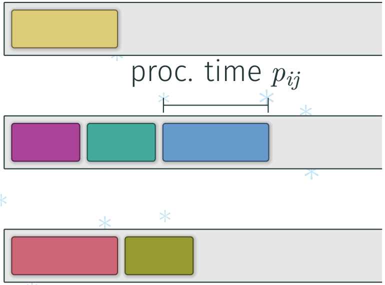
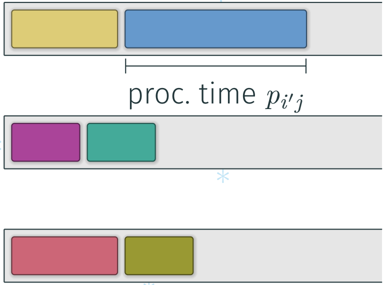
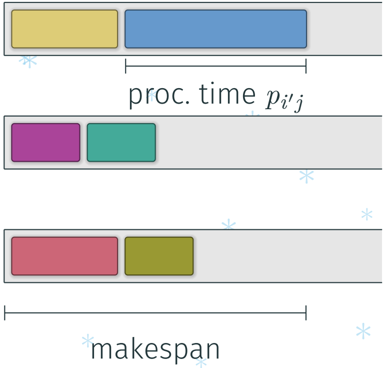
Better Trees for Santa Claus

Étienne Bamas Lars Rohwedder

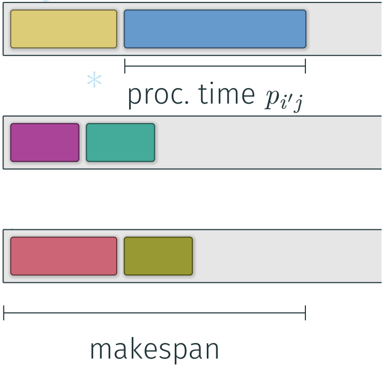
MAPSP'24


proc. time p_{ij}

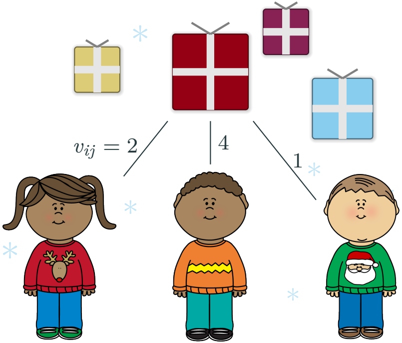

proc. time $p_{i'j}$


proc. time $p_{i'j}$
makespan

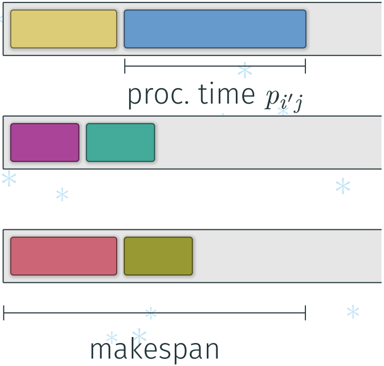
Makespan scheduling



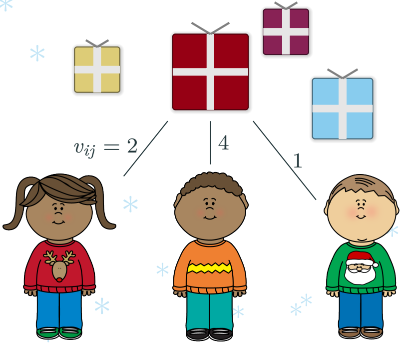
Max-min fair allocation (Santa Claus problem)



Makespan scheduling



Max-min fair allocation (Santa Claus problem)

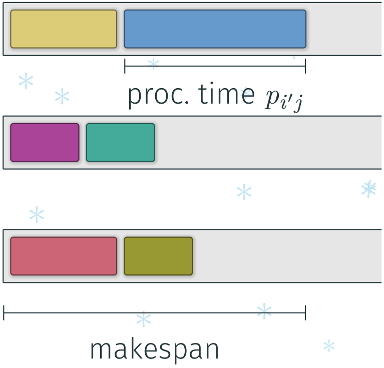


Big open question:

$(2 - 1/\alpha)\mathrm{OPT}$?

$\alpha \cdot \mathrm{OPT}$?

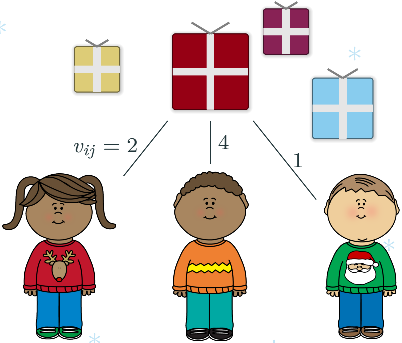
Makespan scheduling



Big open question:

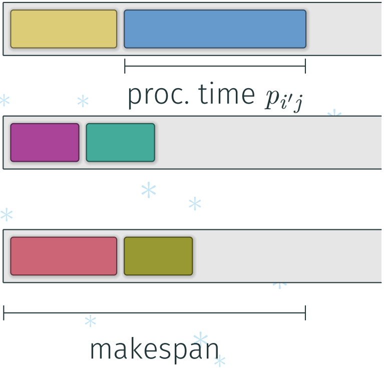
$$(2 - 1/\alpha)\mathrm{OPT}\ ?$$

Max-min fair allocation (Santa Claus problem)

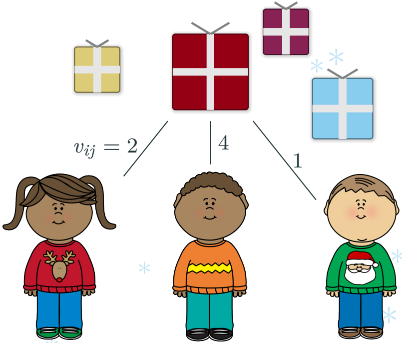


$$\alpha \cdot \mathrm{OPT}\ ?$$

Makespan scheduling



Max-min fair allocation (Santa Claus problem)



Big open question: Bamas, Lindermayr, Megow, **R.**, Schlöter [SODA'24]

$(2 - 1/\alpha)\mathrm{OPT}$? $\Longrightarrow$ $\alpha \cdot \mathrm{OPT}$?

Status of the Santa Claus problem

Open: constant approximation

Chakrabarty, Chuzhoy, Khanna [FOCS'09]: $(\log n)^{O(1)}$-approximation (quasi-polynomial time)

Status of the Santa Claus problem

Open: constant approximation

> **Chakrabarty, Chuzhoy, Khanna [FOCS'09]:** $(\log n)^{O(1)}$-approximation (quasi-polynomial time)

A lot of progress for special cases via configuration LP ...

Problem: configuration LP has high integrality gap in general

Status of the Santa Claus problem

Open: constant approximation

> **Chakrabarty, Chuzhoy, Khanna [FOCS'09]:** $(\log n)^{O(1)}$-approximation (quasi-polynomial time)

A lot of progress for special cases via configuration LP ...

Problem: configuration LP has high integrality gap in general

Idea: focus on special cases with high integrality gap. Only example in literature:

> **Bateni, Charikar, Guruswami [STOC'09]:** $(\log n)^{O(1)}$-approximation for max-min degree arborescence (quasi-polynomial time)

Status of the Santa Claus problem

Open: constant approximation

Chakrabarty, Chuzhoy, Khanna [FOCS'09]: $(\log n)^{O(1)}$-approximation (quasi-polynomial time)

A lot of progress for special cases via configuration LP ...

Problem: configuration LP has high integrality gap in general

Idea: focus on special cases with high integrality gap. Only example in literature:

Bateni, Charikar, Guruswami [STOC'09]: $(\log n)^{O(1)}$-approximation for max-min degree arborescence (quasi-polynomial time)

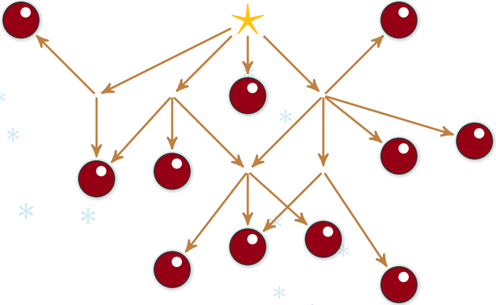
Our contribution: $(\log\log n)^{O(1)}$-approximation for max-min degree arborescence (quasi-polynomial time)

Max-min degree arborescence

Input: directed graph $G = (V, E)$, source ★$\in V$ and sinks ●$\subseteq V \setminus \{s\}$

Output: arborescence $A \subseteq G$ with root ★, leaves in ●, and outdegree k for all non-sink vertices

Goal: maximize k

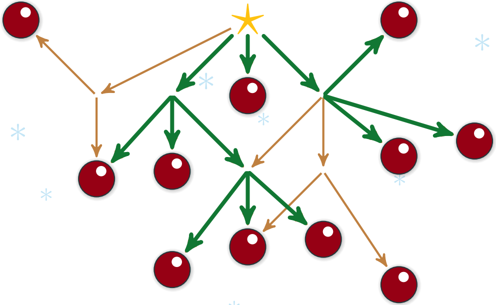


Max-min degree arborescence

Input: directed graph $G = (V, E)$, source ★ $\in V$ and sinks ● $\subseteq V \setminus \{s\}$

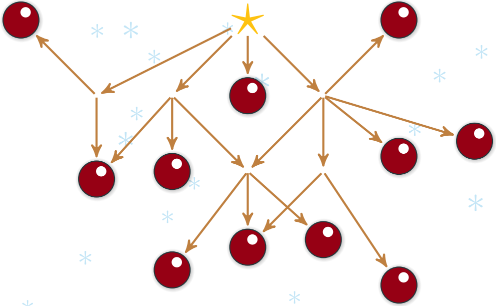
Output: arborescence $A \subseteq G$ with root ★, leaves in ●, and outdegree k for all non-sink vertices

Goal: maximize k



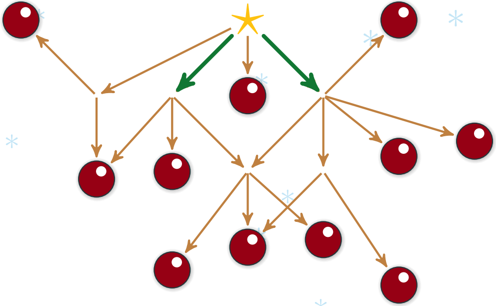
Randomized rounding by Bateni, Charikar, Guruswami

randomly select children of root, then recurse on children...



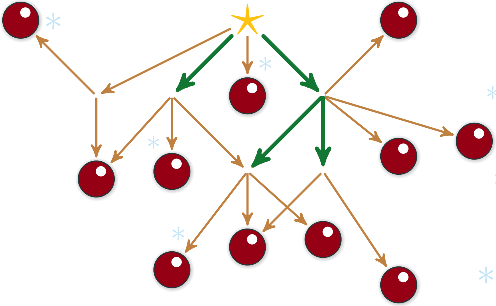
Randomized rounding by Bateni, Charikar, Guruswami

randomly select children of root, then
recurse on children…



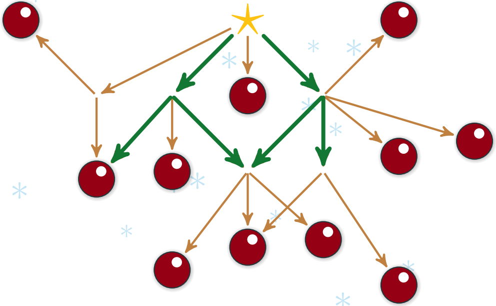
Randomized rounding by Bateni, Charikar, Guruswami

randomly select children of root, then
recurse on children...



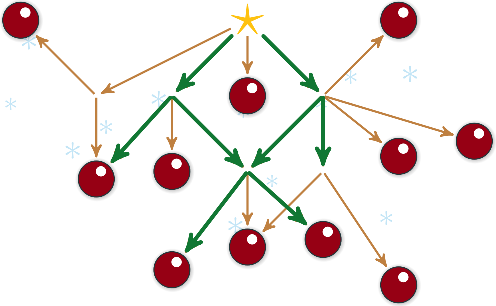
Randomized rounding by Bateni, Charikar, Guruswami

randomly select children of root, then recurse on children…



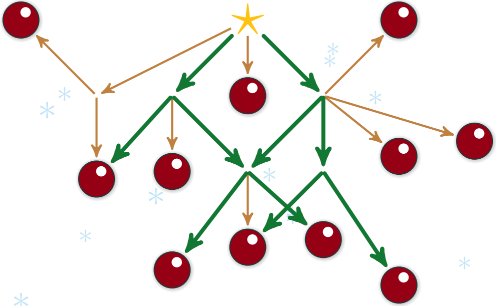
Randomized rounding by Bateni, Charikar, Guruswami

randomly select children of root, then
recurse on children...



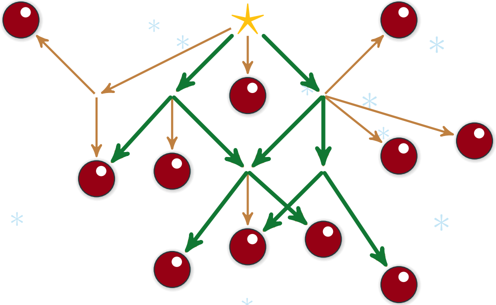
Randomized rounding by Bateni, Charikar, Guruswami

randomly select children of root, then
recurse on children...



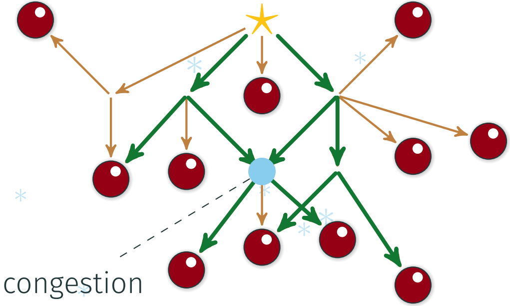
Randomized rounding by Bateni, Charikar, Guruswami

- randomly select children of root, then recurse on children...



Randomized rounding by Bateni, Charikar, Guruswami

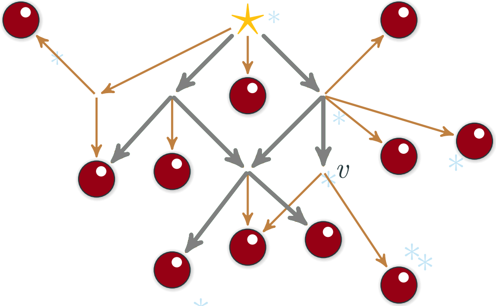
randomly select children of root, then recurse on children...



Bound on congestion can be turned into approximation rate

Randomized rounding by Bateni, Charikar, Guruswami

randomly select children of root, then recurse on children...

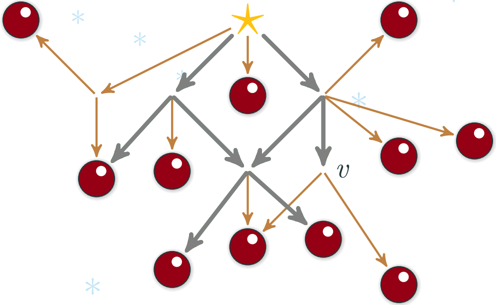


Bound on congestion can be turned into approximation rate

But what distribution to use?

Randomized rounding by Bateni, Charikar, Guruswami

randomly select children of root, then recurse on children...



Bound on congestion can be turned into approximation rate

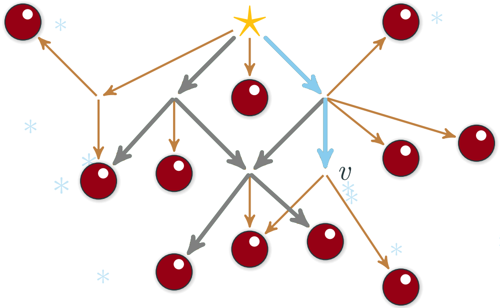
But what distribution to use?

Naive: at vertex v, for $e \in \delta^+(v)$ use probability $x_e/x(\delta^-(v))$.

$\rightsquigarrow$ does not even work for x convex combination of integral solutions...

Randomized rounding by Bateni, Charikar, Guruswami

randomly select children of root, then recurse on children...



Bound on congestion can be turned into approximation rate

But what distribution to use?

Naive: at vertex v, for $e \in \delta^+(v)$ use probability $x_e/x(\delta^-(v))$.
$\rightsquigarrow$ does not even work for x convex combination of integral solutions...

Smart: use Sherali-Adams hierarchy (on naive relaxation) and condition on path to v
$\rightsquigarrow$ congestion $O(\log n)$ w.h.p.

Chernoff bound + union bound over n vertices $\rightsquigarrow$ hard to avoid $O(\log n)$

Chernoff bound + union bound over n vertices $\rightsquigarrow$ hard to avoid $O(\log n)$

Bamas, R. [STOC'23]: $(\log \log n)^{O(1)}$

Chernoff bound + union bound over n vertices $\rightsquigarrow$ hard to avoid $O(\log n)$

Bamas, R. [STOC'23]: $(\log \log n)^{O(1)}$

Idea: obtain $O(\log n)$ congestion through randomized rounding, then to reduce it, remove half of the children of each node.

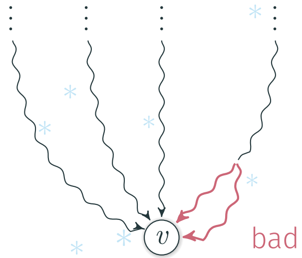
Works if initial solution has "locally low congestion" (new concept).

Chernoff bound + union bound over n vertices $\rightsquigarrow$ hard to avoid $O(\log n)$

Bamas, R. [STOC'23]: $(\log \log n)^{O(1)}$

Idea: obtain $O(\log n)$ congestion through randomized rounding, then to reduce it, remove half of the children of each node.

Works if initial solution has "locally low congestion" (new concept).



Using Lovász Local Lemma (LLL) we can assume that congestion comes from paths that are disjoint on last $O(\log \log n)$ edges*

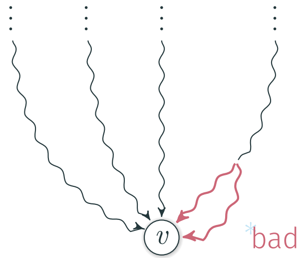
*oversimplifying slightly

Chernoff bound + union bound over n vertices $\rightsquigarrow$ hard to avoid $O(\log n)$

Bamas, R. [STOC'23]: $(\log\log n)^{O(1)}$

Idea: obtain $O(\log n)$ congestion through randomized rounding, then to reduce it, remove half of the children of each node.

Works if initial solution has "locally low congestion" (new concept).



Using Lovász Local Lemma (LLL) we can assume that congestion comes from paths that are disjoint on last $O(\log\log n)$ edges*
$\rightsquigarrow$ congestion reduces by $O(\log n)$ in expectation + strong concentration

---

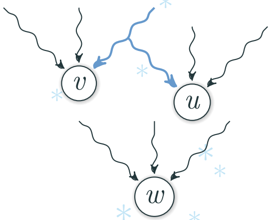
*oversimplifying slightly

Chernoff bound + union bound over n vertices $\rightsquigarrow$ hard to avoid $O(\log n)$

Bamas, R. [STOC'23]: $(\log \log n)^{O(1)}$

Idea: obtain $O(\log n)$ congestion through randomized rounding, then to reduce it, remove half of the children of each node.

Works if initial solution has "locally low congestion" (new concept).



Using Lovász Local Lemma (LLL) we can assume that congestion comes from paths that are disjoint on last $O(\log \log n)$ edges*

$\rightsquigarrow$ congestion reduces by $O(\log n)$ in expectation + strong concentration

Limited dependencies $\rightsquigarrow$ LLL applicable

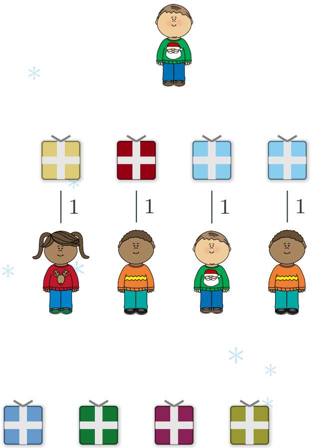
---

*oversimplifying slightly

Future: possible implications for Santa Claus

Algorithm by Chakrabarty, Chuzhoy, Khanna

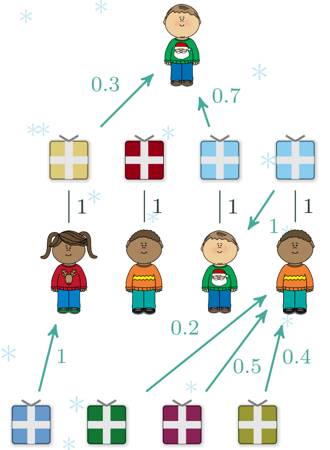
Suppose we build a solution by iteratively growing set of happy children (w.l.o.g. $\Leftrightarrow$ that have value at least 1).



Assume for now that currently every happy child has single (very valuable) gift*

Algorithm by Chakrabarty, Chuzhoy, Khanna

Suppose we build a solution by iteratively growing set of happy children (w.l.o.g. ⇔ that have value at least 1).



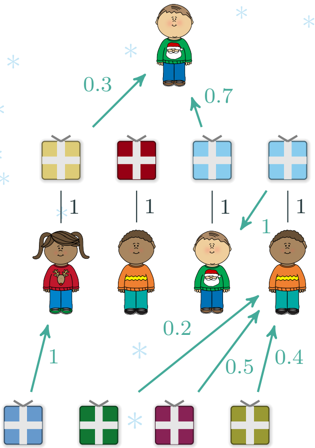
Assume for now that currently every happy child has single (very valuable) gift*

Augment towards solution that makes additional child happy

⇝ is weighted version of max-min degree arborescence

Algorithm by Chakrabarty, Chuzhoy, Khanna

Suppose we build a solution by iteratively growing set of happy children (w.l.o.g. ⇔ that have value at least 1).



Assume for now that currently every happy child has single (very valuable) gift*

Augment towards solution that makes additional child happy
⇝ is weighted version of max-min degree arborescence

still $(\log n)^{O(1)}$-approximation possible, (*) can be established with loss of $(\log n)^{O(1)}$
⇝ $(\log n)^{O(1)}$ approximation