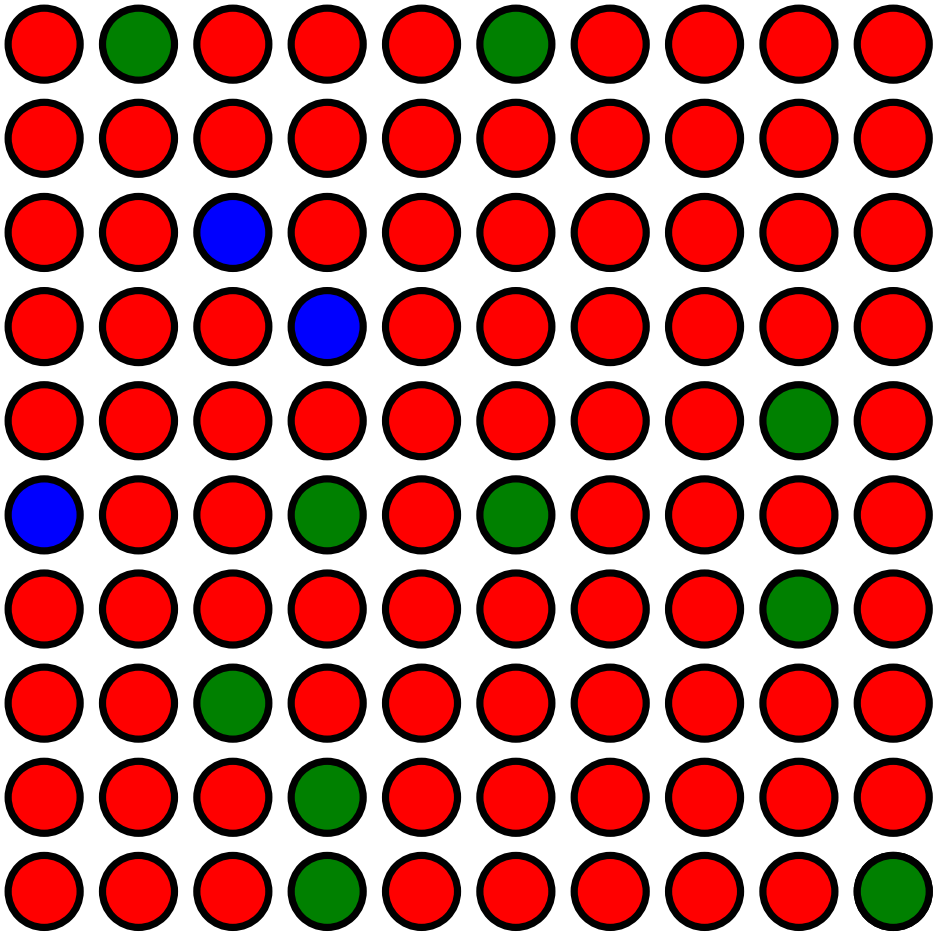
Primer 1 $(\mathtt{r},\mathtt{g},\mathtt{b}) \sim (0.6, 0.25, 0.15)$

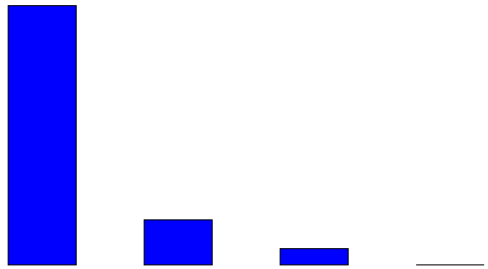




Pristop 2-vzorčenja (1)

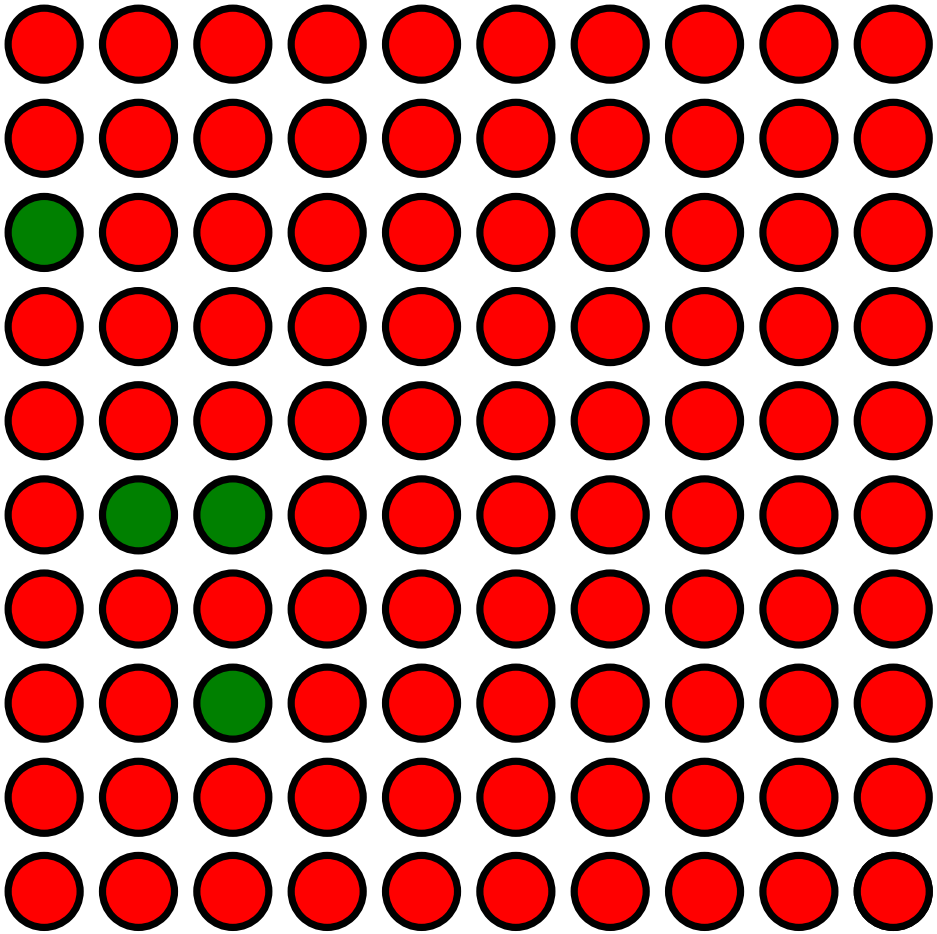
Primer 1

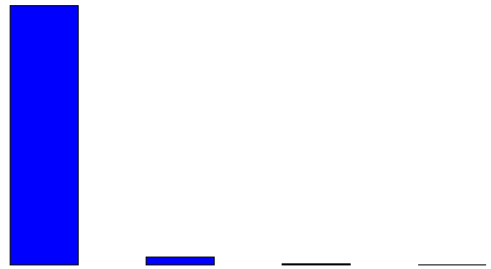




Pristop 2-vzorčenja (2)

Primer 1

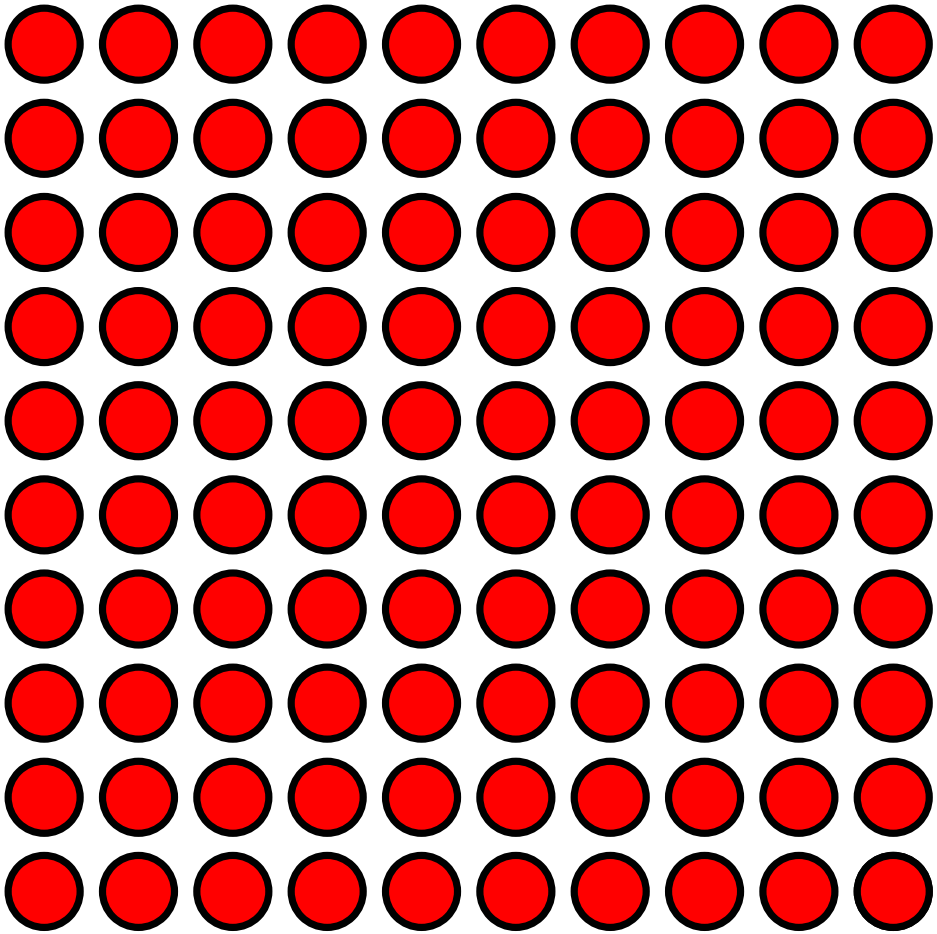


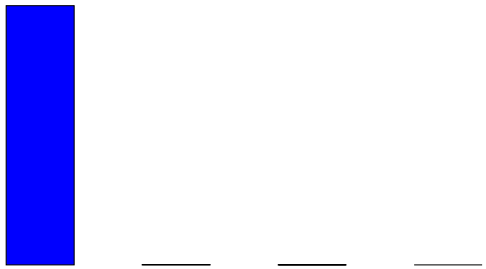


UNIVERSITÄT SALZBURG

Pristop 2-vzorčenja (3)

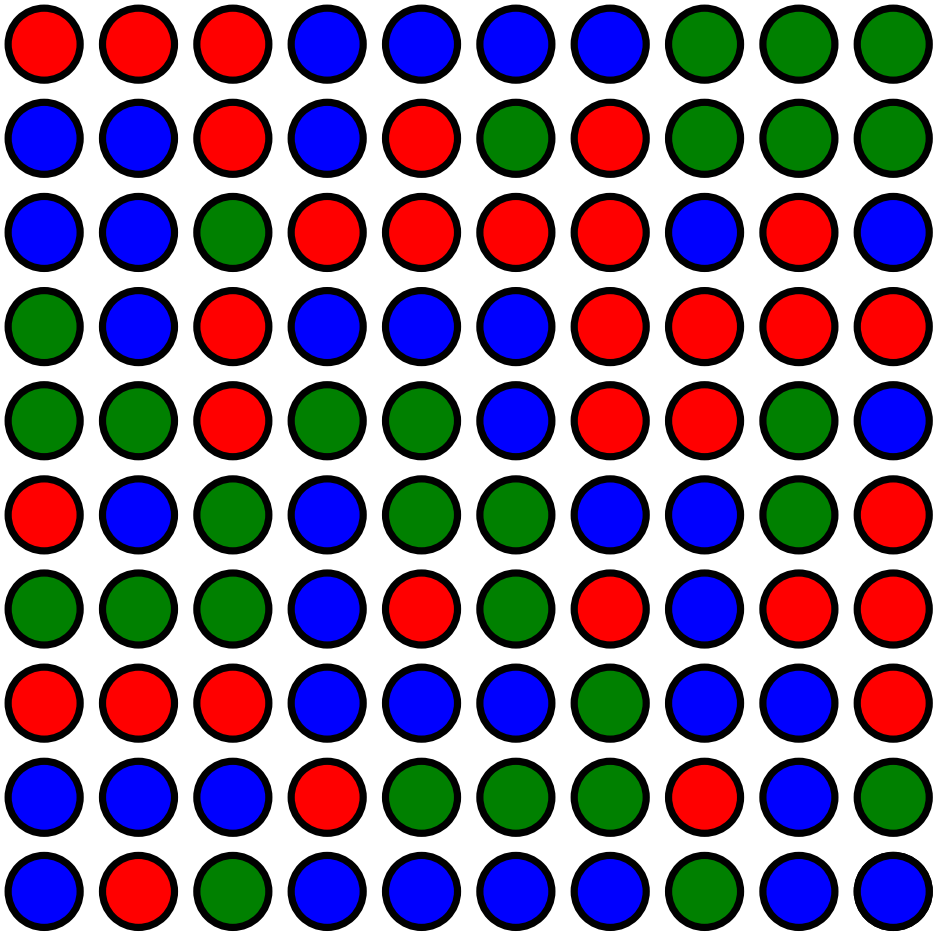
Primer 1

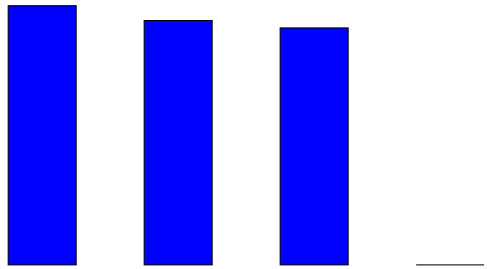




Pristop 2-vzorčenja (0)

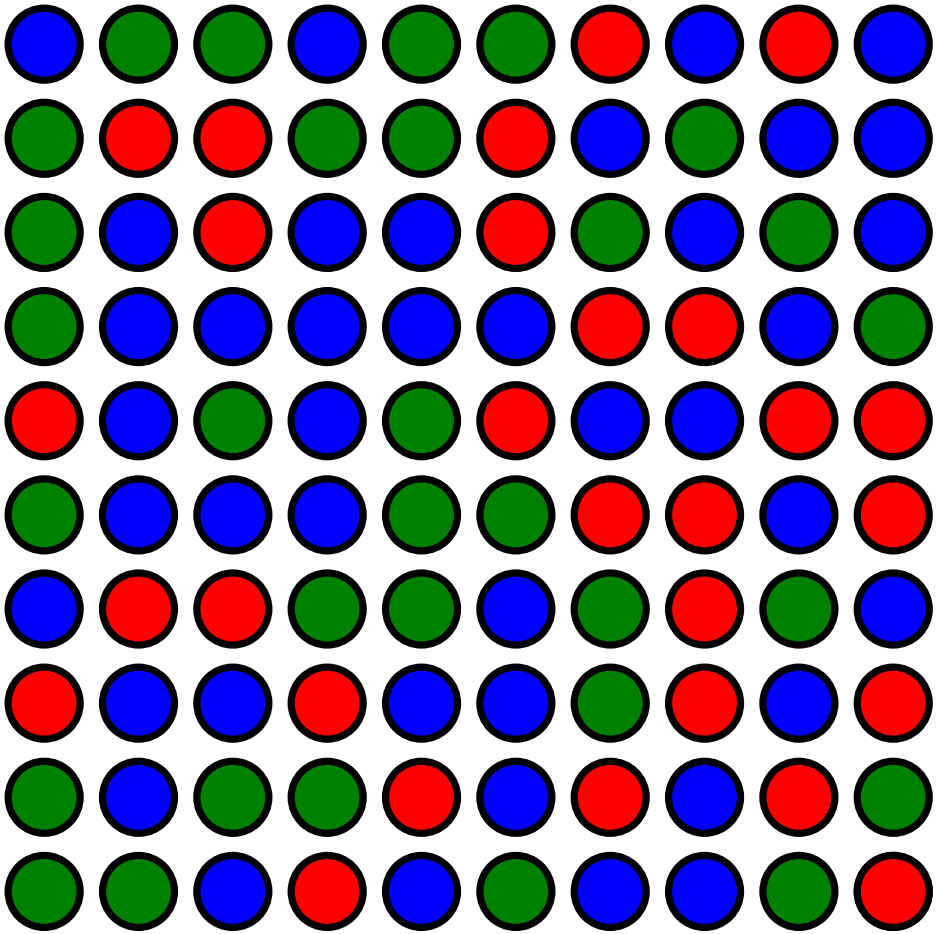
Primer 2: distribucija $(\mathtt{r}, \mathtt{g}, \mathtt{b}) \sim (0.35, 0.33, 0.32)$

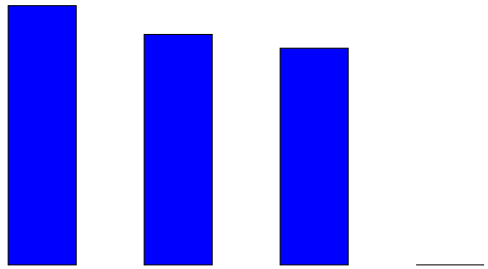




Pristop 2-vzorčenja (1)

Primer 2: distribucija $(\mathtt{r},\mathtt{g},\mathtt{b}) \sim (0.37, 0.33, 0.30)$

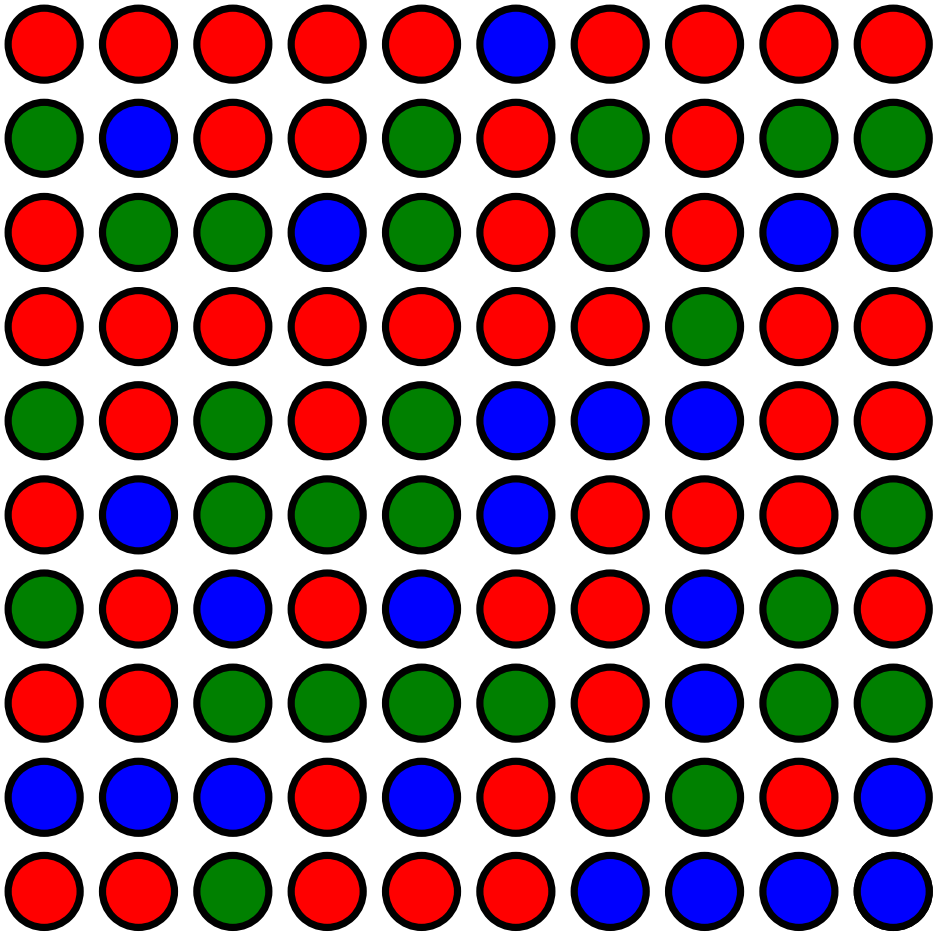


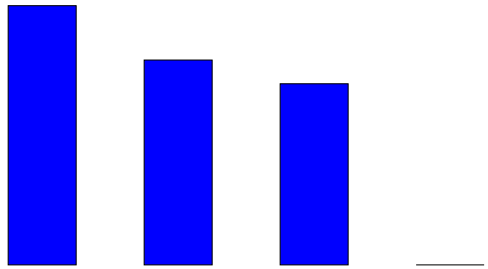


UNIVERSITÄT SALZBURG

Pristop 2-vzorčenja (2)

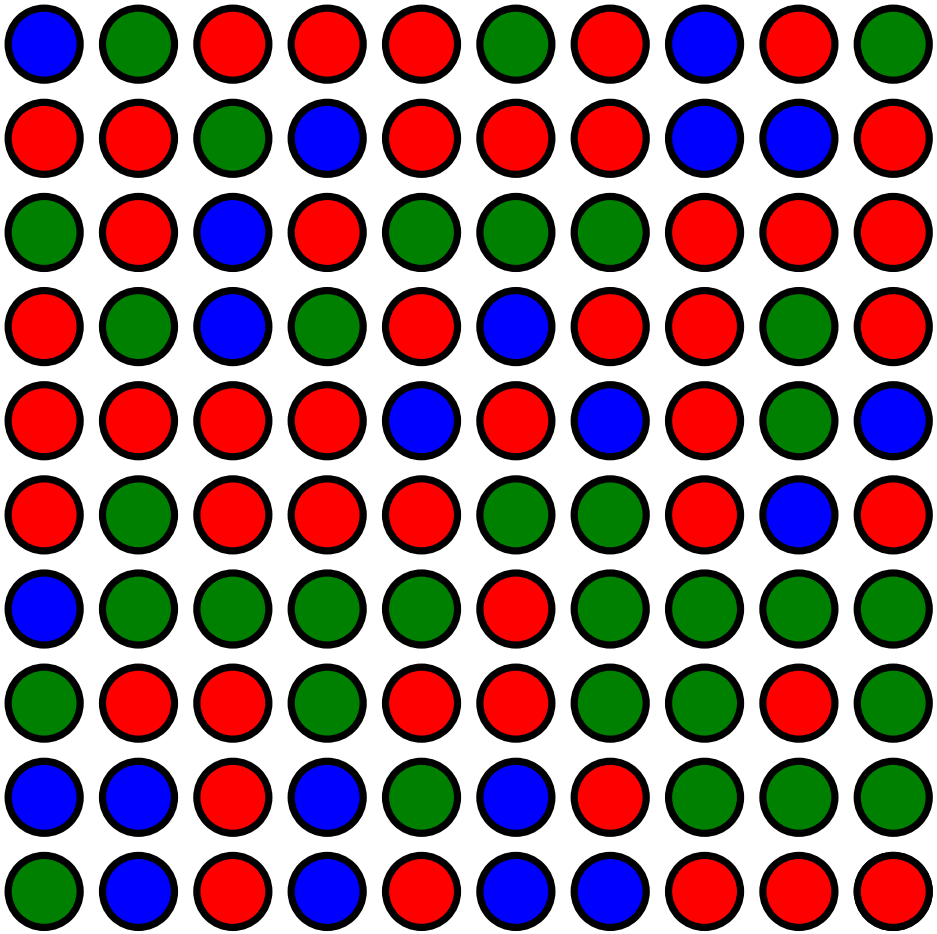
Primer 2: distribucija $(\mathtt{r}, \mathtt{g}, \mathtt{b}) \sim (0.40, 0.32, 0.28)$

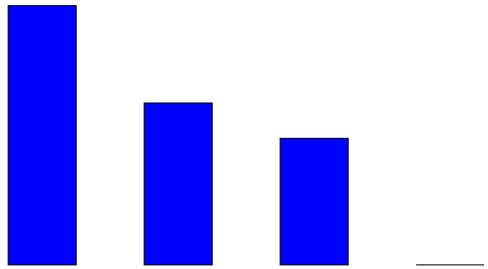




Pristop 2-vzorčenja (3)

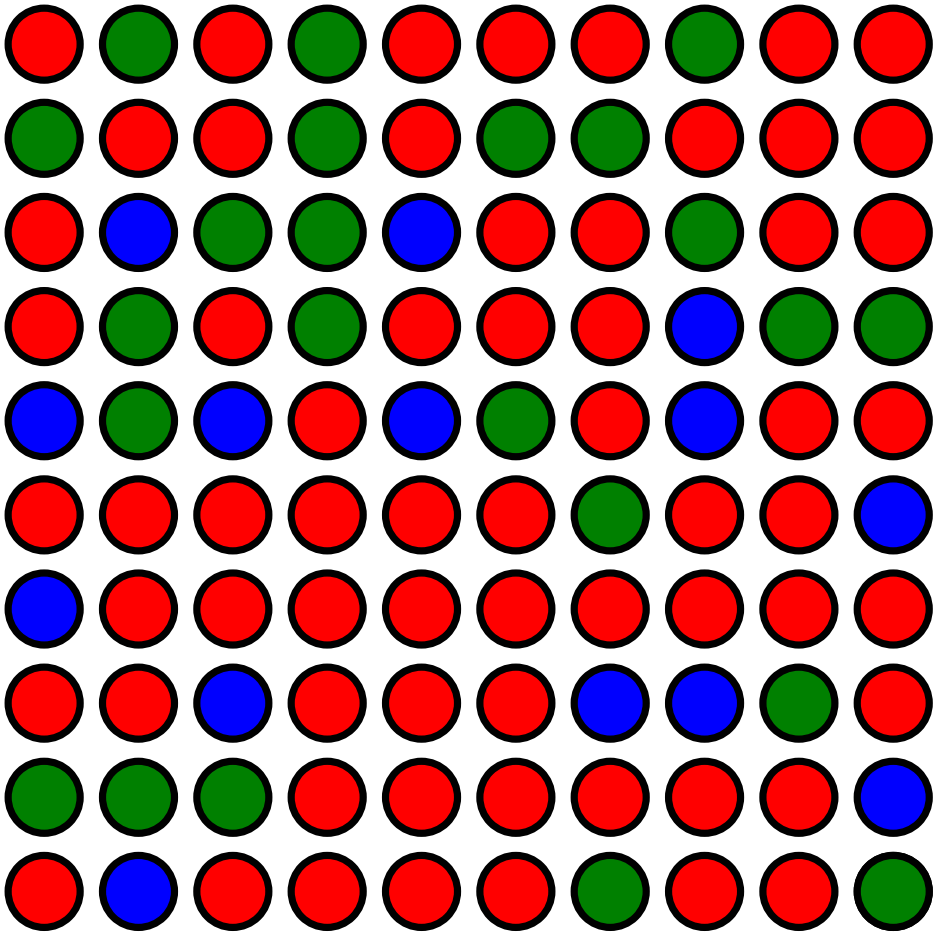
Primer 2: distribucija $(\mathtt{r}, \mathtt{g}, \mathtt{b}) \sim (0.47, 0.30, 0.23)$

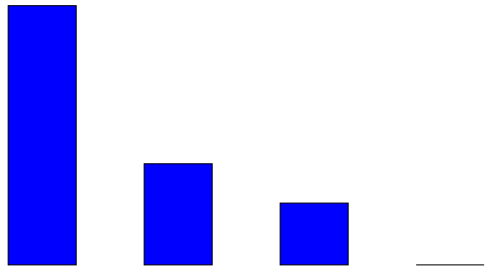




Pristop 2-vzorčenja (4)

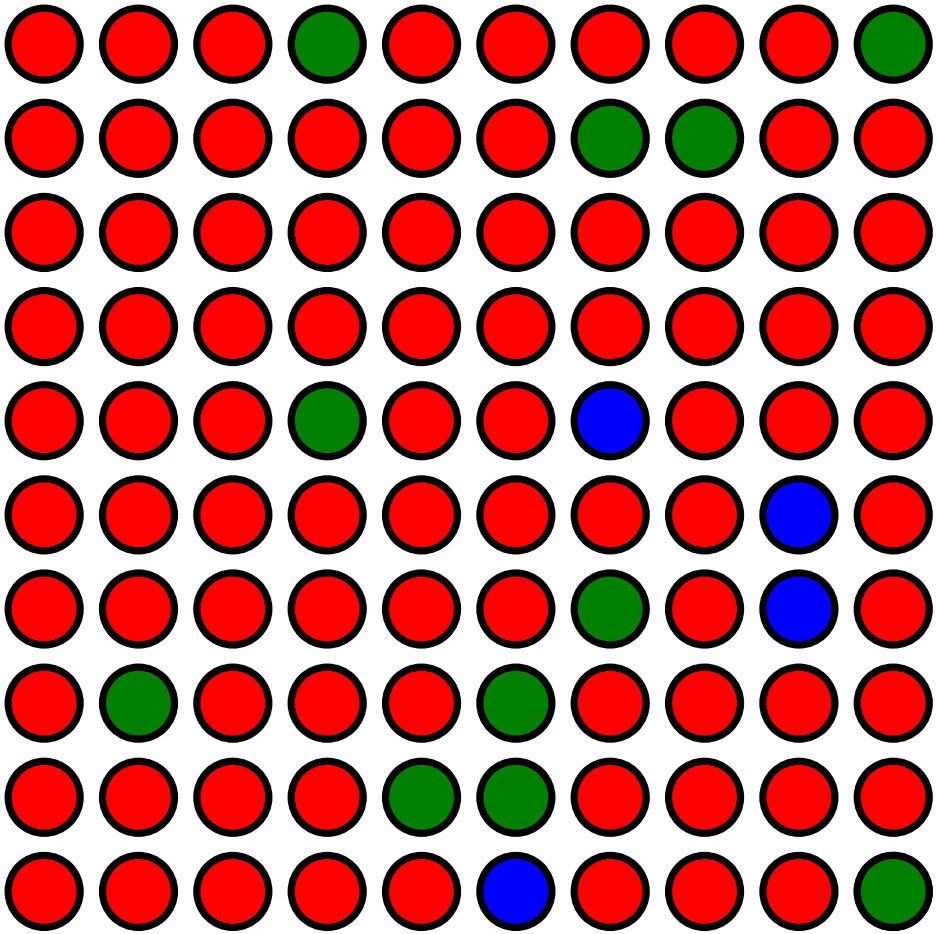
Primer 2: distribucija $(\mathtt{r}, \mathtt{g}, \mathtt{b}) \sim (0.61, 0.24, 0.15)$

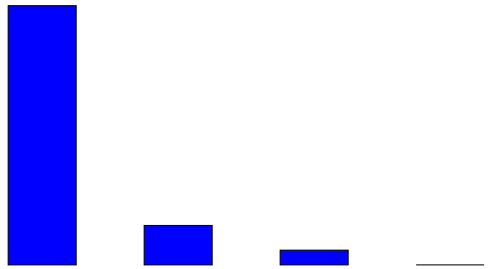




Pristop 2-vzorčenja (5)

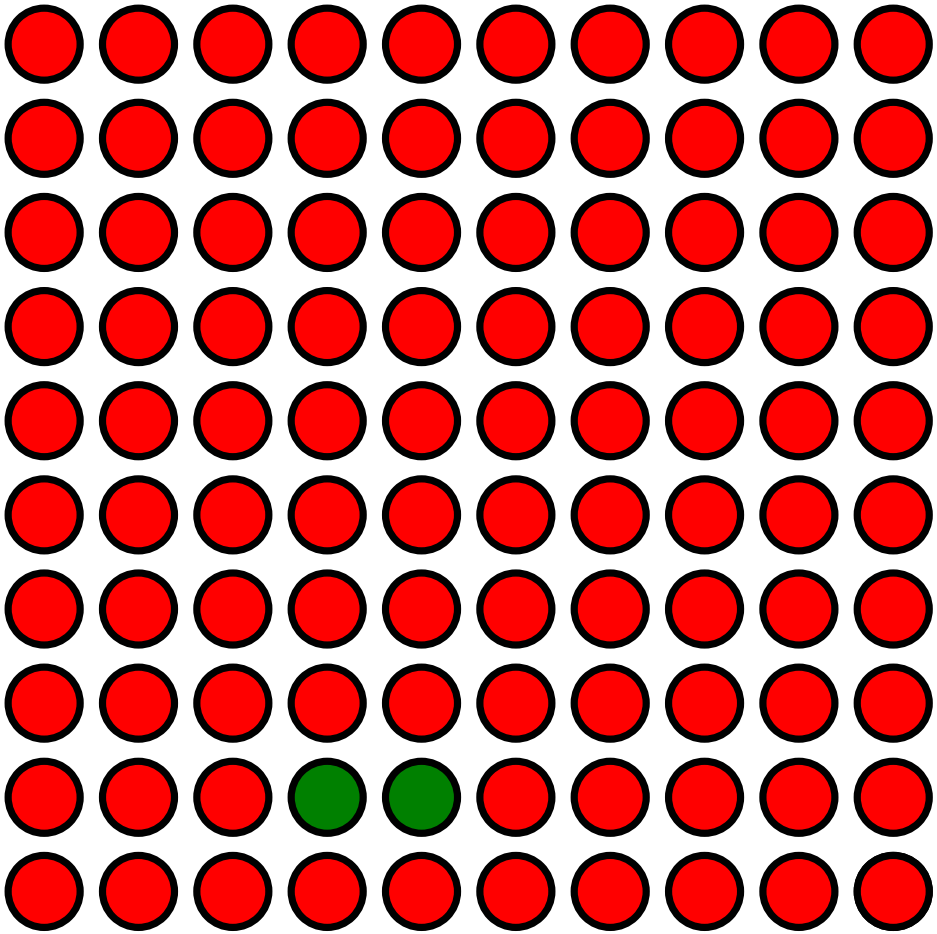
Primer 2: distribucija $(\mathtt{r}, \mathtt{g}, \mathtt{b}) \sim (0.83, 0.13, 0.05)$

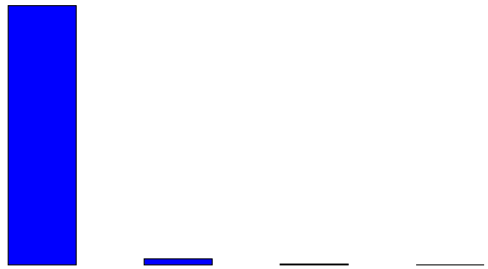




Pristop 2-vzorčenja (6)

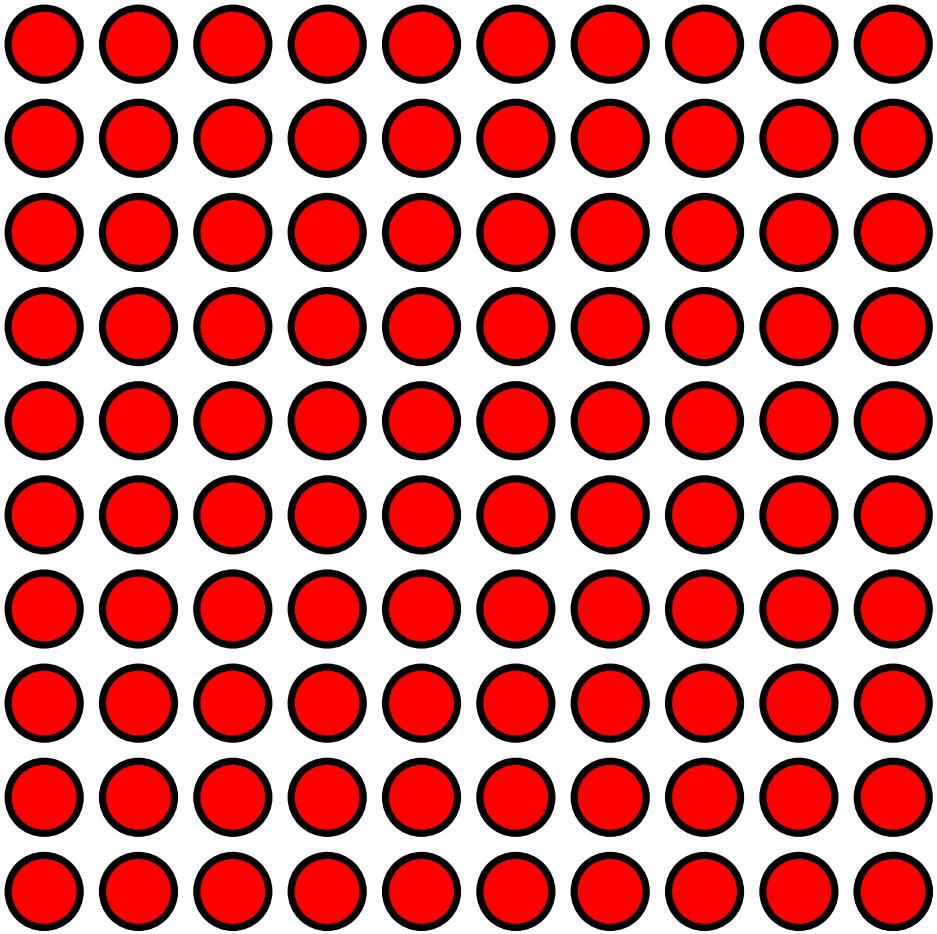
Primer 2: distribucija $(\mathtt{r}, \mathtt{g}, \mathtt{b}) \sim (0.98, 0.02, 0)$

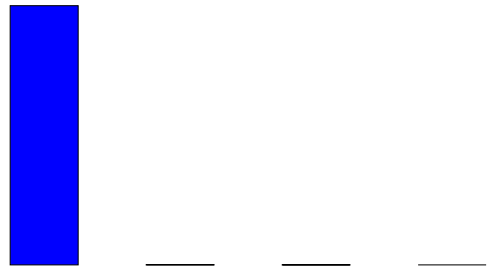




Pristop 2-vzorčenja (7)

Primer 2





UNIVERSITÄT SALZBURG

Pristop 2-vzorčenja

Povezane objave

- Hassin-Peleg, Nakata et al.: general graphs
- Cruise-Ghanesh: complete graph, traditional model
- Doerr et al.: complete graph, median rule
- Mossel et al.: expander, majority consensus
- Abdullah-Draief: random graph, five-sample voting
- Becchetti et al. or Angluin et al.: 3-majority protocol, 3-state protocol
- Two-Party Two-Sample Voting:
  - Cooper, Elsässer and Radzik (2014)
  - Cooper, Elsässer, Radzik, Rivera and Shiraga (2015)

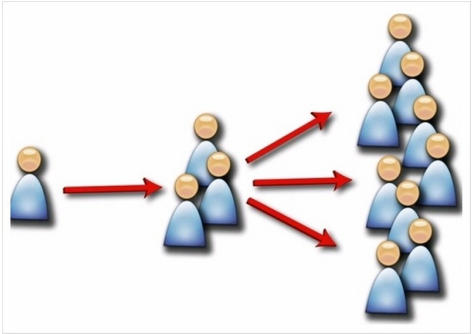
Algoritem lahko pospešimo s pomočjo t.i. ***gossip* algoritmov**.

UNIVERSITÄT SALZBURG

Oris vsebine

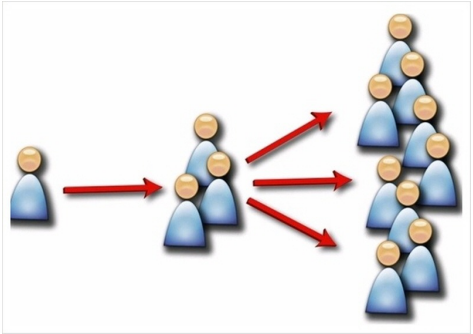
UNIVERSITÄT SALZBURG

Gossip algoritmi
Osnovna ideja



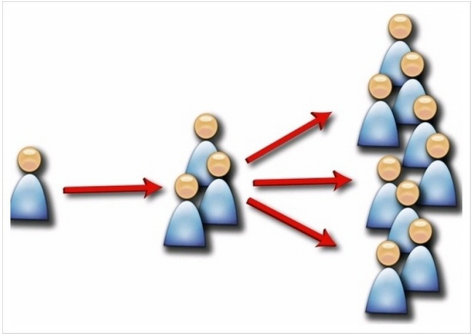
- Cilj: razširit informacijo med vsa vozlišča.
- Vsako informirano vozlišče v vsaki čas. enoti pošlje svojo informacijo nekaj naključnim sosedom; ti bodo v naslednjem krogu informirani.
- Odličen čas prenosa informacije, can be very fast and *message-efficient*
- V modelu naključnih telefonskih klicev z n objekti popolno informiranje traja $O(\log n)$.

Gossip algoritmi
Osnovna ideja



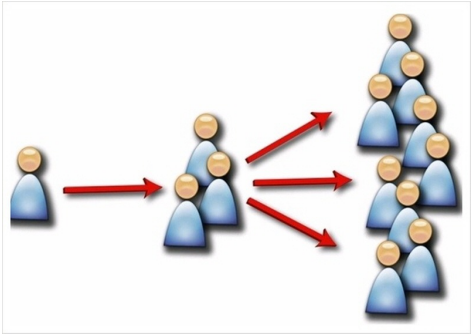
- Cilj: razširit informacijo med vsa vozlišča.
- Vsako informirano vozlišče v vsaki čas. enoti pošlje svojo informacijo nekaj naključnim sosedom; ti bodo v naslednjem krogu informirani.
- Odličen čas prenosa informacije, can be very fast and *message-efficient*
- V modelu naključnih telefonskih klicev z n objekti popolno informiranje traja $O(\log n)$.

Gossip algoritmi
Osnovna ideja



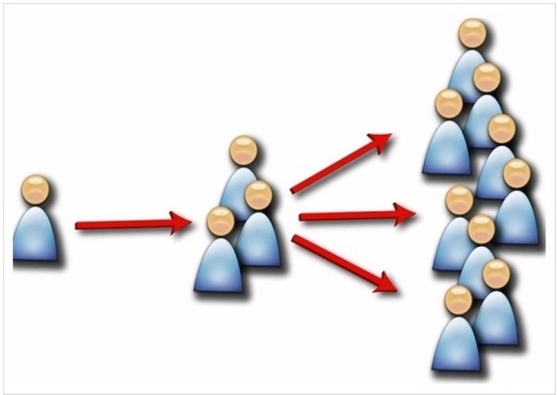
- Cilj: razširit informacijo med vsa vozlišča.
- Vsako informirano vozlišče v vsaki čas. enoti pošlje svojo informacijo nekaj naključnim sosedom; ti bodo v naslednjem krogu informirani.
- Odličen čas prenosa informacije, can be very fast and *message-efficient*
- V modelu naključnih telefonskih klicev z n objekti popolno informiranje traja $O(\log n)$.

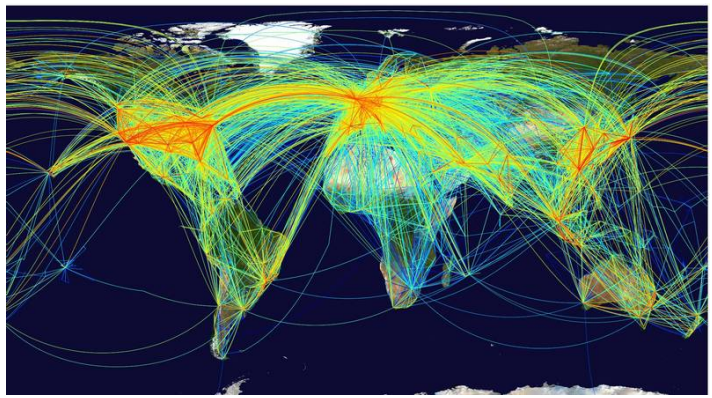
Gossip algoritmi
Osnovna ideja



- Cilj: razširit informacijo med vsa vozlišča.
- Vsako informirano vozlišče v vsaki čas. enoti pošlje svojo informacijo nekaj naključnim sosedom; ti bodo v naslednjem krogu informirani.
- Odličen čas prenosa informacije, can be very fast and *message-efficient*
- V modelu naključnih telefonskih klicev z n objekti popolno informiranje traja $O(\log n)$.

Gossip algoritmi
Primeri

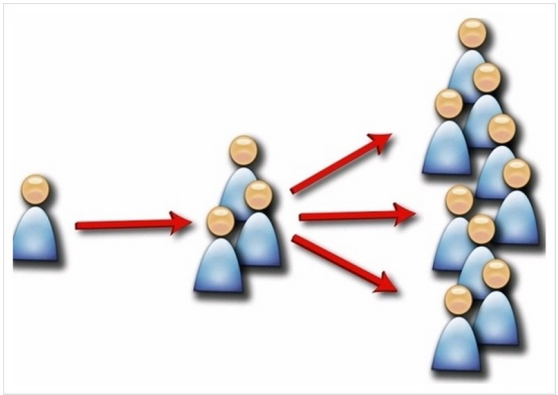


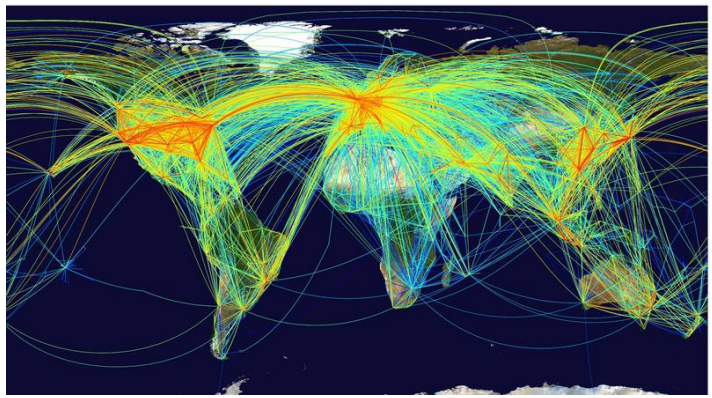


- širjenje virusov/epidemij/govoric
- PULL oz. PUSH algoritmi,
- oddajanje zadnjega bloka od uspešnega rudarja, do vseh drugih

UNIVERSITÄT SALZBURG

Gossip algoritmi
Primeri

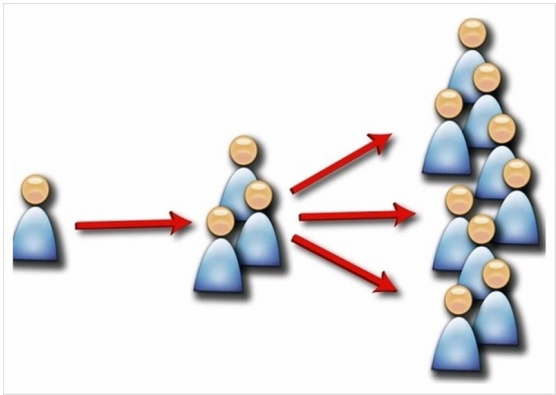


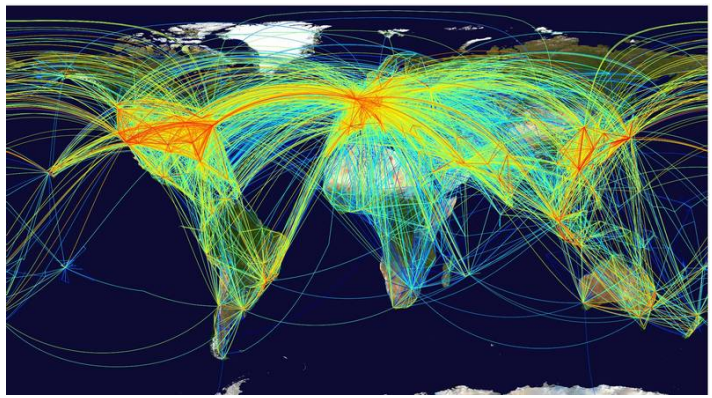


- širjenje virusov/epidemij/govoric
- PULL oz. PUSH algoritmi,
- oddajanje zadnjega bloka od uspešnega rudarja, do vseh drugih

UNIVERSITÄT SALZBURG

Gossip algoritmi
Primeri





- širjenje virusov/epidemij/govoric
- PULL oz. PUSH algoritmi,
- oddajanje zadnjega bloka od uspešnega rudarja, do vseh drugih

UNIVERSITÄT SALZBURG

Oris vsebine

UNIVERSITÄT SALZBURG

Pristop *One-Extra-Bit*

- Prednosti gossip-pristopa:
  - majhen informiran vzorec ljudi se hitro razširi,
  - razmerja iz začetnega vzorca se pri tem ne pokvarijo preveč
- Prednosti pristopa 2-vzorčenja:
  - po vsaki iteraciji se razmerja iz prejšnjega vzorca močno izboljšajo,
  - imamo zagotovilo da sistem konvergira k večinski barvi
- `One-Extra-Bit` uporabi prednosti obeh!

UNIVERSITÄT SALZBURG

Pristop *One-Extra-Bit*

- Prednosti gossip-pristopa:
  - majhen informiran vzorec ljudi se hitro razširi,
  - razmerja iz začetnega vzorca se pri tem ne pokvarijo preveč
- Prednosti pristopa 2-vzorčenja:
  - po vsaki iteraciji se razmerja iz prejšnjega vzorca močno izboljšajo,
  - imamo zagotovilo da sistem konvergira k večinski barvi
- `One-Extra-Bit` uporabi prednosti obeh!

UNIVERSITÄT SALZBURG

Pristop *One-Extra-Bit*

- Prednosti gossip-pristopa:
  - majhen informiran vzorec ljudi se hitro razširi,
  - razmerja iz začetnega vzorca se pri tem ne pokvarijo preveč
- Prednosti pristopa 2-vzorčenja:
  - po vsaki iteraciji se razmerja iz prejšnjega vzorca močno izboljšajo,
  - imamo zagotovilo da sistem konvergira k večinski barvi
- `One-Extra-Bit` uporabi prednosti obeh!

Pristop *One-Extra-Bit*

- Prednosti gossip-pristopa:
  - majhen informiran vzorec ljudi se hitro razširi,
  - razmerja iz začetnega vzorca se pri tem ne pokvarijo preveč
- Prednosti pristopa 2-vzorčenja:
  - po vsaki iteraciji se razmerja iz prejšnjega vzorca močno izboljšajo,
  - imamo zagotovilo da sistem konvergira k večinski barvi
- `One-Extra-Bit` uporabi prednosti obeh!

UNIVERSITÄT SALZBURG

Pristop *One-Extra-Bit*

- Prednosti gossip-pristopa:
  - majhen informiran vzorec ljudi se hitro razširi,
  - razmerja iz začetnega vzorca se pri tem ne pokvarijo preveč
- Prednosti pristopa 2-vzorčenja:
  - po vsaki iteraciji se razmerja iz prejšnjega vzorca močno izboljšajo,
  - imamo zagotovilo da sistem konvergira k večinski barvi
- `One-Extra-Bit` uporabi prednosti obeh!

UNIVERSITÄT SALZBURG

Pristop *One-Extra-Bit*

- Prednosti gossip-pristopa:
  - majhen informiran vzorec ljudi se hitro razširi,
  - razmerja iz začetnega vzorca se pri tem ne pokvarijo preveč
- Prednosti pristopa 2-vzorčenja:
  - po vsaki iteraciji se razmerja iz prejšnjega vzorca močno izboljšajo,
  - imamo zagotovilo da sistem konvergira k večinski barvi
- `One-Extra-Bit` uporabi prednosti obeh!

UNIVERSITÄT SALZBURG

Pristop *One-Extra-Bit*

- Prednosti gossip-pristopa:
  - majhen informiran vzorec ljudi se hitro razširi,
  - razmerja iz začetnega vzorca se pri tem ne pokvarijo preveč
- Prednosti pristopa 2-vzorčenja:
  - po vsaki iteraciji se razmerja iz prejšnjega vzorca močno izboljšajo,
  - imamo zagotovilo da sistem konvergira k večinski barvi
- `One-Extra-Bit` uporabi prednosti obeh!

UNIVERSITÄT SALZBURG

Pristop One-Extra-Bit

Algoritem:

V vsaki izmed $O(\log \log n)$ faz:

- poljubno vozlišče v izvede korak 2-vzorčenja, ter v primeru neuspeha izbriše svojo barvo
- v nadalnjih $O(\log k + \log \log n)$ korakih sledi gossip-pristop:
  - če je v brez barve, naključno izžreba vozlišče u
  - če ima u barvo, postane to tudi barva od v

Povezane objave:

- Elsässer, Friedetzky, Kaaser, Mallmann-Trenn and Trinker (2017+)
- Berenbrink, Friedetzky, Giakkoupis, Kling, (2016). *Efficient plurality consensus, or: the benefits of cleaning up from time to time.*
- Ghaffari, Parter, (2016). *A polylogarithmic gossip algorithm for plurality consensus.*

UNIVERSITÄT SALZBURG

Pristop One-Extra-Bit

Algoritem:

V vsaki izmed $O(\log \log n)$ faz:

- poljubno vozlišče v izvede korak 2-vzorčenja, ter v primeru neuspeha izbriše svojo barvo
- v nadaljnjih $O(\log k + \log \log n)$ korakih sledi gossip-pristop:
  - če je v brez barve, naključno izžreba vozlišče u
  - če ima u barvo, postane to tudi barva od v

Povezane objave:

- Elsässer, Friedetzky, Kaaser, Mallmann-Trenn and Trinker (2017+)
- Berenbrink, Friedetzky, Giakkoupis, Kling, (2016). *Efficient plurality consensus, or: the benefits of cleaning up from time to time.*
- Ghaffari, Parter, (2016). *A polylogarithmic gossip algorithm for plurality consensus.*

UNIVERSITÄT SALZBURG

Pristop One-Extra-Bit

Algoritem:

V vsaki izmed $O(\log \log n)$ faz:

- poljubno vozlišče v izvede korak 2-vzorčenja, ter v primeru neuspeha izbriše svojo barvo
- v nadalnjih $O(\log k + \log \log n)$ korakih sledi gossip-pristop:
  - če je v brez barve, naključno izžreba vozlišče u
  - če ima u barvo, postane to tudi barva od v

Povezane objave:

- Elsässer, Friedetzky, Kaaser, Mallmann-Trenn and Trinker (2017+)
- Berenbrink, Friedetzky, Giakkoupis, Kling, (2016). *Efficient plurality consensus, or: the benefits of cleaning up from time to time.*
- Ghaffari, Parter, (2016). *A polylogarithmic gossip algorithm for plurality consensus.*

UNIVERSITÄT SALZBURG

Pristop One-Extra-Bit

Algoritem:

V vsaki izmed $O(\log \log n)$ faz:

- poljubno vozlišče v izvede korak 2-vzorčenja, ter v primeru neuspeha izbriše svojo barvo
- v nadalnjih $O(\log k + \log \log n)$ korakih sledi gossip-pristop:
  - če je v brez barve, naključno izžreba vozlišče u
  - če ima u barvo, postane to tudi barva od v

Povezane objave:

- Elsässer, Friedetzky, Kaaser, Mallmann-Trenn and Trinker (2017+)
- Berenbrink, Friedetzky, Giakkoupis, Kling, (2016). *Efficient plurality consensus, or: the benefits of cleaning up from time to time.*
- Ghaffari, Parter, (2016). *A polylogarithmic gossip algorithm for plurality consensus.*

Pristop One-Extra-Bit

Algoritem:

V vsaki izmed $O(\log \log n)$ faz:

- poljubno vozlišče v izvede korak 2-vzorčenja, ter v primeru neuspeha izbriše svojo barvo
- v nadalnjih $O(\log k + \log \log n)$ korakih sledi gossip-pristop:
  - če je v brez barve, naključno izžreba vozlišče u
  - če ima u barvo, postane to tudi barva od v

Povezane objave:

- Elsässer, Friedetzky, Kaaser, Mallmann-Trenn and Trinker (2017+)
- Berenbrink, Friedetzky, Giakkoupis, Kling, (2016). *Efficient plurality consensus, or: the benefits of cleaning up from time to time.*
- Ghaffari, Parter, (2016). *A polylogarithmic gossip algorithm for plurality consensus.*

UNIVERSITÄT SALZBURG

Pristop One-Extra-Bit

Algoritem:

V vsaki izmed $O(\log \log n)$ faz:

- poljubno vozlišče v izvede korak 2-vzorčenja, ter v primeru neuspeha izbriše svojo barvo
- v nadalnjih $O(\log k + \log \log n)$ korakih sledi gossip-pristop:
  - če je v brez barve, naključno izžreba vozlišče u
  - če ima u barvo, postane to tudi barva od v

Povezane objave:

- Elsässer, Friedetzky, Kaaser, Mallmann-Trenn and Trinker (2017+)
- Berenbrink, Friedetzky, Giakkoupis, Kling, (2016). *Efficient plurality consensus, or: the benefits of cleaning up from time to time.*
- Ghaffari, Parter, (2016). *A polylogarithmic gossip algorithm for plurality consensus.*

UNIVERSITÄT SALZBURG

Pristop One-Extra-Bit

Algoritem:

V vsaki izmed $O(\log \log n)$ faz:

- poljubno vozlišče v izvede korak 2-vzorčenja, ter v primeru neuspeha izbriše svojo barvo
- v nadalnjih $O(\log k + \log \log n)$ korakih sledi gossip-pristop:
  - če je v brez barve, naključno izžreba vozlišče u
  - če ima u barvo, postane to tudi barva od v

Povezane objave:

- Elsässer, Friedetzky, Kaaser, Mallmann-Trenn and Trinker (2017+)
- Berenbrink, Friedetzky, Giakkoupis, Kling, (2016). *Efficient plurality consensus, or: the benefits of cleaning up from time to time.*
- Ghaffari, Parter, (2016). *A polylogarithmic gossip algorithm for plurality consensus.*

UNIVERSITÄT SALZBURG

Pristop One-Extra-Bit

Algoritem:

V vsaki izmed $O(\log \log n)$ faz:

- poljubno vozlišče v izvede korak 2-vzorčenja, ter v primeru neuspeha izbriše svojo barvo
- v nadalnjih $O(\log k + \log \log n)$ korakih sledi gossip-pristop:
  - če je v brez barve, naključno izžreba vozlišče u
  - če ima u barvo, postane to tudi barva od v

Povezane objave:

- Elsässer, Friedetzky, Kaaser, Mallmann-Trenn and Trinker (2017+)
- Berenbrink, Friedetzky, Giakkoupis, Kling, (2016). *Efficient plurality consensus, or: the benefits of cleaning up from time to time.*
- Ghaffari, Parter, (2016). *A polylogarithmic gossip algorithm for plurality consensus.*

UNIVERSITÄT SALZBURG

Pristop One-Extra-Bit

Algoritem:

V vsaki izmed $O(\log \log n)$ faz:

- poljubno vozlišče v izvede korak 2-vzorčenja, ter v primeru neuspeha izbriše svojo barvo
- v nadalnjih $O(\log k + \log \log n)$ korakih sledi gossip-pristop:
  - če je v brez barve, naključno izžreba vozlišče u
  - če ima u barvo, postane to tudi barva od v

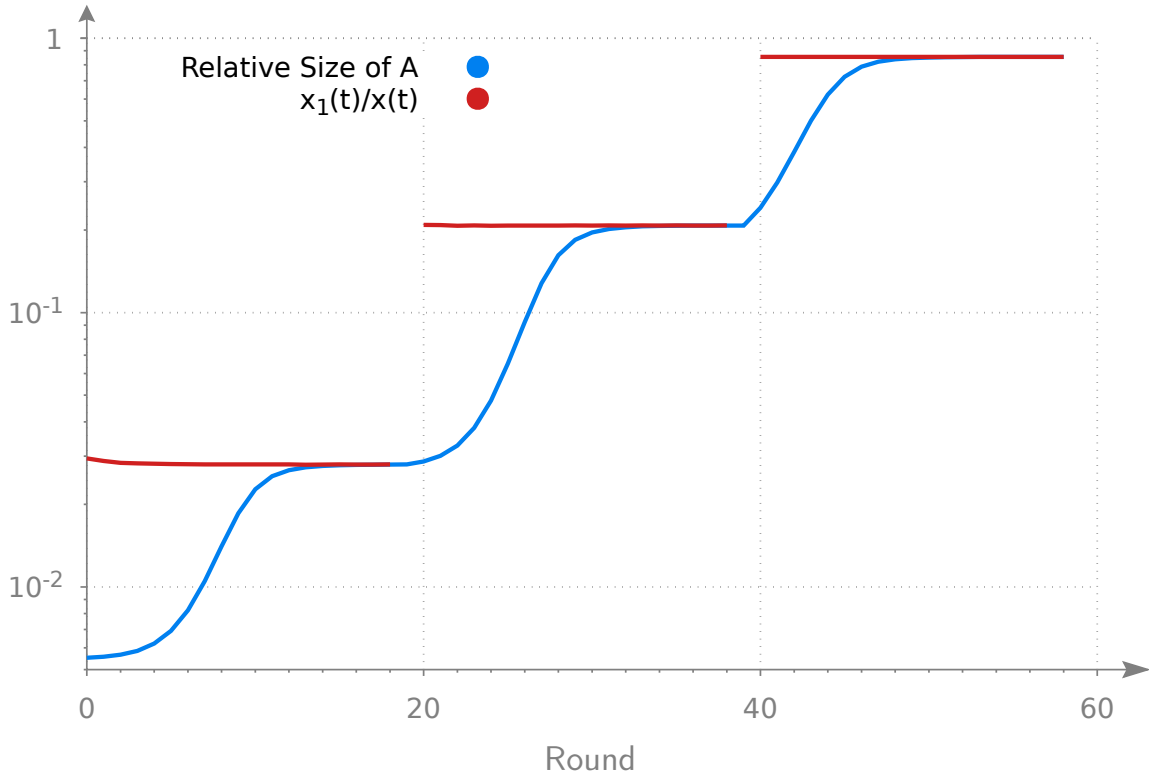
Povezane objave:

- Elsässer, Friedetzky, Kaaser, Mallmann-Trenn and Trinker (2017+)
- Berenbrink, Friedetzky, Giakkoupis, Kling, (2016). *Efficient plurality consensus, or: the benefits of cleaning up from time to time.*
- Ghaffari, Parter, (2016). *A polylogarithmic gossip algorithm for plurality consensus.*

UNIVERSITÄT SALZBURG

One-Extra-Bit

Napredovanje začetno-dominantne barve skozi čas izvedve algoritma One-Extra-Bit



Problem konsenza

Demonstracija...

Oris vsebine

UNIVERSITÄT SALZBURG

Kam naprej?
Prostor za izboljšave...

- Za vsako vozlišče v si zapomnimo posebno število, t.i. *generacijo*, ki je povezana z verjetnostjo da je v pravilno informirana.
- Iz začetnih podatkov si lahko vsako vozlišče izračuna ciljno generacijo, kjer se bo zgodil konsenz.
- V vsaki fazi ni potrebno čakat, da vsa vozlišča dosežejo neko generacijo – izguba časa! Po enem 2-vzorčenju naredimo le $\sim \log \frac{(\alpha+k)^2}{\alpha^2+k} = O(\log k)$ *gossip*-korakov.
- Prevedba v ne-sinhroniziran modelu z zakasnitvami: – vpeljava majhnih de-centraliziranih gruč.

UNIVERSITÄT SALZBURG

Kam naprej?
Prostor za izboljšave...

- Za vsako vozlišče v si zapomnimo posebno število, t.i. *generacijo*, ki je povezana z verjetnostjo da je v pravilno informirana.
- Iz začetnih podatkov si lahko vsako vozlišče izračuna ciljno generacijo, kjer se bo zgodil konsenz.
- V vsaki fazi ni potrebno čakat, da vsa vozlišča dosežejo neko generacijo – izguba časa! Po enem 2-vzorčenju naredimo le $\sim \log \frac{(\alpha+k)^2}{\alpha^2+k} = O(\log k)$ *gossip*-korakov.
- Prevedba v ne-sinhroniziran modelu z zakasnitvami: – vpeljava majhnih de-centraliziranih gruč.

UNIVERSITÄT SALZBURG

Kam naprej?
Prostor za izboljšave...

- Za vsako vozlišče v si zapomnimo posebno število, t.i. *generacijo*, ki je povezana z verjetnostjo da je v pravilno informirana.
- Iz začetnih podatkov si lahko vsako vozlišče izračuna ciljno generacijo, kjer se bo zgodil konsenz.
- V vsaki fazi ni potrebno čakat, da vsa vozlišča dosežejo neko generacijo – izguba časa! Po enem 2-vzorčenju naredimo le $\sim \log \frac{(\alpha+k)^2}{\alpha^2+k} = O(\log k)$ *gossip*-korakov.
- Prevedba v ne-sinhroniziran modelu z zakasnitvami: – vpeljava majhnih de-centraliziranih gruč.

UNIVERSITÄT SALZBURG

Kam naprej?
Prostor za izboljšave...

- Za vsako vozlišče v si zapomnimo posebno število, t.i. *generacijo*, ki je povezana z verjetnostjo da je v pravilno informirana.
- Iz začetnih podatkov si lahko vsako vozlišče izračuna ciljno generacijo, kjer se bo zgodil konsenz.
- V vsaki fazi ni potrebno čakat, da vsa vozlišča dosežejo neko generacijo – izguba časa! Po enem 2-vzorčenju naredimo le $\sim \log \frac{(\alpha+k)^2}{\alpha^2+k} = O(\log k)$ *gossip*-korakov.
- Prevedba v ne-sinhroniziran modelu z zakasnitvami: – vpeljava majhnih de-centraliziranih gruč.

UNIVERSITÄT SALZBURG

Problem konsenza

Psevdokoda

Ideja pri sinhroniziranem modelu

- Vsaka generacija loči dve "življenski obdobji": Obdobje 2-vzorčenja ter "gossip" obdobje.
- Algoritem pametno preklaplja med obema, ter ob primernem času dodaja nove generacije.

Algorithm 1 The basic synchronous procedure.

Require: a vertex $v \in V(G)$

1: sample neighbors v' and v'' u.a.r.
2: wlog. assume $\text{gen}(v') \geq \text{gen}(v'')$
3: **if** $t \in \{t_i\}_{i \in [\mathcal{G}^*]}$ and $\text{gen}(v) \leq \text{gen}(v') = \text{gen}(v'')$ and $col(v') = col(v'')$ **then**
4: $\quad \text{gen}(v) \leftarrow \text{gen}(v') + 1$
5: $\quad col(v) \leftarrow col(v')$
6: **else if** $\text{gen}(v') > \text{gen}(v)$ **then**
7: $\quad \text{gen}(v) \leftarrow \text{gen}(v')$
8: $\quad col(v) \leftarrow col(v')$

UNIVERSITÄT SALZBURG

Problem konsenza
Psevdokoda

Ideja pri sinhroniziranem modelu

- Vsaka generacija loči dve "življenski obdobji": Obdobje 2-vzorčenja ter "gossip" obdobje.
- Algoritem pametno preklaplja med obema, ter ob primernem času dodaja nove generacije.

Algorithm 1 The basic synchronous procedure.

Require: a vertex $v \in V(G)$

1: sample neighbors v' and v'' u.a.r.
2: wlog. assume $\text{gen}(v') \geq \text{gen}(v'')$
3: **if** $t \in \{t_i\}_{i \in [\mathcal{G}^*]}$ and $\text{gen}(v) \leq \text{gen}(v') = \text{gen}(v'')$ and $col(v') = col(v'')$ **then**
4: $\quad \text{gen}(v) \leftarrow \text{gen}(v') + 1$
5: $\quad col(v) \leftarrow col(v')$
6: **else if** $\text{gen}(v') > \text{gen}(v)$ **then**
7: $\quad \text{gen}(v) \leftarrow \text{gen}(v')$
8: $\quad col(v) \leftarrow col(v')$

UNIVERSITÄT SALZBURG

Problem konsenza
Psevdokoda

Ideja pri sinhroniziranem modelu

- Vsaka generacija loči dve "življenski obdobji": Obdobje 2-vzorčenja ter "gossip" obdobje.
- Algoritem pametno preklaplja med obema, ter ob primernem času dodaja nove generacije.

Algorithm 1 The basic synchronous procedure.

Require: a vertex $v \in V(G)$

1: sample neighbors v' and v'' u.a.r.
2: wlog. assume $\text{gen}(v') \geq \text{gen}(v'')$
3: **if** $t \in \{t_i\}_{i \in [\mathcal{G}^*]}$ and $\text{gen}(v) \leq \text{gen}(v') = \text{gen}(v'')$ and $col(v') = col(v'')$ **then**
4: $\quad \text{gen}(v) \leftarrow \text{gen}(v') + 1$
5: $\quad col(v) \leftarrow col(v')$
6: **else if** $\text{gen}(v') > \text{gen}(v)$ **then**
7: $\quad \text{gen}(v) \leftarrow \text{gen}(v')$
8: $\quad col(v) \leftarrow col(v')$

UNIVERSITÄT SALZBURG

Oris vsebine

UNIVERSITÄT SALZBURG

Analiza ene faze algoritma
Nekaj definicij

- $g_t(i)$ – delež vozlišč, ki so v času t v generaciji i.
- $c_{i,j,t}$ – delež vozlišč, ki so ob času t iz generacije i, ter hkrati barve j.
- Izmed števil $1, \ldots, k$ na bosta a in b zaporedoma začetno najštevilčnejši barvi.
- $\alpha_{i,t} = \frac{c_{a,i,t}}{c_{b,i,t}}$ – pristranskost k barvi a, v generaciji i ob času t.

UNIVERSITÄT SALZBURG

Analiza ene faze algoritma

Nekaj definicij

- $g_t(i)$ – delež vozlišč, ki so v času t v generaciji i.
- $c_{i,j,t}$ – delež vozlišč, ki so ob času t iz generacije i, ter hkrati barve j.
- Izmed števil $1, \ldots, k$ na bosta a in b zaporedoma začetno najštevilčnejši barvi.
- $\alpha_{i,t} = \frac{c_{a,i,t}}{c_{b,i,t}}$ – pristranskost k barvi a, v generaciji i ob času t.

UNIVERSITÄT SALZBURG

Analiza ene faze algoritma

Nekaj definicij

- $g_t(i)$ – delež vozlišč, ki so v času t v generaciji i.
- $c_{i,j,t}$ – delež vozlišč, ki so ob času t iz generacije i, ter hkrati barve j.
- Izmed števil $1, \ldots, k$ na bosta a in b zaporedoma začetno najštevilčnejši barvi.
- $\alpha_{i,t} = \frac{c_{a,i,t}}{c_{b,i,t}}$ – pristranskost k barvi a, v generaciji i ob času t.

UNIVERSITÄT SALZBURG

Analiza ene faze algoritma
Nekaj definicij

- $g_t(i)$ – delež vozlišč, ki so v času t v generaciji i.
- $c_{i,j,t}$ – delež vozlišč, ki so ob času t iz generacije i, ter hkrati barve j.
- Izmed števil $1, \ldots, k$ na bosta a in b zaporedoma začetno najštevilčnejši barvi.
- $\alpha_{i,t} = \frac{c_{a,i,t}}{c_{b,i,t}}$ – pristranskost k barvi a, v generaciji i ob času t.

UNIVERSITÄT SALZBURG

Analiza ene faze algoritma
Distribucija generacij

- Verjetnost da je naključni 2-vzorec uspešen označimo s $p_i \sim \frac{\alpha_0^{2^{i+1}}+k-1}{\left(\alpha_0^{2^i}+k-1\right)^2}$.
- Zgornja meja potrebnega časa, da se velikost generacije i *namnoži* do linearnega delaža γ je: $X_i = \log_{2-\gamma}\left(\frac{1}{p_i\gamma}\right)+2$.
- Nova generacija t_i se *rodi* ob predpisanem času $t_i = 1+\sum_{j=1}^{i-1} X_j$.
- V prvi časovni enoti "po rojstvu", sklepamo koliko se se nova generacija "namnoži":

$$g_{t_i-1}(i-1) \geq \gamma \Longrightarrow g_{t_i}(i) > \gamma^2 p_i \left(1-n^{-\varepsilon}\right) > \frac{\gamma^2}{k}\left(1-n^{-\varepsilon}\right).$$

- V nadalnjih $X_i - 1$ korakih delež generacije i zraste vsaj do γ – kar je pogoj za rojstvo generacije $i+1$.

UNIVERSITÄT SALZBURG

Analiza ene faze algoritma
Distribucija generacij

- Verjetnost da je naključni 2-vzorec uspešen označimo s $p_i \sim \frac{\alpha_0^{2^{i+1}}+k-1}{\left(\alpha_0^{2^i}+k-1\right)^2}$.
- Zgornja meja potrebnega časa, da se velikost generacije *i namnoži* do linearnega delaža γ je: $X_i = \log_{2-\gamma}\left(\frac{1}{p_i\gamma}\right)+2$.
- Nova generacija t_i se *rodi* ob predpisanem času $t_i = 1+\sum_{j=1}^{i-1} X_j$.
- V prvi časovni enoti "po rojstvu", sklepamo koliko se se nova generacija "namnoži":

$$g_{t_i-1}(i-1) \geq \gamma \Longrightarrow g_{t_i}(i) > \gamma^2 p_i \left(1-n^{-\varepsilon}\right) > \frac{\gamma^2}{k}\left(1-n^{-\varepsilon}\right).$$

- V nadalnjih $X_i - 1$ korakih delež generacije i zraste vsaj do γ – kar je pogoj za rojstvo generacije $i+1$.

UNIVERSITÄT SALZBURG

Analiza ene faze algoritma
Distribucija generacij

- Verjetnost da je naključni 2-vzorec uspešen označimo s $p_i \sim \frac{\alpha_0^{2^{i+1}}+k-1}{\left(\alpha_0^{2^i}+k-1\right)^2}$.
- Zgornja meja potrebnega časa, da se velikost generacije i *namnoži* do linearnega delaža γ je: $X_i = \log_{2-\gamma}\left(\frac{1}{p_i\gamma}\right)+2$.
- Nova generacija t_i se *rodi* ob predpisanem času $t_i = 1+\sum_{j=1}^{i-1} X_j$.
- V prvi časovni enoti "po rojstvu", sklepamo koliko se se nova generacija "namnoži":

$$g_{t_i-1}(i-1) \geq \gamma \Longrightarrow g_{t_i}(i) > \gamma^2 p_i \left(1-n^{-\varepsilon}\right) > \frac{\gamma^2}{k}\left(1-n^{-\varepsilon}\right).$$

- V nadalnjih $X_i - 1$ korakih delež generacije i zraste vsaj do γ – kar je pogoj za rojstvo generacije $i+1$.

UNIVERSITÄT SALZBURG

Analiza ene faze algoritma
Distribucija generacij

- Verjetnost da je naključni 2-vzorec uspešen označimo s $p_i \sim \frac{\alpha_0^{2^{i+1}}+k-1}{\left(\alpha_0^{2^i}+k-1\right)^2}$.
- Zgornja meja potrebnega časa, da se velikost generacije i *namnoži* do linearnega delaža γ je: $X_i = \log_{2-\gamma}\left(\frac{1}{p_i\gamma}\right)+2$.
- Nova generacija t_i se *rodi* ob predpisanem času $t_i = 1+\sum_{j=1}^{i-1} X_j$.
- V prvi časovni enoti "po rojstvu", sklepamo koliko se se nova generacija "namnoži":

$$g_{t_i-1}(i-1) \geq \gamma \Longrightarrow g_{t_i}(i) > \gamma^2 p_i \left(1-n^{-\varepsilon}\right) > \frac{\gamma^2}{k}\left(1-n^{-\varepsilon}\right).$$

- V nadalnjih $X_i - 1$ korakih delež generacije i zraste vsaj do γ – kar je pogoj za rojstvo generacije $i+1$.

Analiza ene faze algoritma
Distribucija generacij

- Verjetnost da je naključni 2-vzorec uspešen označimo s $p_i \sim \frac{\alpha_0^{2^{i+1}}+k-1}{\left(\alpha_0^{2^i}+k-1\right)^2}$.
- Zgornja meja potrebnega časa, da se velikost generacije i *namnoži* do linearnega delaža γ je: $X_i = \log_{2-\gamma}\left(\frac{1}{p_i\gamma}\right)+2$.
- Nova generacija t_i se *rodi* ob predpisanem času $t_i = 1+\sum_{j=1}^{i-1} X_j$.
- V prvi časovni enoti "po rojstvu", sklepamo koliko se se nova generacija "namnoži":

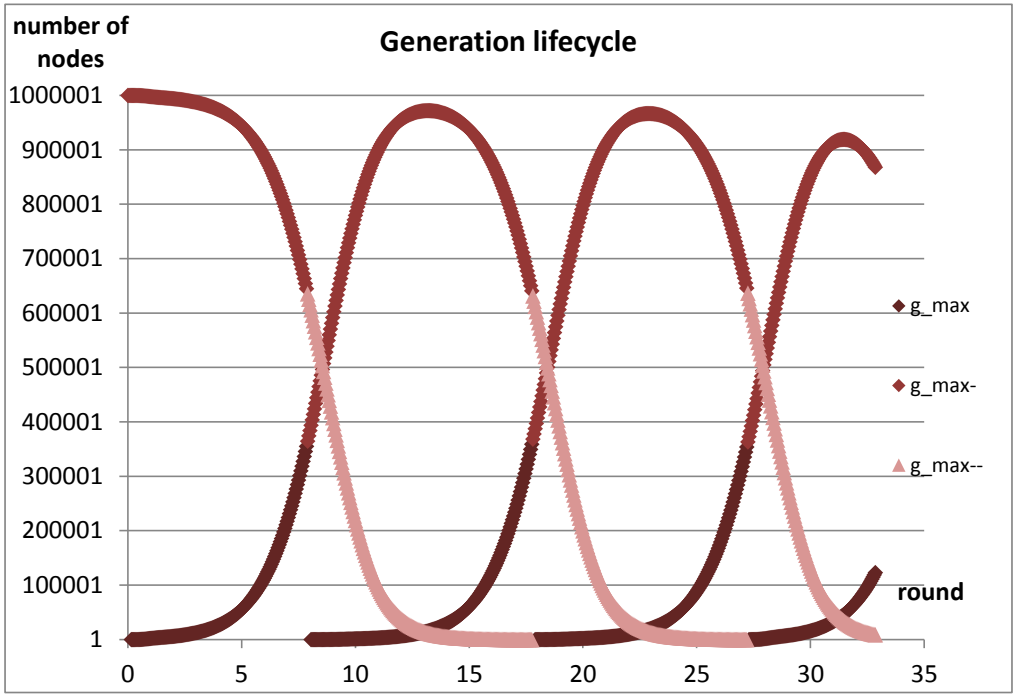
$$g_{t_i-1}(i-1) \geq \gamma \Longrightarrow g_{t_i}(i) > \gamma^2 p_i \left(1-n^{-\varepsilon}\right) > \frac{\gamma^2}{k}\left(1-n^{-\varepsilon}\right).$$

- V nadalnjih $X_i - 1$ korakih delež generacije i zraste vsaj do γ – kar je pogoj za rojstvo generacije $i+1$.

UNIVERSITÄT SALZBURG

Analiza ene faze algoritma

Življenjski cikel generacije



Analiza ene faze algoritma

Glavne trditve

Lemma (Prvi korak v fazi.)

Ob "rojstvu" generacije i je vrednost α_{i,t_i} zgoščena okrog pričakovane vrednosti α^2_{i-1,t_i-1}.

Lemma (Rast v poljubnem gossip-koraku)

Med poljubnim gossip-korakom se pristranskost ohranja.

Corollary

Z veliko verjetnostjo je vrednost $\alpha_{i+1,t_{i+1}}$ zelo blizu pričakovani vrednosti α^2_{i,t_i}.

UNIVERSITÄT SALZBURG

Analiza ene faze algoritma

Glavne trditve

Lemma (Prvi korak v fazi.)

Ob "rojstvu" generacije i je vrednost α_{i,t_i} zgoščena okrog pričakovane vrednosti α^2_{i-1,t_i-1}.

Lemma (Rast v poljubnem gossip-koraku)

Med poljubnim gossip-korakom se pristranskost ohranja.

Corollary

Z veliko verjetnostjo je vrednost $\alpha_{i+1,t_{i+1}}$ zelo blizu pričakovani vrednosti α^2_{i,t_i}.

UNIVERSITÄT SALZBURG

Analiza ene faze algoritma

Glavne trditve

Lemma (Prvi korak v fazi.)

Ob "rojstvu" generacije i je vrednost α_{i,t_i} zgoščena okrog pričakovane vrednosti α^2_{i-1,t_i-1}.

Lemma (Rast v poljubnem gossip-koraku)

Med poljubnim gossip-korakom se pristranskost ohranja.

Corollary

Z veliko verjetnostjo je vrednost $\alpha_{i+1,t_{i+1}}$ zelo blizu pričakovani vrednosti α^2_{i,t_i}.

UNIVERSITÄT SALZBURG

Analiza ene faze algoritma

Prva lema: začetna pristranskost v novi generaciji.

For $t = t_i - 1$, $\alpha_{i-1,t} \leq k$ and $\delta = \frac{k \cdot n^{\varepsilon}}{\sqrt{n}}$ with $0 < \varepsilon < \frac{1}{2} - \log_n k$, we have

$$\alpha_{i,t+1} = \frac{B\left(n, (g_t(i-1)c_{a,i-1,t})^2\right)}{B\left(n, (g_t(i-1)c_{b,i-1,t})^2\right)}$$

$$\overset{\text{whp.}}{>} \frac{c^2_{a,i-1,t}(1-\delta)}{c^2_{b,i-1,t}(1+\delta)} \geq \alpha^2_{i-1,t} \cdot (1-2\delta).$$

For large values of $\alpha_{i-1,t} \geq \sqrt{n}$ are handled separately.

UNIVERSITÄT SALZBURG

Analiza ene faze algoritma

Prva lema: začetna pristranskost v novi generaciji.

For $t = t_i - 1$, $\alpha_{i-1,t} \leq k$ and $\delta = \frac{k \cdot n^{\varepsilon}}{\sqrt{n}}$ with $0 < \varepsilon < \frac{1}{2} - \log_n k$, we have

$$\alpha_{i,t+1} = \frac{B\left(n, (g_t(i-1)c_{a,i-1,t})^2\right)}{B\left(n, (g_t(i-1)c_{b,i-1,t})^2\right)}$$
$$\overset{\text{whp.}}{>} \frac{c^2_{a,i-1,t}(1-\delta)}{c^2_{b,i-1,t}(1+\delta)} \geq \alpha^2_{i-1,t} \cdot (1-2\delta).$$

For large values of $\alpha_{i-1,t} \geq \sqrt{n}$ are handled separately.

UNIVERSITÄT SALZBURG

Analiza ene faze algoritma

Druga lema: "šum" v enem časovnem koraku.

Set again $\delta = \frac{k \cdot n^{\varepsilon}}{\sqrt{n}}$ with $0 < \varepsilon < \frac{1}{2} - \log_n k$ and $x = g_t(i)$, an increase of nodes of opinion j in time $t \to t+1$ may be modeled by $B\left(n(1-x), x \cdot c_{j,i,t}\right)$. Then,

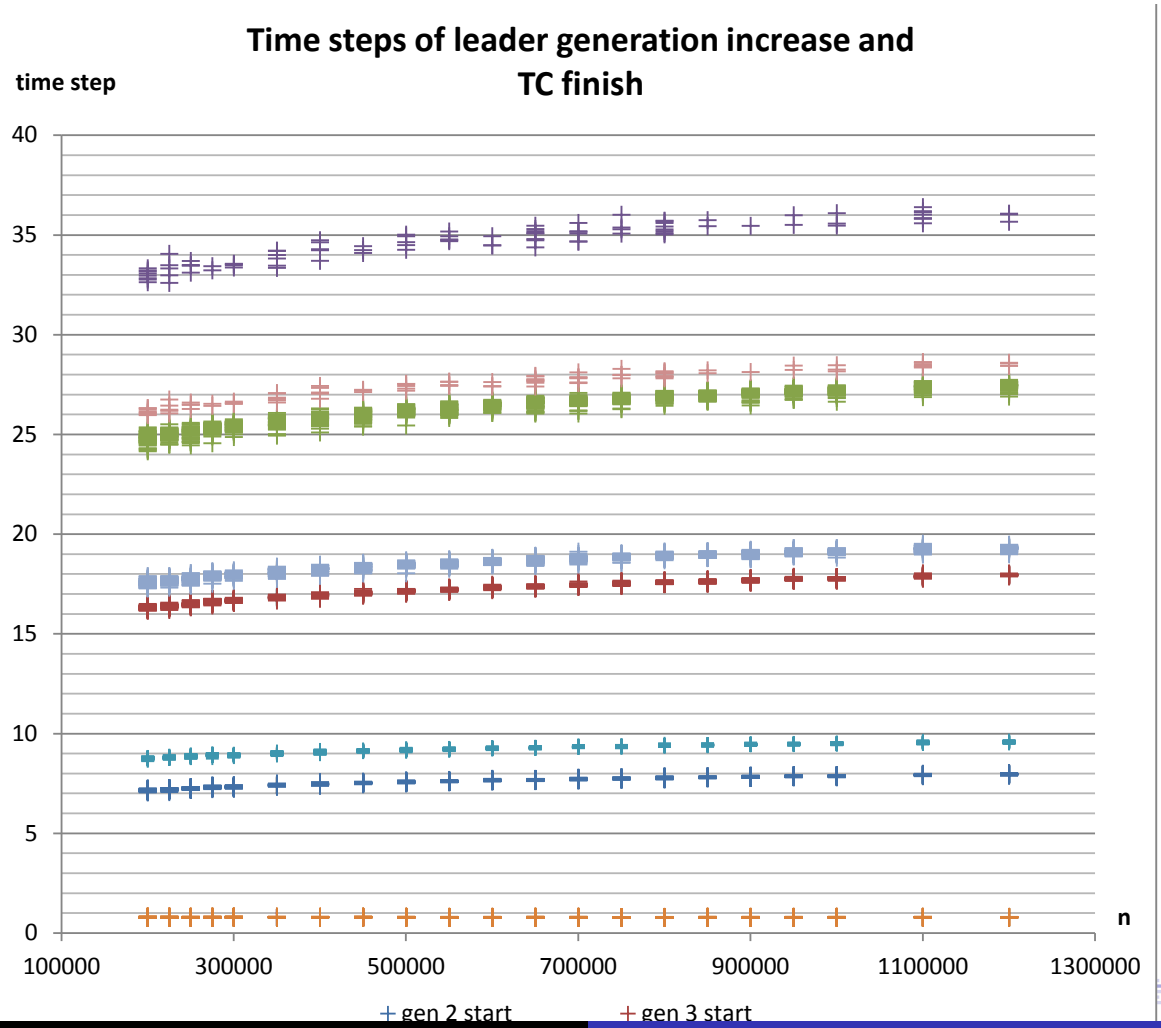
$$\alpha_{i,t+1} = \frac{c_{a,i,t+1}}{c_{b,i,t+1}} = \frac{n \cdot x \cdot c_{a,i,t} + B\left(n(1-x), x \cdot c_{a,i,t}\right)}{n \cdot x \cdot c_{b,i,t} + B\left(n(1-x), x \cdot c_{b,i,t}\right)}$$

$$\overset{\text{whp.}}{>} \frac{c_{a,i,t} + (1-x)\, c_{a,i,t}\,(1-\delta)}{c_{b,i,t} + (1-x)\, c_{b,i,t}\,(1+\delta)}$$

$$= \alpha_{i,t} \cdot \left(1 - \frac{(1-x)\,2\delta}{2+\delta}\right) > \alpha_{i,t} \cdot (1-\delta).$$

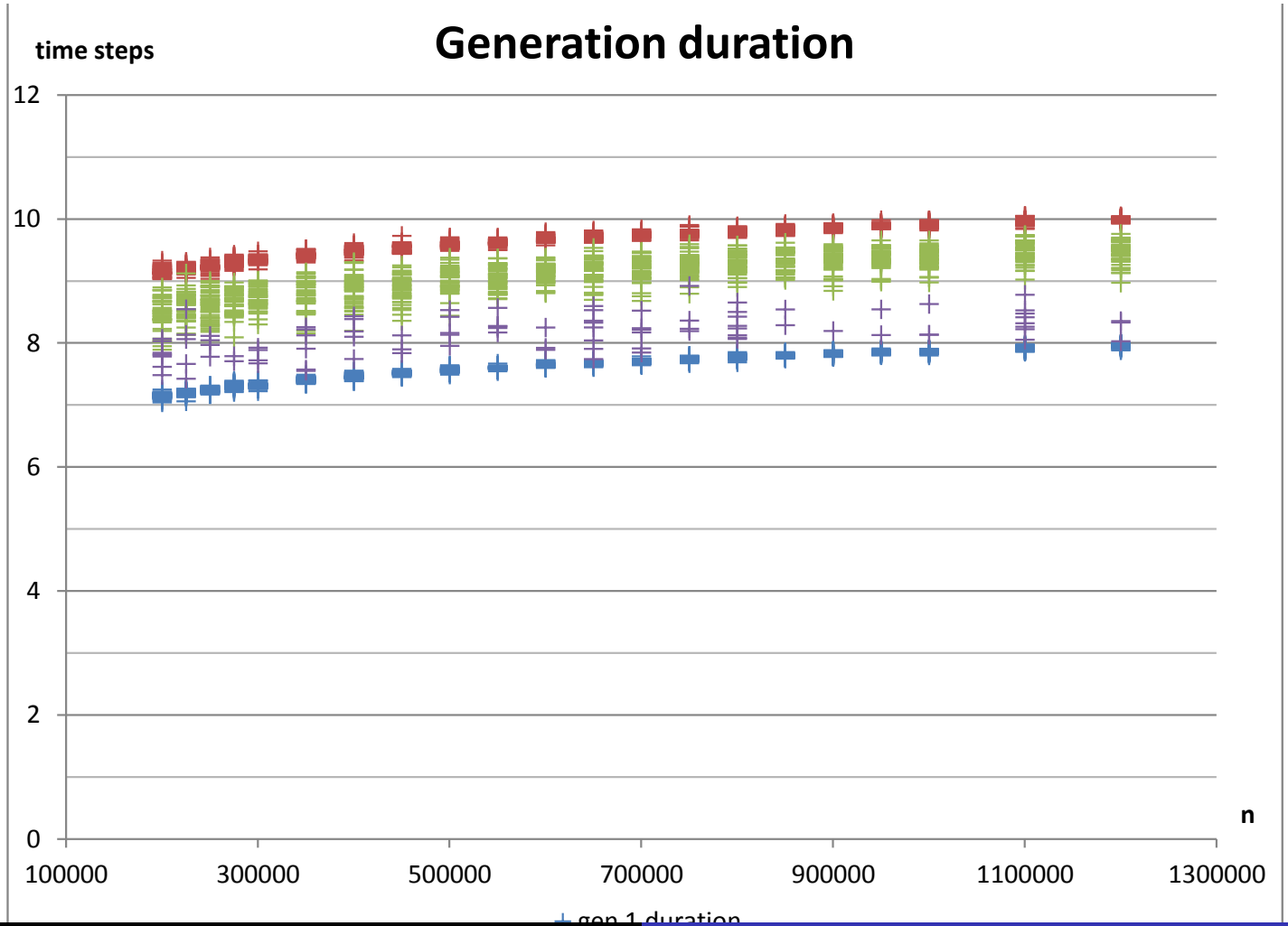
Analiza ene faze algoritma

Prikaz "šuma" tekom izvajanja.



Analiza ene faze algoritma

Prikaz "šuma" tekom izvajanja.



Analiza ene faze algoritma

Lepljenje obeh lem.

- Intuition: throughout the phase, measure the multiplicative error.
- Use first lemma once, and second lemma $X_i - 1 = O(\log k)$ times.

Corollary

Whp. $\alpha_{g+1,t_{g+1}}$ is very close to its expected value α^2_{g,t_g}, in particular we have

$$\alpha_{g+1,t_{g+1}} \geq \alpha^2_{g,t_g} \cdot \left(1 - n^{-\varepsilon'}\right),$$

for any $0 < \varepsilon' < \varepsilon$.

Analiza ene faze algoritma

Lepljenje obeh lem.

- Intuition: throughout the phase, measure the multiplicative error.
- Use first lemma once, and second lemma $X_i - 1 = O(\log k)$ times.

Corollary

Whp. $\alpha_{g+1,t_{g+1}}$ is very close to its expected value α^2_{g,t_g}, in particular we have

$$\alpha_{g+1,t_{g+1}} \geq \alpha^2_{g,t_g} \cdot \left(1 - n^{-\varepsilon'}\right),$$

for any $0 < \varepsilon' < \varepsilon$.

Analiza ene faze algoritma

Lepljenje obeh lem.

- Intuition: throughout the phase, measure the multiplicative error.
- Use first lemma once, and second lemma $X_i - 1 = O(\log k)$ times.

Corollary

Whp. $\alpha_{g+1,t_{g+1}}$ *is very close to its expected value* α^2_{g,t_g}*, in particular we have*

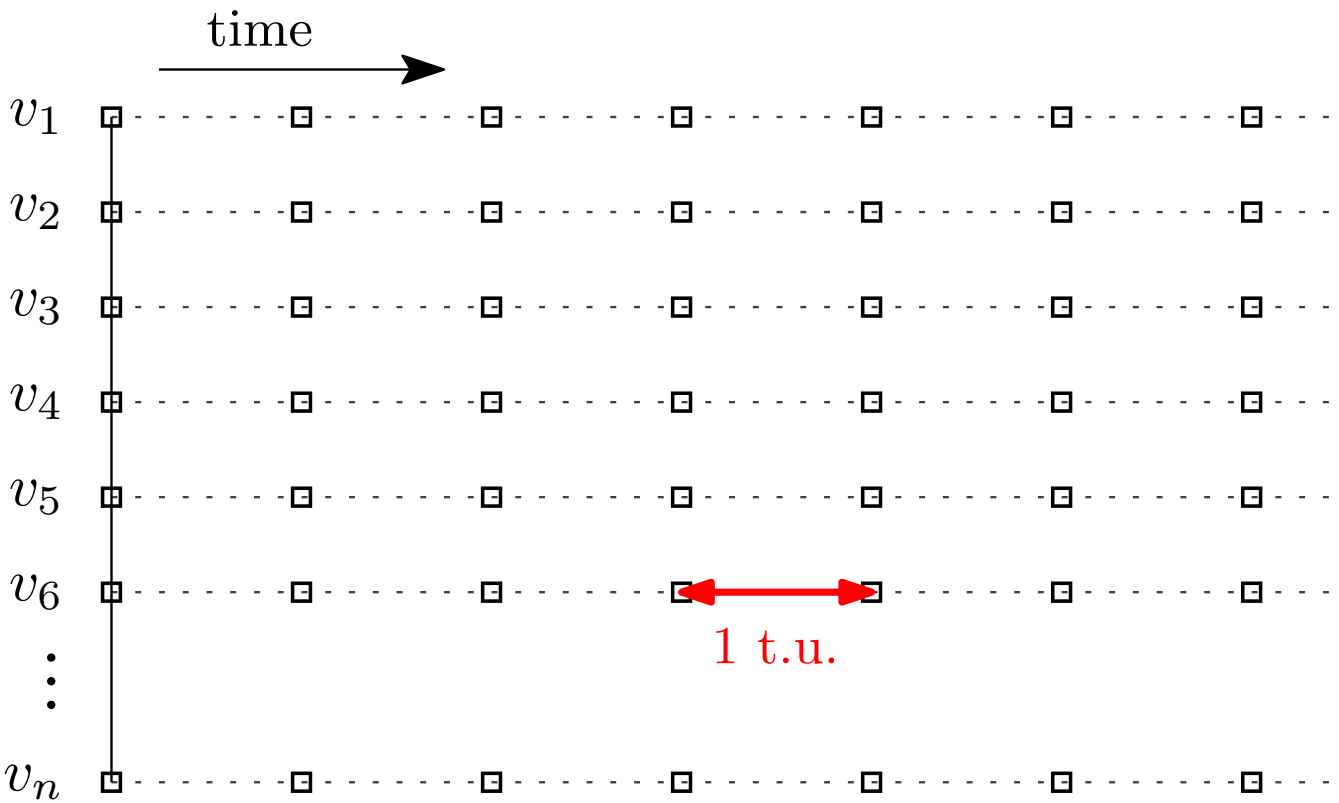
$$\alpha_{g+1,t_{g+1}} \geq \alpha^2_{g,t_g} \cdot \left(1 - n^{-\varepsilon'}\right),$$

for any $0 < \varepsilon' < \varepsilon$.

Oris vsebine

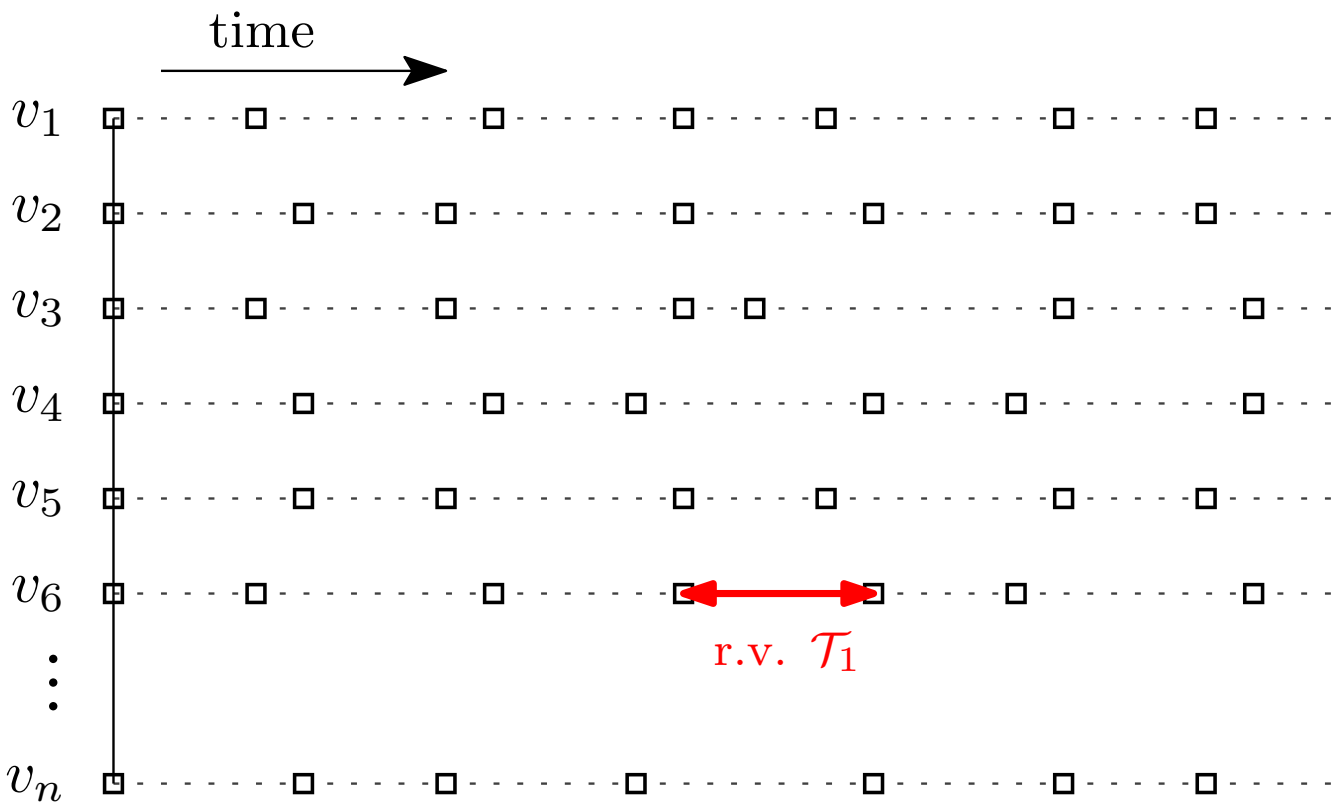
UNIVERSITÄT SALZBURG

Sinhroniziran model



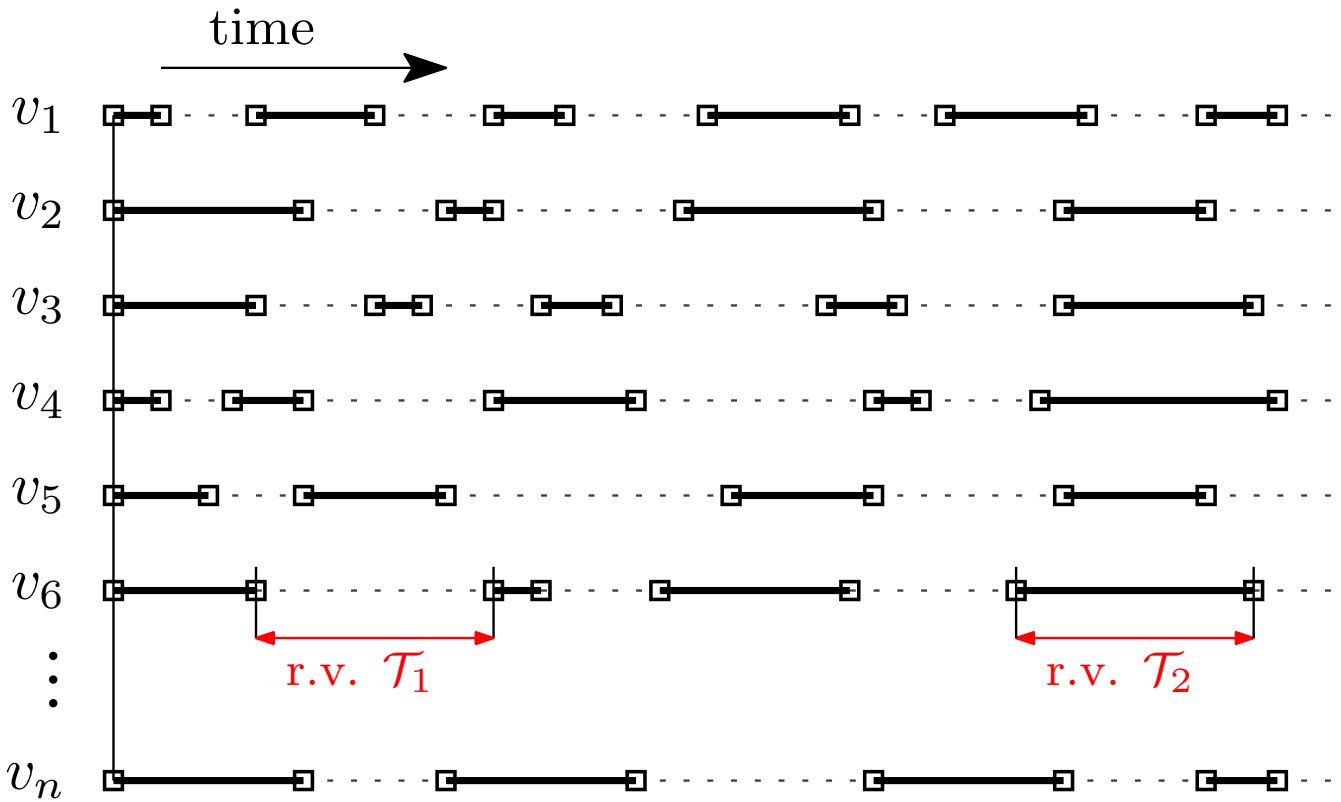
Model brez sinhronizacije

Poenostavljen model



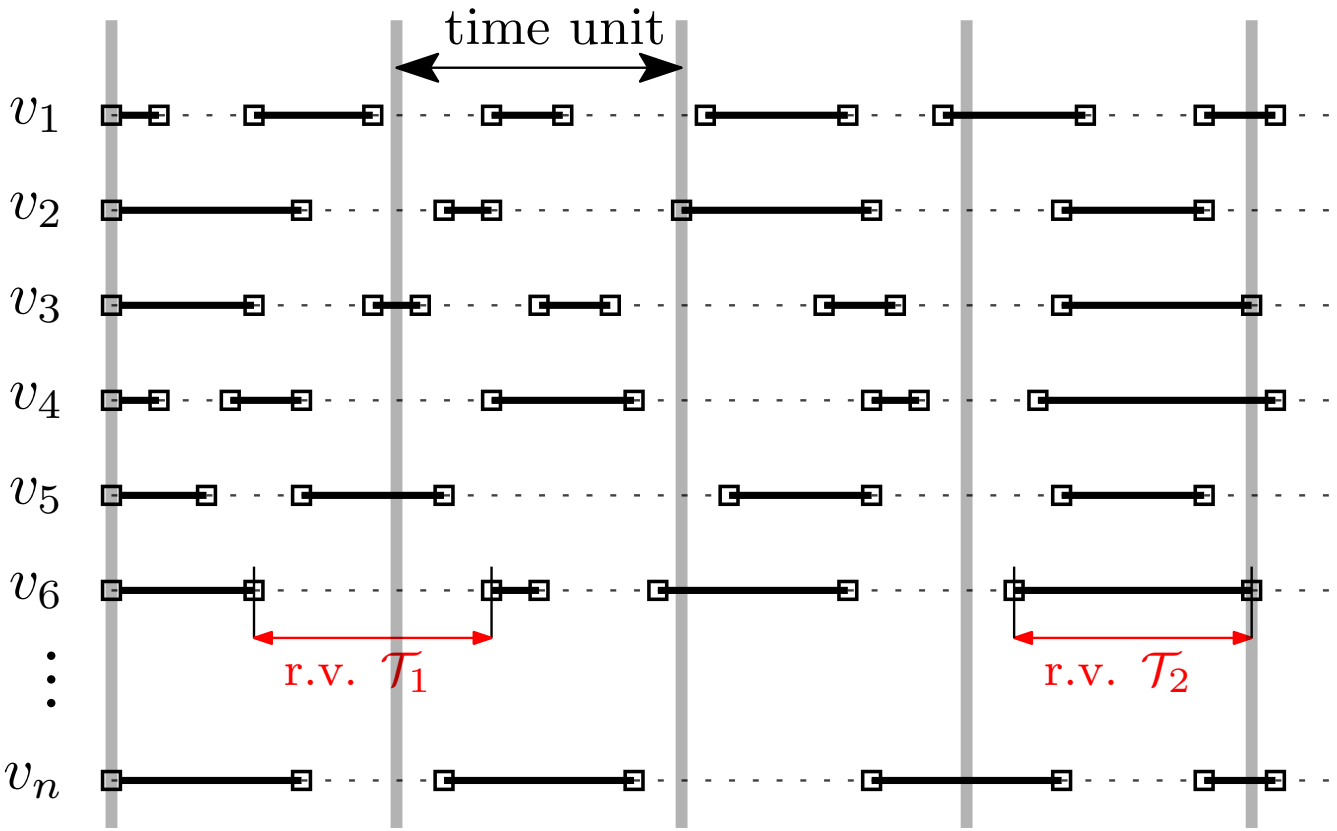
Model brez sinhronizacije

Model z zakasnitvami



Model brez sinhronizacije

Model z zakasnitvami



Ne-sinhroniziran model
Opis modela

1. Poljubno vozlišče čaka $\mathscr{T}_1$ časa, preden začne z izvajanjem ($\mathscr{T}_1$ je zvezna sluč. spremenljivka).
2. Čas izvajanja algoritma (npr. zaradi vzorčenja) ni zanemarljivo kratek, ampak traja $\mathscr{T}_2$ časovnih enot.
3. Vozlišče lahko vzpostavi komunikacijo z naključnim vozliščem omrežja, ali pa tudi z enim izmed že-kontaktiranih (iz zgodovine) – zapomnimo si lahko le en tak naslov.
4. Vozlišča poznajo število n, lahko "odgovorijo" na omejeno število zahtevkov – $\log n$, ter znajo generirati naključna števila.

UNIVERSITÄT SALZBURG

Ne-sinhroniziran model

Opis modela

1. Poljubno vozlišče čaka $\mathscr{T}_1$ časa, preden začne z izvajanjem ($\mathscr{T}_1$ je zvezna sluč. spremenljivka).
2. Čas izvajanja algoritma (npr. zaradi vzorčenja) ni zanemarljivo kratek, ampak traja $\mathscr{T}_2$ časovnih enot.
3. Vozlišče lahko vzpostavi komunikacijo z naključnim vozliščem omrežja, ali pa tudi z enim izmed že-kontaktiranih (iz zgodovine) – zapomnimo si lahko le en tak naslov.
4. Vozlišča poznajo število n, lahko "odgovorijo" na omejeno število zahtevkov – $\log n$, ter znajo generirati naključna števila.

UNIVERSITÄT SALZBURG

Ne-sinhroniziran model
Opis modela

1. Poljubno vozlišče čaka $\mathscr{T}_1$ časa, preden začne z izvajanjem ($\mathscr{T}_1$ je zvezna sluč. spremenljivka).
2. Čas izvajanja algoritma (npr. zaradi vzorčenja) ni zanemarljivo kratek, ampak traja $\mathscr{T}_2$ časovnih enot.
3. Vozlišče lahko vzpostavi komunikacijo z naključnim vozliščem omrežja, ali pa tudi z enim izmed že-kontaktiranih (iz zgodovine) – zapomnimo si lahko le en tak naslov.
4. Vozlišča poznajo število n, lahko "odgovorijo" na omejeno število zahtevkov – $\log n$, ter znajo generirati naključna števila.

UNIVERSITÄT SALZBURG

Ne-sinhroniziran model

Opis modela

1. Poljubno vozlišče čaka $\mathscr{T}_1$ časa, preden začne z izvajanjem ($\mathscr{T}_1$ je zvezna sluč. spremenljivka).
2. Čas izvajanja algoritma (npr. zaradi vzorčenja) ni zanemarljivo kratek, ampak traja $\mathscr{T}_2$ časovnih enot.
3. Vozlišče lahko vzpostavi komunikacijo z naključnim vozliščem omrežja, ali pa tudi z enim izmed že-kontaktiranih (iz zgodovine) – zapomnimo si lahko le en tak naslov.
4. Vozlišča poznajo število n, lahko "odgovorijo" na omejeno število zahtevkov – $\log n$, ter znajo generirati naključna števila.

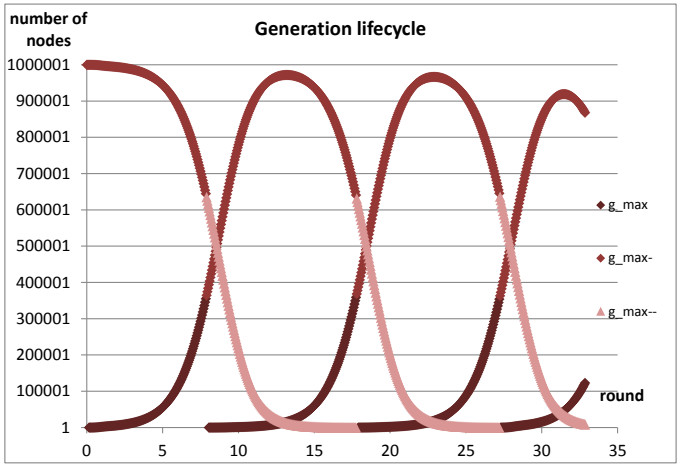
UNIVERSITÄT SALZBURG

Ne-sinhroniziran model

Ideja pri sinhroniziranem modelu

- Vsaka generacija loči dve "življenski obdobji": Obdobje 2-vzorčenja ter "gossip" obdobje.
- Algoritem pametno preklaplja med obema, ter ob primernem času dodaja nove generacije.

Ne potrebujemo podatka o globalnem času; dovolj je za zaznamo določene mejne vrednosti deleža vozlišč v trenutni generaciji.

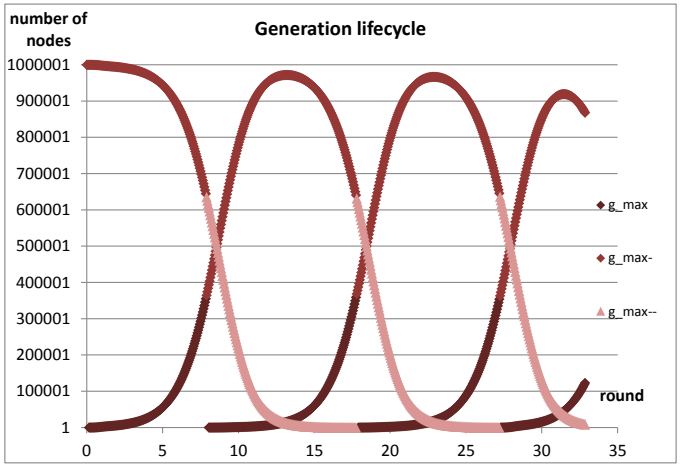


Ne-sinhroniziran model

Ideja pri sinhroniziranem modelu

- Vsaka generacija loči dve "življenski obdobji": Obdobje 2-vzorčenja ter "gossip" obdobje.
- Algoritem pametno preklaplja med obema, ter ob primernem času dodaja nove generacije.

Ne potrebujemo podatka o globalnem času; dovolj je za zaznamo določene mejne vrednosti deleža vozlišč v trenutni generaciji.

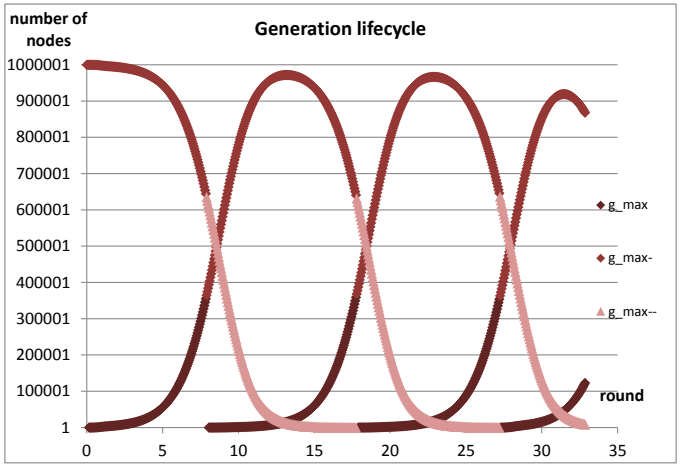


Ne-sinhroniziran model

Ideja pri sinhroniziranem modelu

- Vsaka generacija loči dve "življenski obdobji": Obdobje 2-vzorčenja ter "gossip" obdobje.
- Algoritem pametno preklaplja med obema, ter ob primernem času dodaja nove generacije.

Ne potrebujemo podatka o globalnem času; dovolj je za zaznamo določene mejne vrednosti deleža vozlišč v trenutni generaciji.

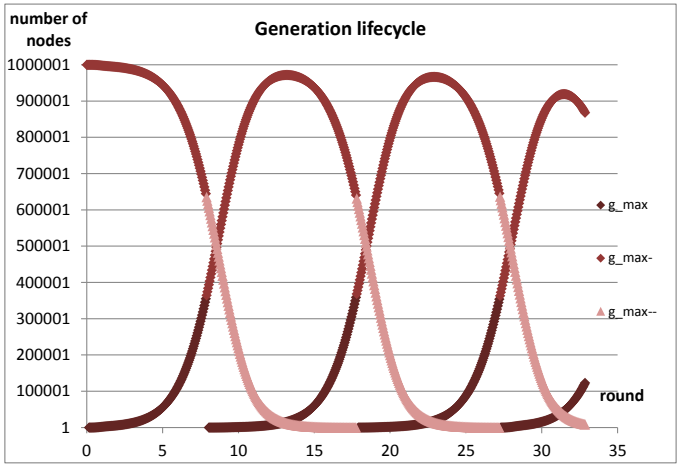


Ne-sinhroniziran model

Ideja pri sinhroniziranem modelu

- Vsaka generacija loči dve "življenski obdobji": Obdobje 2-vzorčenja ter "gossip" obdobje.
- Algoritem pametno preklaplja med obema, ter ob primernem času dodaja nove generacije.

Ne potrebujemo podatka o globalnem času; dovolj je za zaznamo določene mejne vrednosti deleža vozlišč v trenutni generaciji.

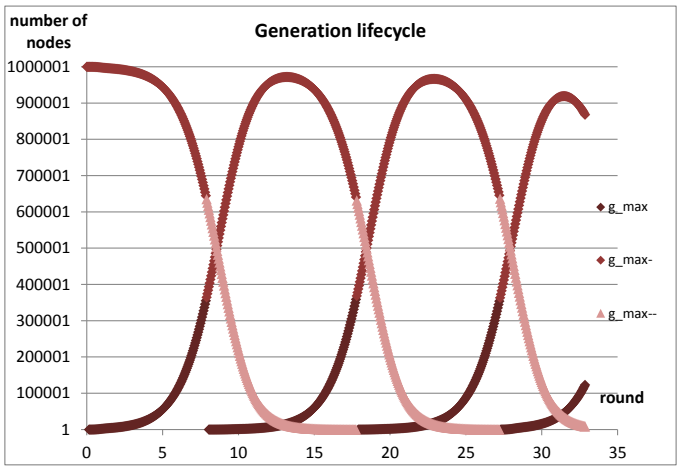


Ne-sinhroniziran model

Ideja pri sinhroniziranem modelu

- Vsaka generacija loči dve "življenski obdobji": Obdobje 2-vzorčenja ter "gossip" obdobje.
- Algoritem pametno preklaplja med obema, ter ob primernem času dodaja nove generacije.

Ne potrebujemo podatka o globalnem času; dovolj je za zaznamo določene mejne vrednosti deleža vozlišč v trenutni generaciji.



Asynchronous Model

How to make a transition while keeping the main ideas/results?

Different approaches:

- Approximate the global time by counting ticks.
- Modify the model by allowing a small shared memory.
- Simulate a small shared memory by a leader-node.
- Simulate one leader by partitioning a graph into several clusters, each *governed* by its own leader-node.

UNIVERSITÄT SALZBURG

Asynchronous Model

How to make a transition while keeping the main ideas/results?

Different approaches:

- Approximate the global time by counting ticks.
- Modify the model by allowing a small shared memory.
- Simulate a small shared memory by a leader-node.
- Simulate one leader by partitioning a graph into several clusters, each *governed* by its own leader-node.

UNIVERSITÄT SALZBURG

Asynchronous Model

How to make a transition while keeping the main ideas/results?

Different approaches:

- Approximate the global time by counting ticks.
- Modify the model by allowing a small shared memory.
- Simulate a small shared memory by a leader-node.
- Simulate one leader by partitioning a graph into several clusters, each *governed* by its own leader-node.

UNIVERSITÄT SALZBURG

Asynchronous Model

How to make a transition while keeping the main ideas/results?

Different approaches:

- Approximate the global time by counting ticks.
- Modify the model by allowing a small shared memory.
- Simulate a small shared memory by a leader-node.
- Simulate one leader by partitioning a graph into several clusters, each *governed* by its own leader-node.

UNIVERSITÄT SALZBURG

Oris vsebine

UNIVERSITÄT SALZBURG

The Procedures

The Simple Leader Election

Algorithm 4 The description of the leader initialization.

Require: a vertex with `leader` $=$ `null` ticked

1: **if** $t = 0$ with probability $\frac{1}{C_0 \log n}$ **then**
2: initialize `leader` $\leftarrow$ `self`
3: $v \leftarrow$ a random neighbor
4: **if** v.`leader` $\neq$ `null` **then**
5: `leader` $\leftarrow v$.`leader`
6: send an 0-signal to `leader`

UNIVERSITÄT SALZBURG

The Procedures
Node's Behavior

- Any fixed node contains the last-seen information from their leader (a generation and a state).
- For each sampled node, we check if it is of higher generation, and in case its leader allows it, we overwrite our color/generation.
- If unsuccessful, we try to preform TC-step, again in accordance with the leader-information of our samples.
- Finally, a node contacts its leader and synchronizes the state and/or highest allowed generation of the cluster.

UNIVERSITÄT SALZBURG

The Procedures
Node's Behavior

- Any fixed node contains the last-seen information from their leader (a generation and a state).
- For each sampled node, we check if it is of higher generation, and in case its leader allows it, we overwrite our color/generation.
- If unsuccessful, we try to preform TC-step, again in accordance with the leader-information of our samples.
- Finally, a node contacts its leader and synchronizes the state and/or highest allowed generation of the cluster.

The Procedures
Node's Behavior

- Any fixed node contains the last-seen information from their leader (a generation and a state).
- For each sampled node, we check if it is of higher generation, and in case its leader allows it, we overwrite our color/generation.
- If unsuccessful, we try to preform TC-step, again in accordance with the leader-information of our samples.
- Finally, a node contacts its leader and synchronizes the state and/or highest allowed generation of the cluster.

The Procedures
Node's Behavior

- Any fixed node contains the last-seen information from their leader (a generation and a state).
- For each sampled node, we check if it is of higher generation, and in case its leader allows it, we overwrite our color/generation.
- If unsuccessful, we try to preform TC-step, again in accordance with the leader-information of our samples.
- Finally, a node contacts its leader and synchronizes the state and/or highest allowed generation of the cluster.

The Procedures

Node's Behavior

Algorithm 5 The basic procedure of a node after clustering.

Require: a vertex v with `leader` $\neq$ `null` ticked

1: from v', v'', sampled u.a.r., request information
2: **if** $\exists \bar{v} \in \{v', v''\} : g(v) < g(\bar{v})$ and $\bar{v}$`.l.state=propagate` **then**
3: $\quad$ `col`$(v) \leftarrow$ `col`(v^*)
4: $\quad$ `gen`$(v) \leftarrow$ `gen`(v^*)
5: **else if** $\exists \bar{v} \in \{v', v''\} : \bar{v}$`.l.gen` $>$ `gen`$(v') =$ `gen`(v'') and `col`$(v') =$ `col`(v'') **then**
6: $\quad$ `col`$(v) \leftarrow$ `col`(v')
7: $\quad$ `gen`$(v) \leftarrow$ `gen`$(v') + 1$
8: contact `leader`, and if applicable:
9: $\quad$ update own `l.gen` and `l.state`,
10: $\quad$ notify `leader` whether any of v', v'' allow gossiping to v`.l.gen`
11: $\quad$ send a generation-update signal

UNIVERSITÄT SALZBURG

The Procedures

Leader Behavior

Algorithm 7 The description of the signals, received by leader.

Require: a i-signal

1: **if** $i = 0$ **then**
2: increment `card`
3: **else if** $i >$ `gen` **then**
4: new-generation(i)
5: **if** $i =$ `gen` **then**
6: increment `gen` $-$ `size`
7: **if** `state` $=$ `null` and `card` $\geq C_3 \log n$ **then** ▷ initial partitioning done
8: `state` $\leftarrow S_1$
9: **else if** `state` $= S_1$ and S_2-condition **then** ▷ allow gossiping
10: `state` $\leftarrow S_2$
11: **else if** `state` $= S_2$ and $\frac{\texttt{gen}-\texttt{size}}{\texttt{card}} \geq \gamma_2$ and `gen` $< \left\lceil \log \frac{\log n}{\alpha_0 - 1} \right\rceil$ **then**
12: new_generation$(\texttt{gen} + 1)$
13: `state` $\leftarrow S_1$

UNIVERSITÄT SALZBURG

Nesinhroniziran model
Nadležni problemi

- Out-of-Sync Scenarios:
  - everyone is monotonically on the path towards higher awareness, hence an out-of-date information is not mis-leading.
- Threshold-based S_2-condition does not suffice anymore!
  - Time based
  - Threshold + Gossip based
- A small group of "delayed nodes" may arrive to a generation from a previous one by the two-choices-method, however, at the time the smaller generation could already be smaller!
- Small clusters?

UNIVERSITÄT SALZBURG

Nesinhroniziran model
Nadležni problemi

- Out-of-Sync Scenarios:
  - everyone is monotonically on the path towards higher awareness, hence an out-of-date information is not mis-leading.
- Threshold-based S_2-condition does not suffice anymore!
  - Time based
  - Threshold + Gossip based
- A small group of "delayed nodes" may arrive to a generation from a previous one by the two-choices-method, however, at the time the smaller generation could already be smaller!
- Small clusters?

UNIVERSITÄT SALZBURG

Nesinhroniziran model
Nadležni problemi

- Out-of-Sync Scenarios:
  - everyone is monotonically on the path towards higher awareness, hence an out-of-date information is not mis-leading.
- Threshold-based S_2-condition does not suffice anymore!
  - Time based
  - Threshold + Gossip based
- A small group of "delayed nodes" may arrive to a generation from a previous one by the two-choices-method, however, at the time the smaller generation could already be smaller!
- Small clusters?

UNIVERSITÄT SALZBURG

Nesinhroniziran model
Nadležni problemi

- Out-of-Sync Scenarios:
  - everyone is monotonically on the path towards higher awareness, hence an out-of-date information is not mis-leading.
- Threshold-based S_2-condition does not suffice anymore!
  - Time based
  - Threshold + Gossip based
- A small group of "delayed nodes" may arrive to a generation from a previous one by the two-choices-method, however, at the time the smaller generation could already be smaller!
- Small clusters?

UNIVERSITÄT SALZBURG

Nesinhroniziran model
Nadležni problemi

- Out-of-Sync Scenarios:
  - everyone is monotonically on the path towards higher awareness, hence an out-of-date information is not mis-leading.
- Threshold-based S_2-condition does not suffice anymore!
  - Time based
  - Threshold + Gossip based
- A small group of "delayed nodes" may arrive to a generation from a previous one by the two-choices-method, however, at the time the smaller generation could already be smaller!
- Small clusters?

UNIVERSITÄT SALZBURG

Nesinhroniziran model
Nadležni problemi

- Out-of-Sync Scenarios:
  - everyone is monotonically on the path towards higher awareness, hence an out-of-date information is not mis-leading.
- Threshold-based S_2-condition does not suffice anymore!
  - Time based
  - Threshold + Gossip based
- A small group of "delayed nodes" may arrive to a generation from a previous one by the two-choices-method, however, at the time the smaller generation could already be smaller!
- Small clusters?

UNIVERSITÄT SALZBURG

Nesinhroniziran model
Nadležni problemi

- Out-of-Sync Scenarios:
  - everyone is monotonically on the path towards higher awareness, hence an out-of-date information is not mis-leading.
- Threshold-based S_2-condition does not suffice anymore!
  - Time based
  - Threshold + Gossip based
- A small group of "delayed nodes" may arrive to a generation from a previous one by the two-choices-method, however, at the time the smaller generation could already be smaller!
- Small clusters?

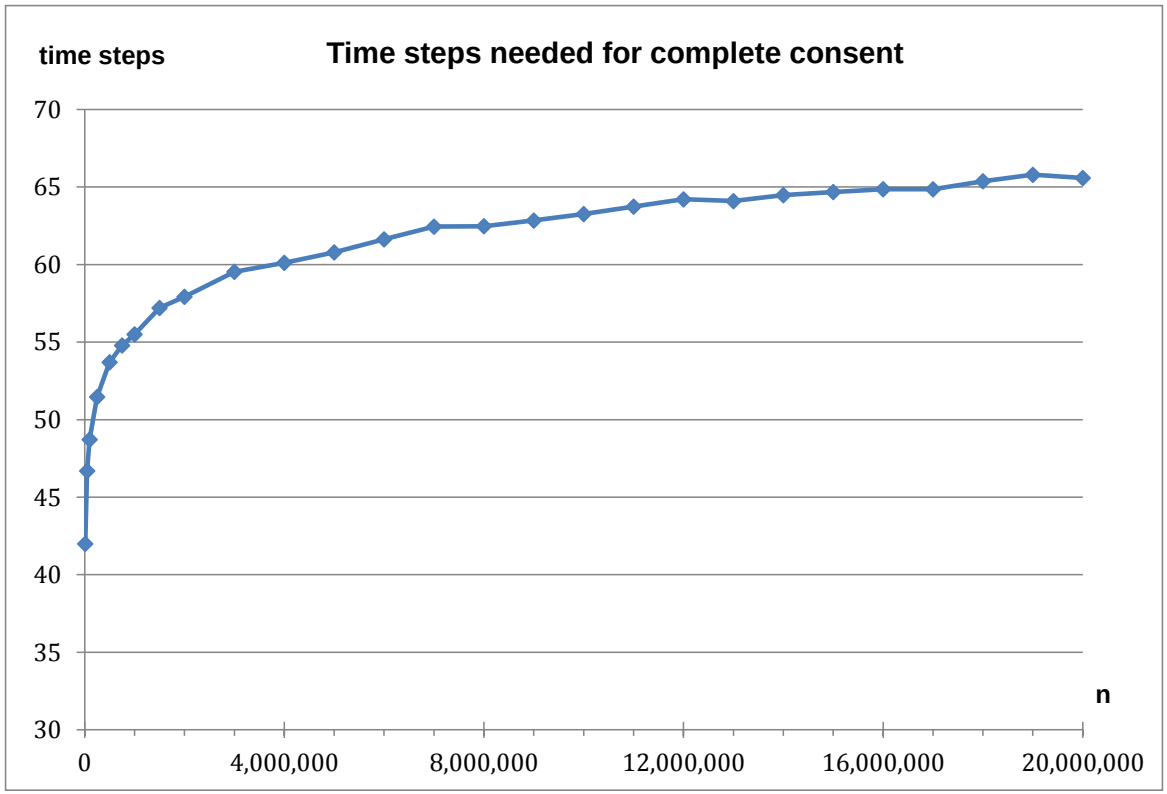
UNIVERSITÄT SALZBURG

Oris vsebine

UNIVERSITÄT SALZBURG

Prikaz rezultatov

Performance with worst-case initial opinions – Asynchronous Protocol



$$k \sim \sqrt[3]{n}$$

$$\alpha \sim 1.8$$

$$c_a : c_b \sim 0.59\% : 0.33\%$$

$$\#\text{gruč} \sim \frac{n}{5 \log n}$$

delay :

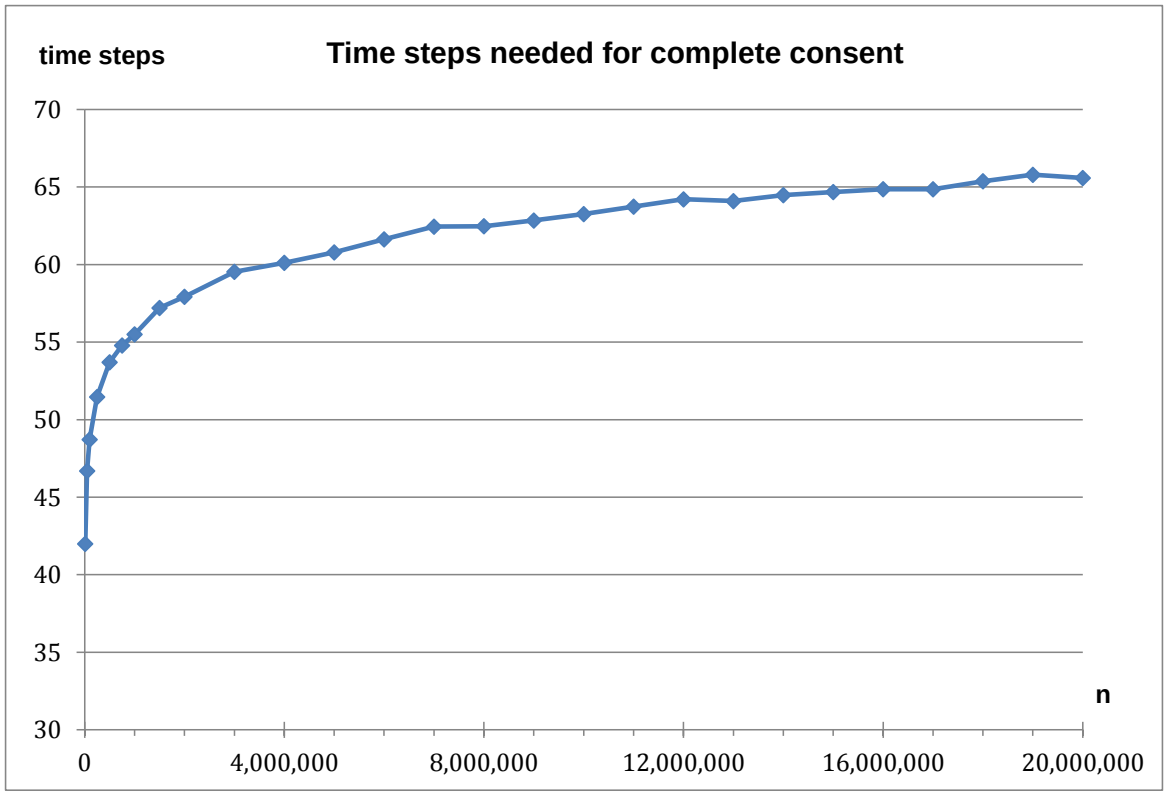
$$\mathcal{T}_1 = \text{Poisson}(1)$$

$$\mathcal{T}_2 = \text{Poisson}\left(\frac{1}{4}\right)$$

UNIVERSITÄT SALZBURG

Prikaz rezultatov

Performance with worst-case initial opinions – Asynchronous Protocol



$$k \sim \sqrt[3]{n}$$

$$\alpha \sim 1.8$$

$$c_a : c_b \sim 0.59\% : 0.33\%$$

$$\#\text{gruč} \sim \frac{n}{5 \log n}$$

delay :

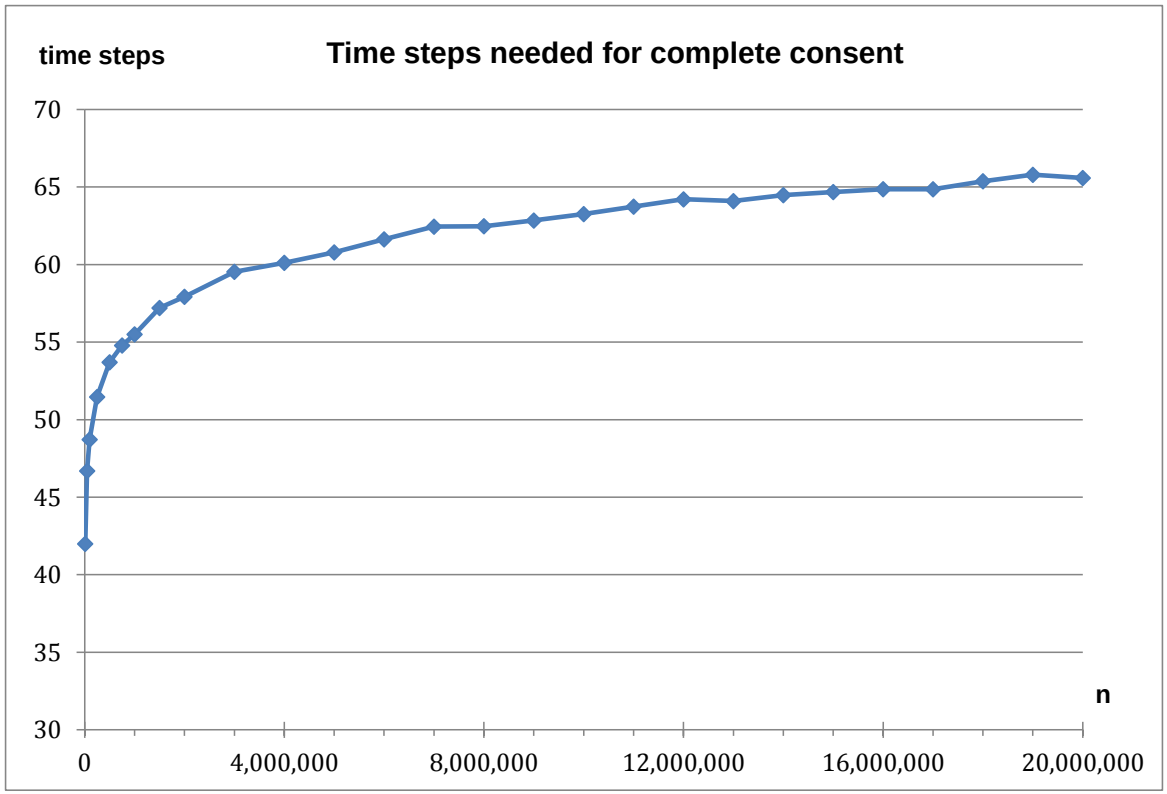
$$\mathcal{T}_1 = \text{Poisson}(1)$$

$$\mathcal{T}_2 = \text{Poisson}\left(\frac{1}{4}\right)$$

UNIVERSITÄT SALZBURG

Prikaz rezultatov

Performance with worst-case initial opinions – Asynchronous Protocol



$$k \sim \sqrt[3]{n}$$

$$\alpha \sim 1.8$$

$$c_a : c_b \sim 0.59\% : 0.33\%$$

$$\#\text{gruč} \sim \frac{n}{5 \log n}$$

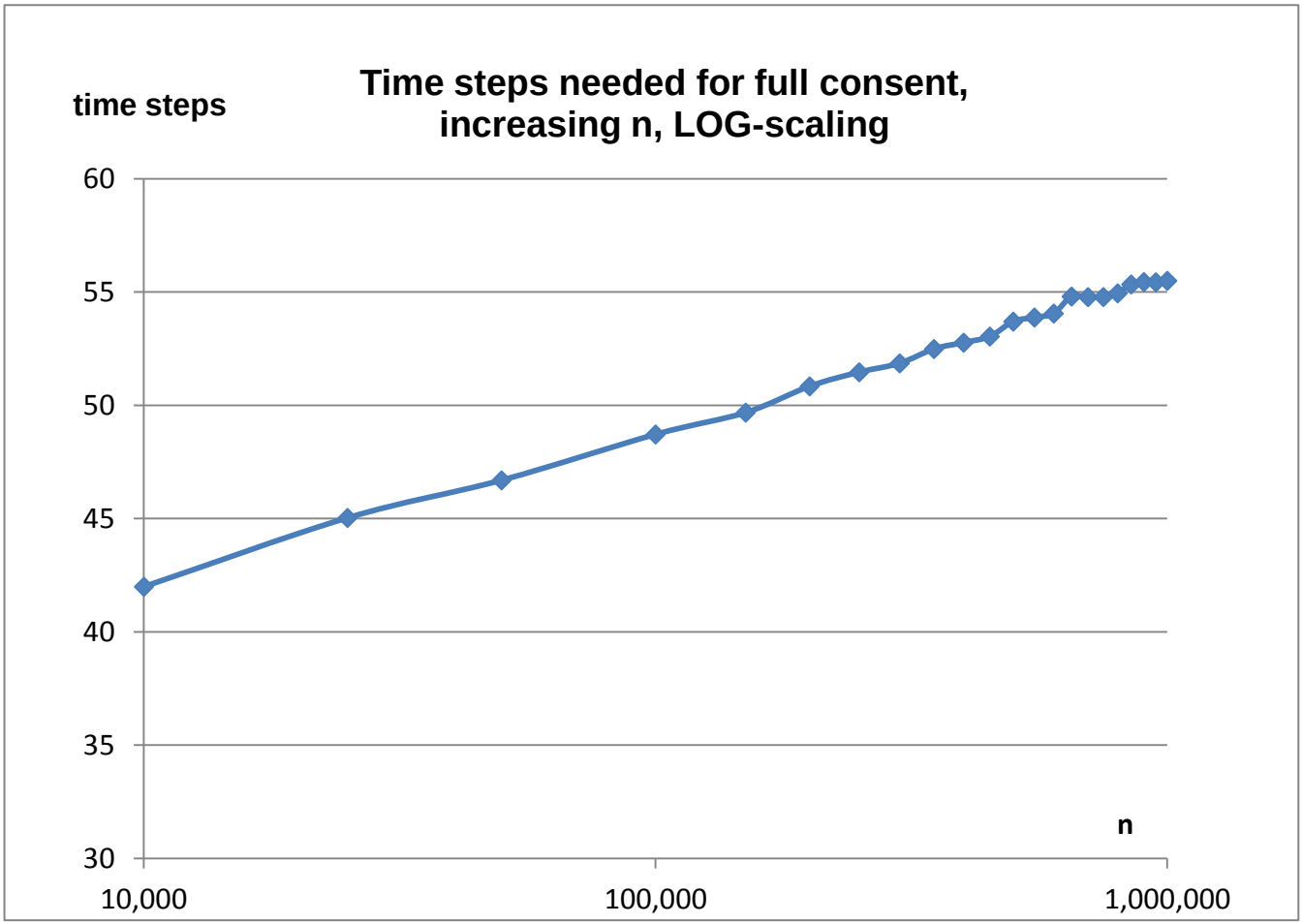
delay :

$$\mathscr{T}_1 = \text{Poisson}(1)$$

$$\mathscr{T}_2 = \text{Poisson}\left(\frac{1}{4}\right)$$

UNIVERSITÄT SALZBURG

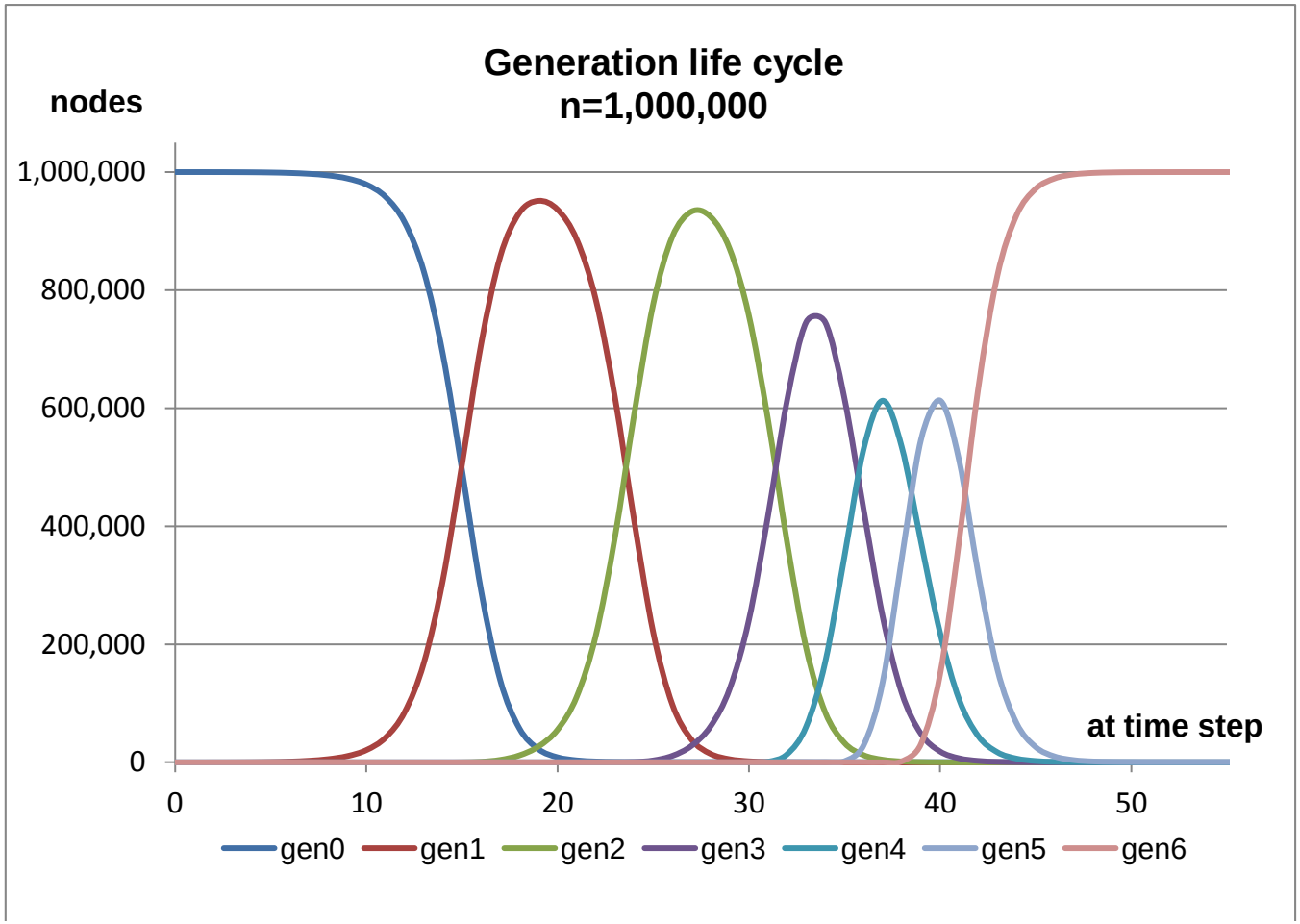
Prikaz rezultatov



UNIVERSITÄT SALZBURG

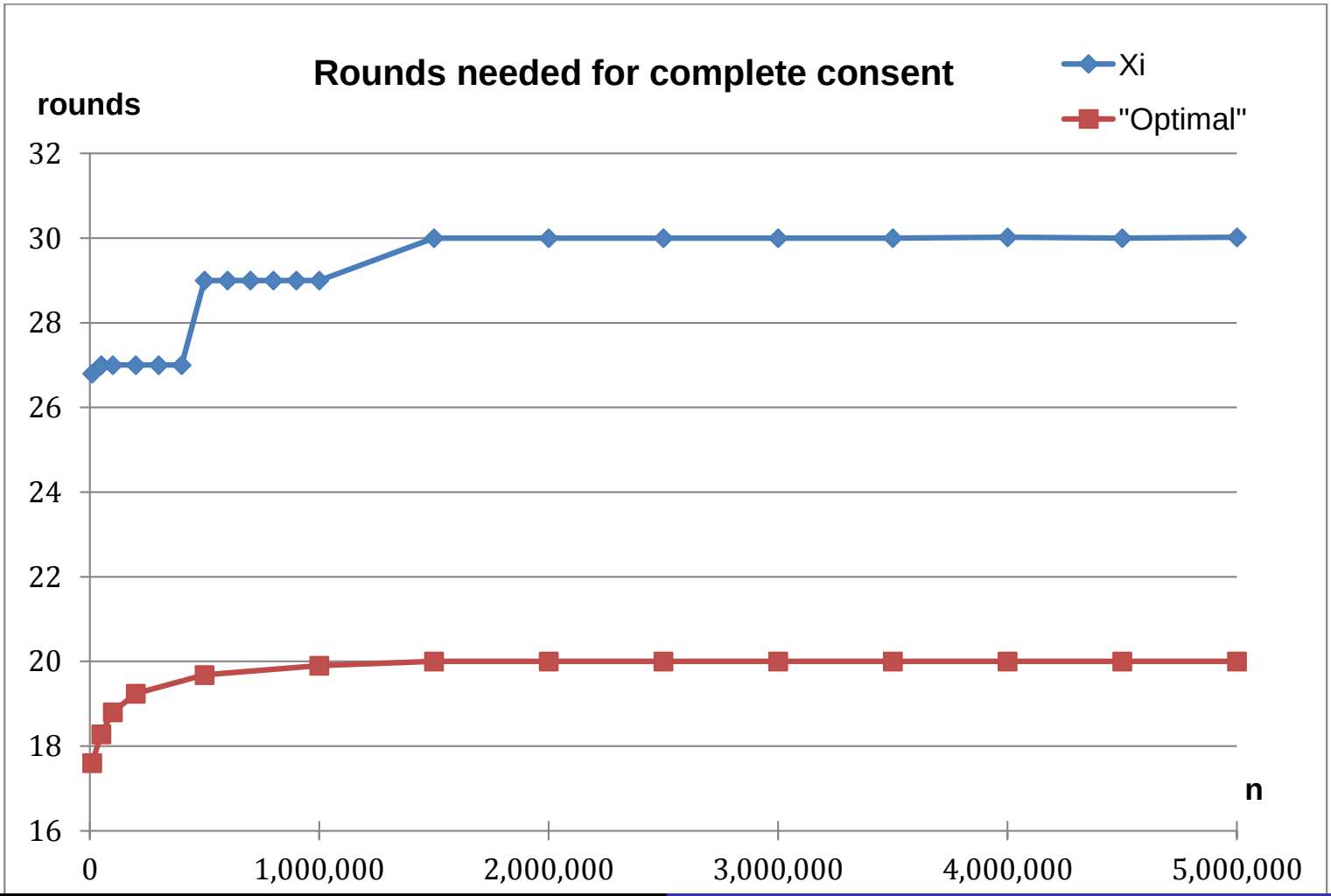
Prikaz rezultatov

Propagacija generacij skozi čas



Prikaz rezultatov

Sinhroniziran model, $k = \ln n$, $\alpha = 1.5$ (worst-case initial coloring)



Oprte sorodne teme

- Robustnost proti zlonamernežem...
- Kako definirati skorajšnji konsenz v velikih omrežjih?
- Algoritem brez mejnih parametrov
- Izboljšana verjetnostna analiza martingalov $\rightarrow$ izboljšana koncentracija rezultatov $\rightarrow$ garancija algoritmov pri razširjenih pogojih
- Pod katerimi pogoji je čas reda $\log \log n$ možen?
- Natančna obravnava ravnovesja med potrebnim pomnilnikom ter časom izvajanja.

UNIVERSITÄT SALZBURG

Oprte sorodne teme

- Robustnost proti zlonamernežem...
- Kako definirati skorajšnji konsenz v velikih omrežjih?
- Algoritem brez mejnih parametrov
- Izboljšana verjetnostna analiza martingalov $\rightarrow$ izboljšana koncentracija rezultatov $\rightarrow$ garancija algoritmov pri razširjenih pogojih
- Pod katerimi pogoji je čas reda $\log \log n$ možen?
- Natančna obravnava ravnovesja med potrebnim pomnilnikom ter časom izvajanja.

UNIVERSITÄT SALZBURG

Oprte sorodne teme

- Robustnost proti zlonamernežem...
- Kako definirati skorajšnji konsenz v velikih omrežjih?
- Algoritem brez mejnih parametrov
- Izboljšana verjetnostna analiza martingalov $\rightarrow$ izboljšana koncentracija rezultatov $\rightarrow$ garancija algoritmov pri razširjenih pogojih
- Pod katerimi pogoji je čas reda $\log \log n$ možen?
- Natančna obravnava ravnovesja med potrebnim pomnilnikom ter časom izvajanja.

UNIVERSITÄT SALZBURG

Oprte sorodne teme

- Robustnost proti zlonamernežem...
- Kako definirati skorajšnji konsenz v velikih omrežjih?
- Algoritem brez mejnih parametrov
- Izboljšana verjetnostna analiza martingalov $\rightarrow$ izboljšana koncentracija rezultatov $\rightarrow$ garancija algoritmov pri razširjenih pogojih
- Pod katerimi pogoji je čas reda $\log \log n$ možen?
- Natančna obravnava ravnovesja med potrebnim pomnilnikom ter časom izvajanja.

UNIVERSITÄT SALZBURG

Oprte sorodne teme

- Robustnost proti zlonamernežem...
- Kako definirati skorajšnji konsenz v velikih omrežjih?
- Algoritem brez mejnih parametrov
- Izboljšana verjetnostna analiza martingalov $\rightarrow$ izboljšana koncentracija rezultatov $\rightarrow$ garancija algoritmov pri razširjenih pogojih
- Pod katerimi pogoji je čas reda $\log \log n$ možen?
- Natančna obravnava ravnovesja med potrebnim pomnilnikom ter časom izvajanja.

UNIVERSITÄT SALZBURG

Oprte sorodne teme

- Robustnost proti zlonamernežem...
- Kako definirati skorajšnji konsenz v velikih omrežjih?
- Algoritem brez mejnih parametrov
- Izboljšana verjetnostna analiza martingalov $\rightarrow$ izboljšana koncentracija rezultatov $\rightarrow$ garancija algoritmov pri razširjenih pogojih
- Pod katerimi pogoji je čas reda $\log \log n$ možen?
- Natančna obravnava ravnovesja med potrebnim pomnilnikom ter časom izvajanja.

UNIVERSITÄT SALZBURG

Problem konsenza
Kaj smo spoznali?

- Problem konsenza je zanimiv in uporaben problem - klasičen problem na področju paralelnih oz. distribuiranih algoritmov.
- Osnovne ideje za reševanje teh in sorodnih problemov so preproste!
- Okvirno smo skicirali primer učinkovite algoritmične rešitve.

UNIVERSITÄT SALZBURG

Problem konsenza
Kaj smo spoznali?

- Problem konsenza je zanimiv in uporaben problem - klasičen problem na področju paralelnih oz. distribuiranih algoritmov.
- Osnovne ideje za reševanje teh in sorodnih problemov so preproste!
- Okvirno smo skicirali primer učinkovite algoritmične rešitve.

UNIVERSITÄT SALZBURG

Problem konsenza
Kaj smo spoznali?

- Problem konsenza je zanimiv in uporaben problem - klasičen problem na področju paralelnih oz. distribuiranih algoritmov.
- Osnovne ideje za reševanje teh in sorodnih problemov so preproste!
- Okvirno smo skicirali primer učinkovite algoritmične rešitve.

UNIVERSITÄT SALZBURG

Hvala za pozornost

Vprašanja!