Efficient Use of Exponential Size Linear Programs

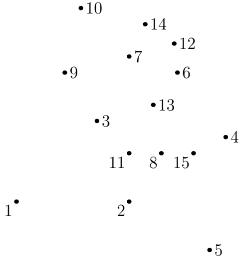
Lukáš Poláček

KTH Royal Institute of Technology, Sweden

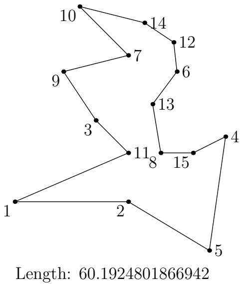
20 March 2015

Traveling Salesman Problem

Suppose Santa Claus wants distribute gifts to children in the shortest possible time: *Traveling Salesman Problem* (TSP).



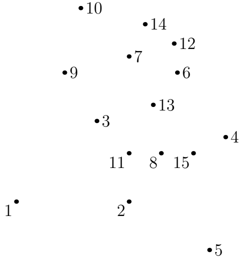
(a) 15 houses to visit



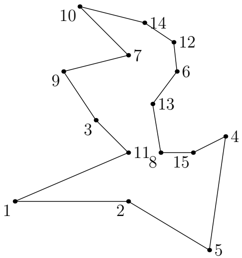
(b) Possible solution

Traveling Salesman Problem

Suppose Santa Claus wants distribute gifts to children in the shortest possible time: *Traveling Salesman Problem* (TSP).

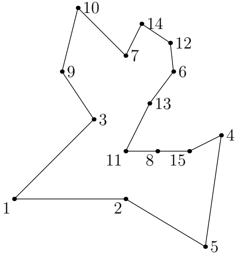


(a) 15 houses to visit



Length: 60.1924801866942

(b) Possible solution



Length: 57.2450308209323

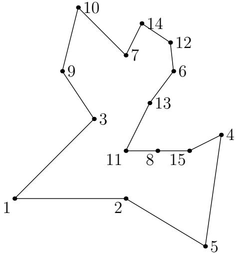
(c) Optimal solution

Optimal Solution

- For 15 points, there are $14! = 87\,178\,291\,200$ different tours.
- The fastest known algorithm [Held-Karp 1962] basically goes through all the possible tours.
- Santa needs to make about 500 000 000 stops. It takes $10^{100\,000\,000}$ years to run the Held-Karp algorithm, even using the fastest computer.

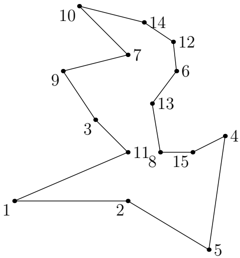
Approximation Algorithms to the Rescue

Approximation: allow suboptimal solutions.



Length: 57.2450308209323

(a) Optimal solution



Length: 60.1924801866942

(b) 5.15% worse

A suboptimal solution is easier to find.

Approximation Algorithms to the Rescue

Christophides algorithm

- At most 50% worse than optimal. Possibly better.
- Only a couple of hours to run on the fastest computer on Earth to visit all children.

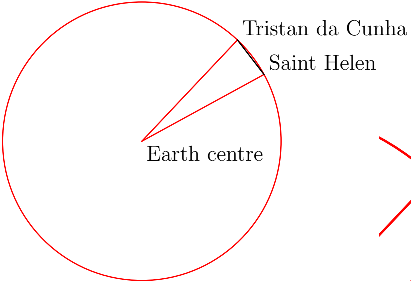
Approximation Algorithms to the Rescue

Arora-Mitchell algorithm

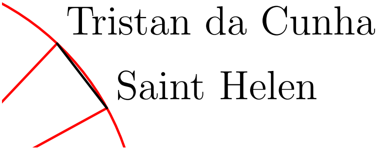
- Works in Euclidean spaces (2D and 3D are Euclidean).
- Can be as close to the optimum as desired, but smaller error leads to worse running time.
- Earth surface is not Euclidean.
- Using a paper map as a 2D space does not work – it's not Euclidean.

Arora-Mitchell Algorithm

- Treat all houses that Santa has to visit as points in 3D.
- Santa has to fly over the surface. Probably not so bad.
- Tristan da Cunha is the most remote place on Earth. Only 0.5% longer route over the surface than under the ground to Saint Helen (2000 km).



(a) Small difference



(b) Closer look

Arora-Mitchell Algorithm

- Aim for 10% error and assume that we lose at most 0.5% by flying over ground.
- Run Arora-Mitchell with 9.45% error ($1.0945 \cdot 1.005 \approx 1.10$).
- Running time is at least

$$n \left(\frac{\sqrt{3}}{0.0945} \right)^{\frac{3}{0.0945^2}}$$

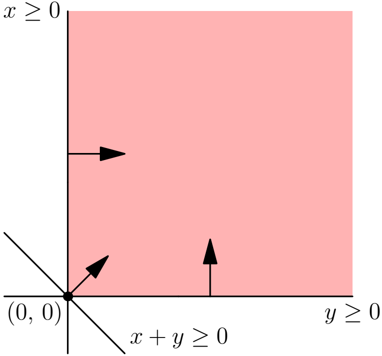
operations, where n is the number of stops. About 10^{1240} times the age of the universe on the fastest computer on Earth.

Heuristics

- In practice, people use *heuristics*: algorithms with weak guarantees.
- For the TSP problem, heuristics often reach 1% error while being very fast.

Linear Programming (LP)

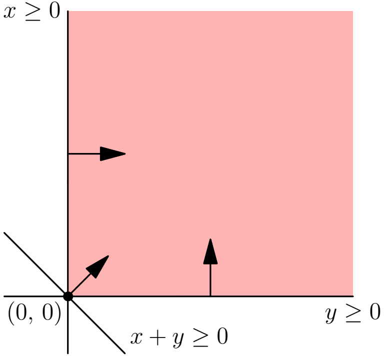
Polyhedron is an intersection of half-planes. Each half-plane is given by an inequality.



(a) $x + y \geq 0$ is redundant

Linear Programming (LP)

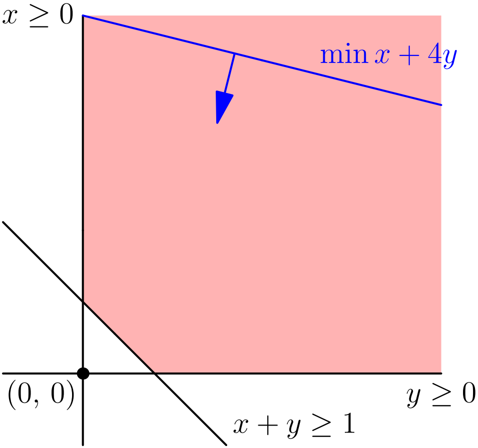
Polyhedron is an intersection of half-planes. Each half-plane is given by an inequality.



(a) $x + y \geq 0$ is redundant (b) $x + y \geq 1$ cuts away a piece

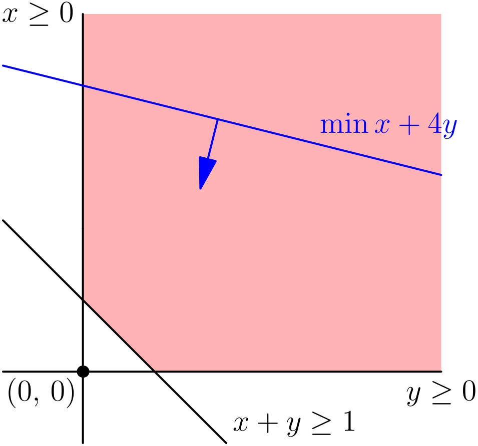
Linear Programming (LP)

Optimize a linear function over the polyhedron.



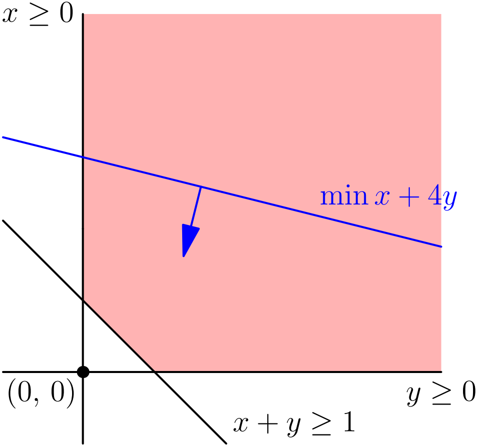
Linear Programming (LP)

Optimize a linear function over the polyhedron.



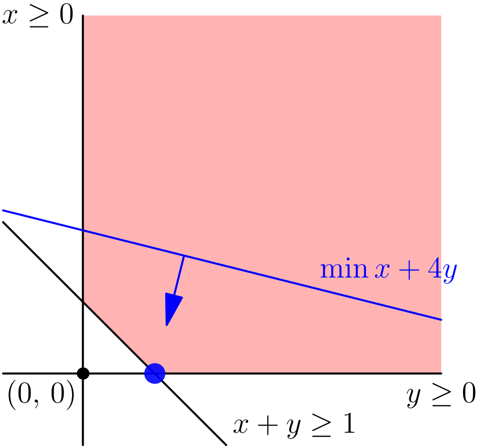
Linear Programming (LP)

Optimize a linear function over the polyhedron.



Linear Programming (LP)

Optimize a linear function over the polyhedron.

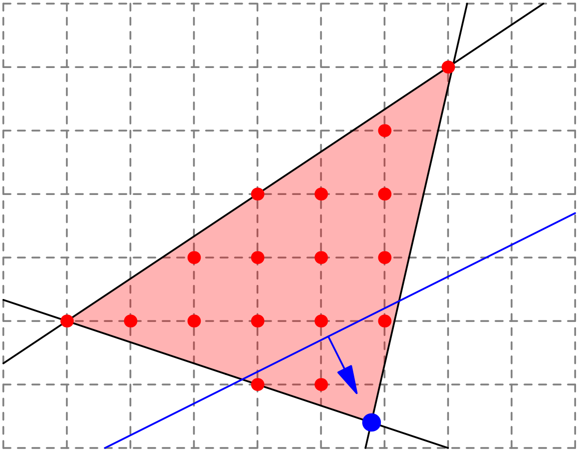


Linear Programming (LP)

- One of the central problems in computer science.
- Simplex method was developed in 1947 by Dantzig and it is still being used. One of the most important algorithms of the 20th century.
- Only in 2001 we understood why it performs well (polynomial time) most of the time [Spielman and Teng].

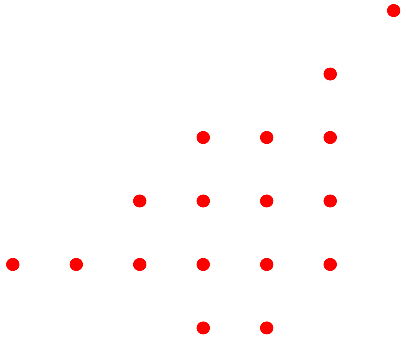
Integer Programming

We usually want integer solutions.



Integer Programming

Easier to optimize over a polyhedron than integer points. Very hard in many dimensions. Less "structure".



This Thesis

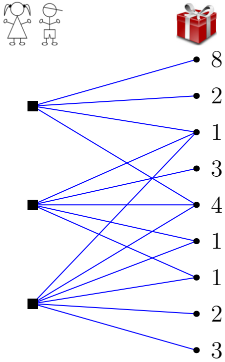
- We use linear programs of exponential size (configuration LP).
- Exponential size is too big but we solve them efficiently in polynomial time.
- Problems studied:
  - Restricted Max-min fair allocation (The Santa Claus problem): faster algorithm
  - Restricted Maximum budgeted allocation (MBA): better approximation ratio and some *negative results*

Santa Claus Problem (Restricted Version)

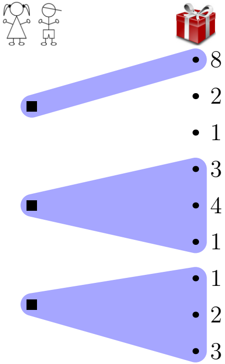
Max-min fair allocation is the Santa Claus problem – Santa Claus has gifts for children. Goal: make the least happy child as happy as possible.

Santa Claus Problem (Restricted Version)

Max-min fair allocation is the Santa Claus problem – Santa Claus has gifts for children. Goal: make the least happy child as happy as possible.



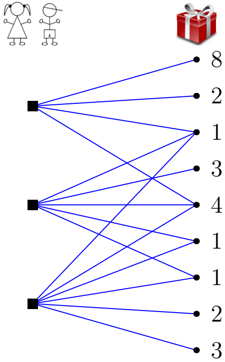
(a) Input



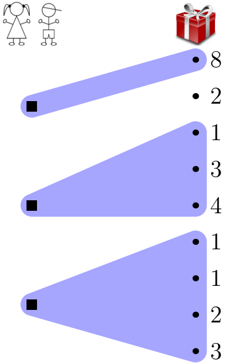
(b) A solution of value 6

Santa Claus Problem (Restricted Version)

Max-min fair allocation is the Santa Claus problem – Santa Claus has gifts for children. Goal: make the least happy child as happy as possible.



(a) Input



(b) A solution of value 7

General and Restricted Version

- In general, valuation of a gift can be different for different children – general version.
- We focus on the restricted version.

Previous Results

Hardness

- It is *NP*-hard to approximate the restricted version within a factor of $1/2$ [Bezáková and Dani 2005].

Previous Results

Hardness

- It is *NP*-hard to approximate the restricted version within a factor of $1/2$ [Bezáková and Dani 2005].

Algorithms

- Restricted version: Constant approximation in polynomial time [Feige 2008] and [Haeupler, Saha and Srinivasan 2010]. The constant is small and not explicitly stated.

Estimation Versus Approximation

[Asadpour, Feige and Saberi 2008] Strange situation for the restricted version – easy to estimate, hard to approximate.

- The integrality gap of the configuration LP is at least $1/4$ – we can estimate the value of the solution in polynomial time.
- One can solve the LP only with an error – $1/(4+\varepsilon)$-estimation.
- Rounding runs in exponential time $O(2^n)$.

Assignment LP

- Gifts: $\mathcal{I}$
- Children: $\mathcal{P}$
- T is the minimal value given to every child.
- 1st constraint – every child gets value at least T.
- 2nd constraint – every gift is allocated at most once.

$$
\begin{array}{rlll}
\text{max} & T & & \\
\text{subject to} & \sum_{j \in \mathcal{I}} x_{ij} p_{ij} & \geq T & \forall i \in \mathcal{P} \\
& \sum_{i \in \mathcal{P}} x_{ij} & \leq 1 & \forall j \in \mathcal{I} \\
& x_{ij} & \geq 0 & \forall i \in \mathcal{P}, \forall j \in \mathcal{I}
\end{array}
$$

Assignment LP

- With 100 kids and 99 gifts of value 1, one child leaves empty-handed.
- Let $x_{ij} = 1/100$ for all i and j, and $T = 99/100$.
- Optimal integer value is 0, while the assignment LP claims 0.99: unbounded *integrality gap*.
- Need a stronger linear program.

Configuration LP

- Think of T as the optimal value.
- We have a variable $x_{iC} \geq 0$ for each $C \in \mathcal{C}(i, T)$, where $\mathcal{C}(i, T)$ are sets of gifts that give value at least T to child i.
- 1st constraint – every child gets value at least T.
- 2nd constraint – every gift is allocated at most once.

$$\sum_{C \in \mathcal{C}(i,T)} x_{iC} \geq 1 \qquad i \in \mathcal{P}$$

$$\sum_{i,C:j \in C} x_{iC} \leq 1 \qquad j \in \mathcal{I}$$

Configuration LP

- If there is a solution such that every child gets value at least T, the LP is feasible.
- Exponentially many variables but can be solved in polynomial time to within some $\varepsilon > 0$ using a separation oracle for the dual [Bansal and Sviridenko 2006].
- The LP is denoted by $CLP(T)$.

Our Result

Theorem

For any $\epsilon \in (0, 1]$, we can find a $\frac{1}{4+\epsilon}$-approximate solution to the restricted Santa Claus problem in time $n^{O\left(\frac{1}{\epsilon}\log n\right)}$.

Algorithm Overview

- Our algorithm
  - either finds a solution of value at least T/α by augmenting trees,
  - or certifies that $CLP(T)$ is not feasible

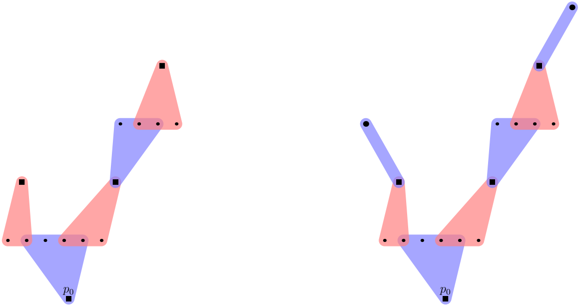
  for $\alpha > 4$.
- Use binary search to find the optimal value of T.
- No need to solve the configuration LP.

Hypergraph Matching

- We want to match gifts to a child – matchings in hypergraphs.
- All edges contain gifts of total value at least T/α, where $\alpha = 4 + \varepsilon$. If we find a perfect matching, all children get gifts of value at least T/α.

Alternating Tree

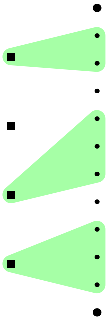
- Alternating trees are similar to alternating paths used for matchings in graphs.
- A good (augmenting) tree increases the size of the matching by one.



(a) Alternating tree (b) Augmenting tree

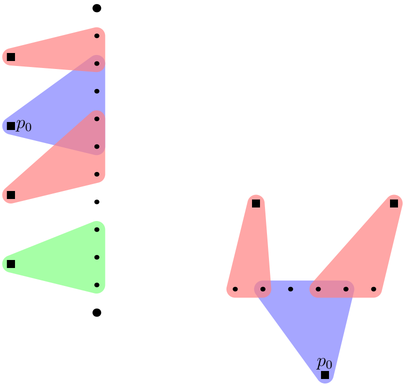
Example of an Algorithm Execution

Legend:
Edge in matching
Blocking edge
Add-edge



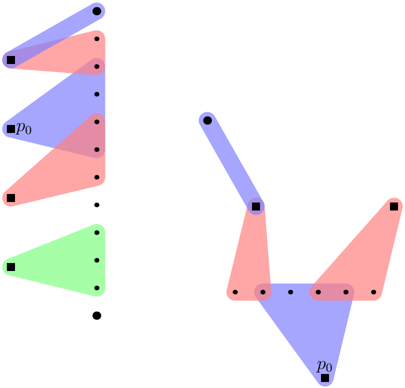
Example of an Algorithm Execution

Legend:
Edge in matching
Blocking edge
Add-edge



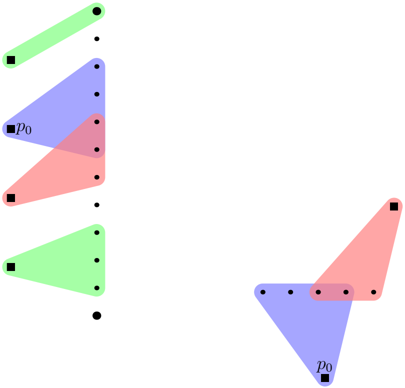
Example of an Algorithm Execution

Legend:
Edge in matching
Blocking edge
Add-edge



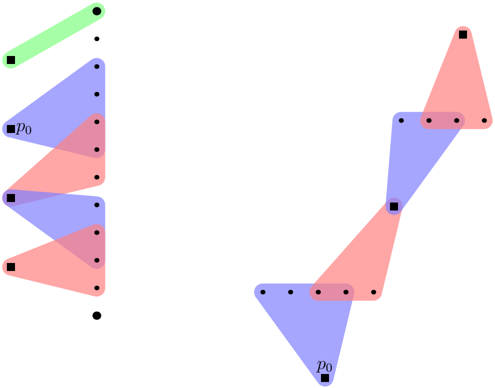
Example of an Algorithm Execution

Legend:
Edge in matching
Blocking edge
Add-edge



Example of an Algorithm Execution

Legend:
Edge in matching
Blocking edge
Add-edge



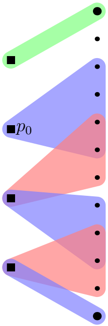
Example of an Algorithm Execution

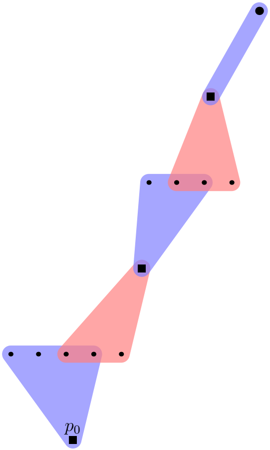
Legend:

Edge in matching

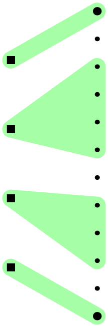
Blocking edge

Add-edge





Example of an Algorithm Execution



Legend:

Edge in matching

Blocking edge

Add-edge

Finding the Right Order

- [AFS08] pick edges in any order – running time is $O(2^n)$.
- We introduce distance from the root. Always add an edge closest to the root.
- We can bound the "height" of the alternating tree to $O(\log n)$.
- This change decreases the running time dramatically to $n^{O(\log n)}$.

Summary

Theorem

For any $\epsilon \in (0, 1]$, we can find a $\frac{1}{4+\epsilon}$-approximate solution to restricted Santa Claus problem in time $n^{O\left(\frac{1}{\epsilon}\log n\right)}$.

We achieved this by

- changing the order of adding edges to the tree,
- analysis using the dual of the configuration LP.

Side-effect: no need to solve the LP.

New Algorithm and Open Problems

- [Annamalai, Kalaitzis and Svensson 2014] provide a $1/12.3$-approximation in polynomial time. Closer, but still far from $1/4$.
- Is estimation easier than approximation?

Maximum budgeted allocation

Advertisers want to buy advertisement banners. They have limited budgets.

Maximum budgeted allocation

Optimal solution. Nike only pays \$1000 for banners of value $700 + 500 = 1200$, because of its limited budget.

Maximum budgeted allocation (restricted version)

- Advertisers $\mathcal{P}$ (also called players): advertiser i has a budget of B_i.
- Advertisement banners $\mathcal{I}$ (also called items).
- Advertiser i is willing to pay at most $p_{ij} \leq B_i$ for banner j.

Maximum budgeted allocation (restricted version)

- Advertisers $\mathcal{P}$ (also called players): advertiser i has a budget of B_i.
- Advertisement banners $\mathcal{I}$ (also called items).
- Advertiser i is willing to pay at most $p_{ij} \leq B_i$ for banner j. Restricted version: $p_{ij} \in \{0, p_j\}$.

Maximum budgeted allocation (restricted version)

- Advertisers $\mathcal{P}$ (also called players): advertiser i has a budget of B_i.
- Advertisement banners $\mathcal{I}$ (also called items).
- Advertiser i is willing to pay at most $p_{ij} \leq B_i$ for banner j. Restricted version: $p_{ij} \in \{0, p_j\}$.
- We want to allocate the advertisement banners (items) to advertisers (players) such that we maximize the total revenue. Advertiser i gets set $\mathcal{C}_i$. The total revenue is

$$\sum_{i \in \mathcal{P}} w_i(\mathcal{C}_i)$$

where

$$w_i(\mathcal{C}_i) = \min \left\{ \sum_{j \in \mathcal{C}_i} p_{ij}, B_i \right\}$$

Previous and Our Results

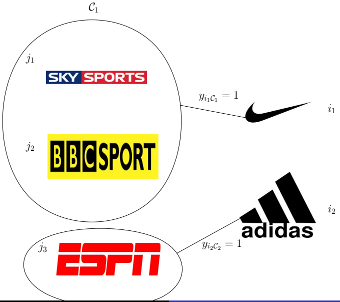
- [Chakrabarty and Goel, 2008] and [Srinivasan, 2008]: approximation ratio $\frac{3}{4}$ using assignment LP.
- [Chakrabarty and Goel, 2008] proved it is NP-hard to approximate within a factor better than $\frac{15}{16}$.

Our Main Theorem

There is a $(3/4 + c)$-approximation algorithm for the restricted MBA.

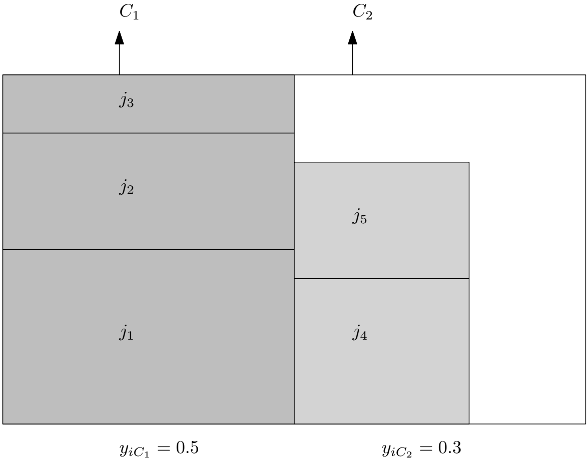
Configuration LP

$$
\begin{array}{llll}
\max & \displaystyle\sum_{i\in\mathcal{P}}\sum_{\mathcal{C}\subseteq\mathcal{I}} w_i(\mathcal{C})y_{i\mathcal{C}} & & \\
\text{subject to} & \displaystyle\sum_{\mathcal{C}\subseteq\mathcal{I}} y_{i\mathcal{C}} & \leq 1 & \forall i\in\mathcal{P} \\
 & \displaystyle\sum_{i\in\mathcal{P},\mathcal{C}\subseteq\mathcal{I}:j\in\mathcal{C}} y_{i\mathcal{C}} & \leq 1 & \forall j\in\mathcal{I} \\
 & y_{i\mathcal{C}} & \geq 0 & \forall i\in\mathcal{P},\forall\mathcal{C}\subseteq\mathcal{I}
\end{array}
$$



Configuration LP

Solution to configuration LP for one player.



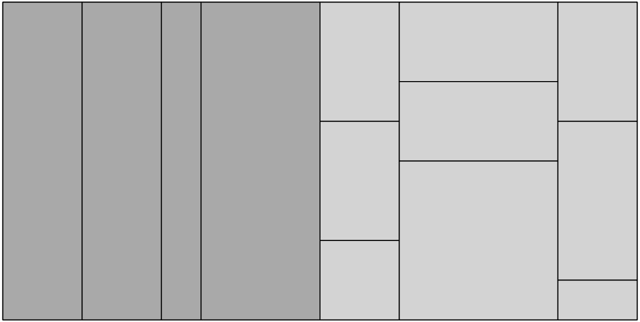
Our approach

- Solve the configuration LP. Convert the solution y to a solution x to the assignment LP.
- Use an algorithm inspired by the classical Shmoys-Tardos algorithm. In many cases it leads to a $(3/4 + c)$-approximation.
- Solve the rest of the cases with a different algorithm using y.

We identify the hard LP solutions for our assignment LP algorithm.
Well-structured solutions:

Big configurations sum up to $1/2$

Small configurations sum up to $1/2$



Recovering the Value of Small Items

- Negatively correlated rounding: For any two players i, i' the events of i and i' being assigned a big item are negatively correlated.
- So that small items have enough free players to be allocated to.
- We recover the full value of big items and a $9/16$-fraction of small items. More than $3/4$ in total.

Other Smaller Results

- A $(3/4 + c)$-algorithm for the graph variant, where each item can only be assigned to 2 players.
- Improved integrality gap example for configuration LP in the general case.
- NP-hardness results for both variants studied.

Open Problems

- Can we close the gap between $3/4 + c$ algorithm and 0.833 integrality gap upper-bound for the restricted case?
- Is there an algorithm having a substantially better than 3/4-approximation algorithms.

Thank you!

Disclaimer: No children nor Santa Claus were harmed during this research.

Thank you!

Disclaimer: No children nor Santa Claus were harmed during this research.

Time for

Photo by NASA's Marshall Space Flight Center.