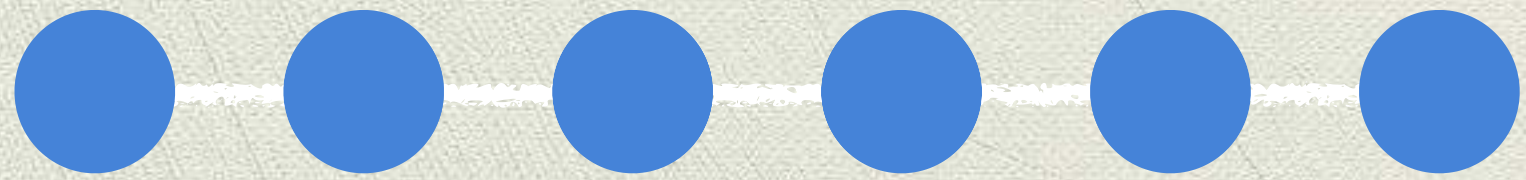
Pareto-Optimal Allocation of Indivisible Goods with Connectivity Constraints

Ayumi Igarashi and Dominik Peters
University of Kyushu University of Oxford

AI³, Stockholm, 2018-07-14

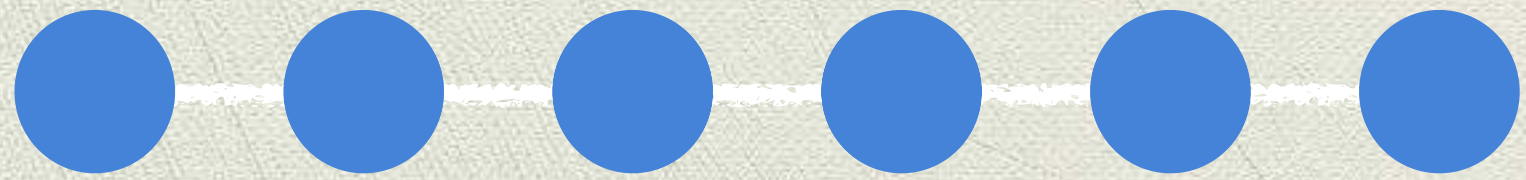
Allocation of a Graph

- Finite set of indivisible items
- Agents have additive preferences over bundles of items
- Goal: Allocate items to agents
- Items are arranged in a graph, only allowed to hand out *connected* bundles



Previously..

- *Fair Division of a Graph*
  IJCAI-17, *Bouveret, Cechlárová, Elkind, Igarashi, and P*.
- NP-complete to decide existence of envy-free or proportional allocations, even on a path
- Tractable if there are few player types
- There always exists an MMS allocation if graph is a tree

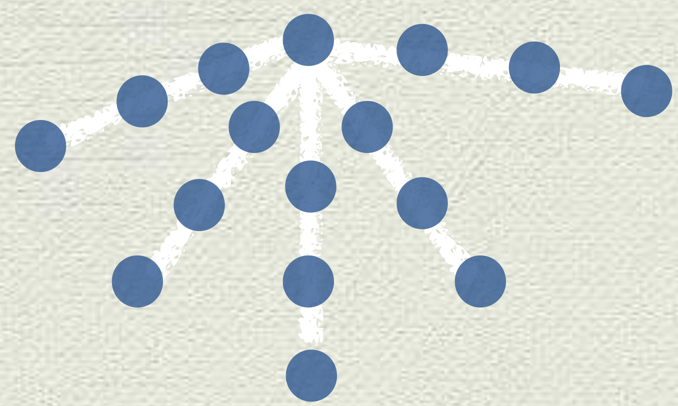


Pareto-Optimality

- A connected allocation is PO if no other connected allocation makes someone better off and no-one worse off.
- Without connectivity, standard way of achieving this: hand each item to agent preferring it most
- MMS exists on trees, so PO + MMS exists. Can we find it efficiently?
- Wait a minute, can we find PO efficiently? we allow 0

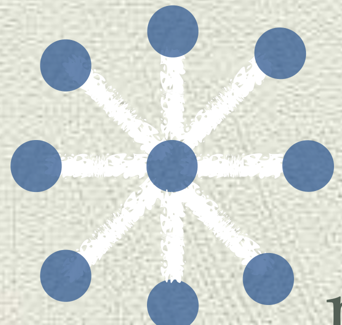
Finding some PO allocation

On paths, version of
serial dictatorship
gives a PO allocation
(but welfare max. is hard)



On general trees,
NP-hard (under Turing red.)
to produce any PO allocation
reduction from X3C via perfection
pathwidth 3, diameter 7

=> welf. max. hard!

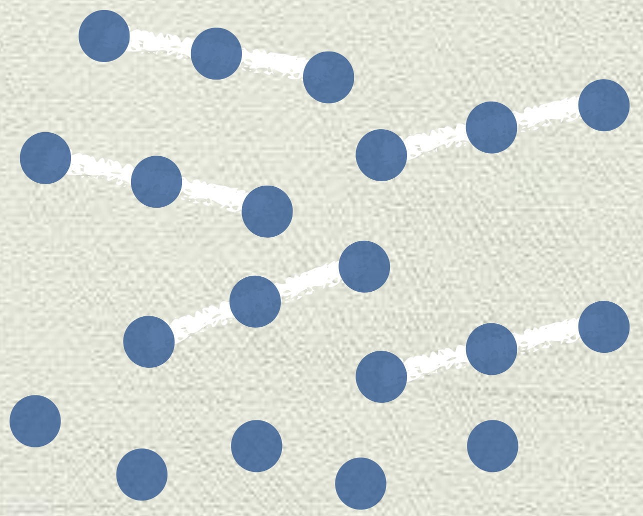


On stars, can use
matching to maximise
utilitarian welfare

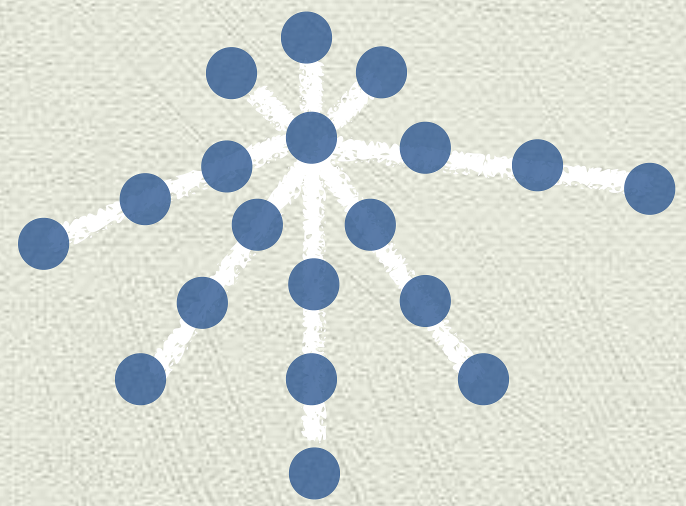
How to Show NP-Hardness

- PO always exists — so how to show hardness?
- Use technique from hedonic games
  H. Aziz, F. Brandt, and P. Harrenstein.
  Pareto optimality in coalition formation. *GEB (2013)*
- Suppose we can show that
  it is NP-complete to decide whether there is an allocation which is *perfect*.
- Then it is NP-hard under Turing reductions to find a PO allocation.

Forests to Trees



simple to find reduction
showing hardness for *forest*
(each player can only get one component)



try to make reduction into a star
— but center player can get two components!

solution: everything x2, then make star
— center player can only mess up one of the copies

MMS

Fair Division of a Graph [IJCAI-17]

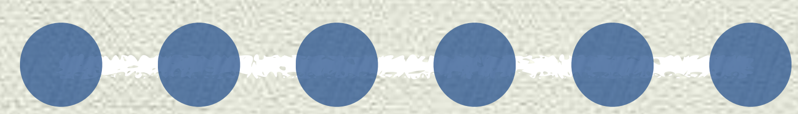
- The maximin share of a player is

$$\mathrm{mms}_i(I) = \max_{(P_1,\ldots,P_n)\in\Pi_n} \min_{j\in[n]} u_i(P_j).$$

- Intuition: Cut into n pieces, choose last.
- Maximise over connected partitions only => MMS values smaller than normal
- Adaptation of a moving knife protocol produces an allocation where every player receives at least their MMS share.

PO + MMS

MMS always exists on trees.
Pareto-improvements preserve MMS.
=> PO + MMS exists on trees.

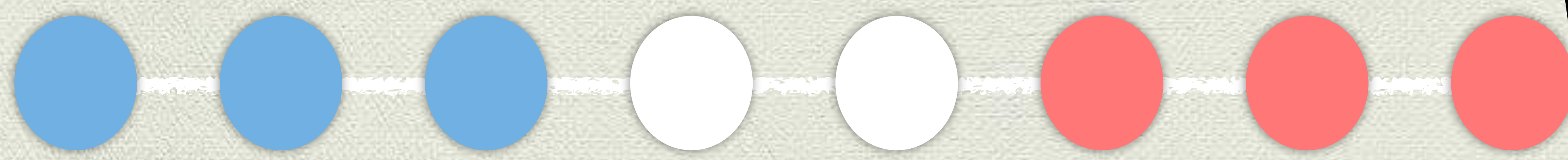


Finding a PO + MMS allocation
is NP-hard on a path

Proof: take forest reduction,
connect into path,
add dummy players
whose MMS guarantee
separate the pieces

remains hard for $\alpha \cdot$MMS
for fixed $\alpha > 0$

PO + EF1



	0	0	0	1	1	0	0	0
	1	1	1	0	0	1	1	1

$\in \Sigma_2^p$

On paths, NP-hard to decide whether a PO + EF1 allocation exists

there is bigger example where 1's form intervals

Future Directions

- Same thing for chores
- Restricted utility classes
- Welfare maximisation (approximations)
- Local envy-freeness: only envy bundles next to yours
- Approximate efficiency? Weaker concepts than PO?