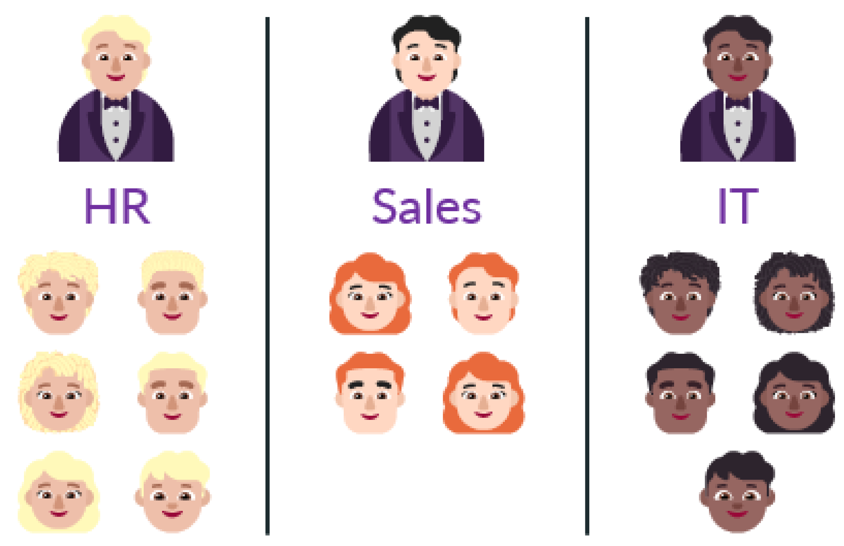
Reachability of Fair Allocations via Sequential Exchanges

Ayumi Igarashi — University of Tokyo
Naoyuki Kamiyama — Kyushu University
Warut Suksompong — National U. of Singapore
Sheung Man Yuen — National U. of Singapore

Fair Division of Indivisible Goods

- The study of allocating goods fairly among competing agents.
- **Example.** A company wishes to allocate its employees to different departments in fair manner.



agent: head of department

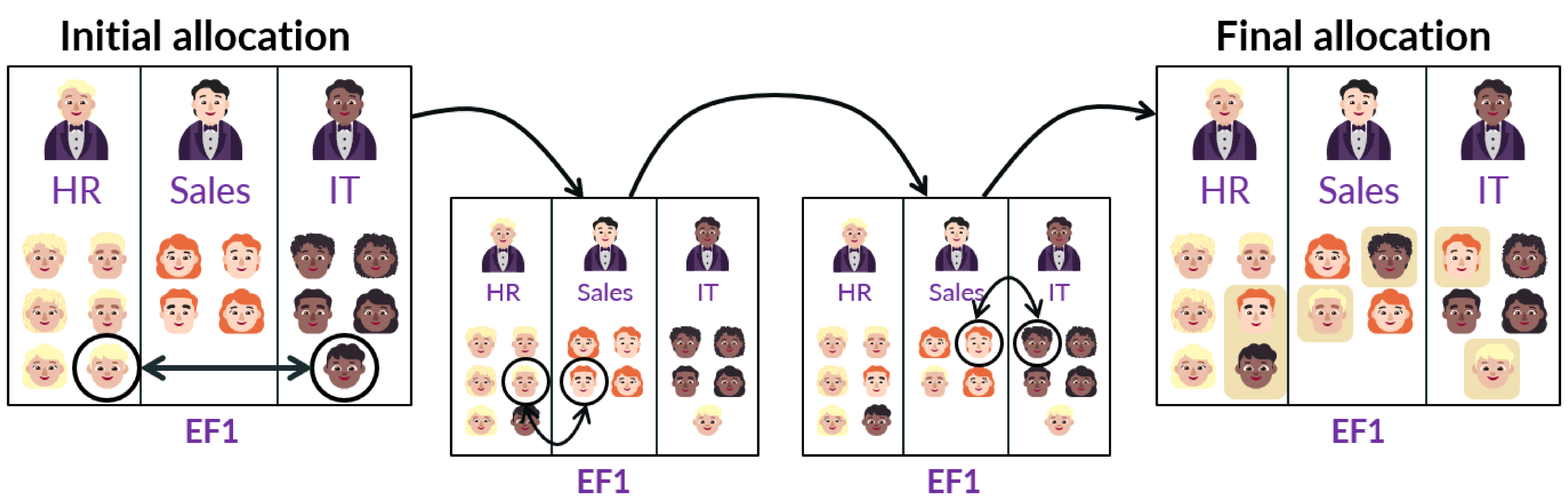
good: employee

Envy-Freeness up to One Good (EF1)

- We consider an allocation fair if it is envy-free up to one good (EF1).
- A head of department is only allowed to envy another department if the envy can be eliminated by removing an employee from that department.
- An EF1 allocation always exists [Lipton et al. '04].

Reachability of EF1 Allocations

- We take a dynamic approach.
- An EF1 allocation is already given, but the CEO of the company is unhappy about the productivity of the employees!
- The CEO wants to redistribute the employees across the departments to increase productivity; she decides on a desired final allocation.



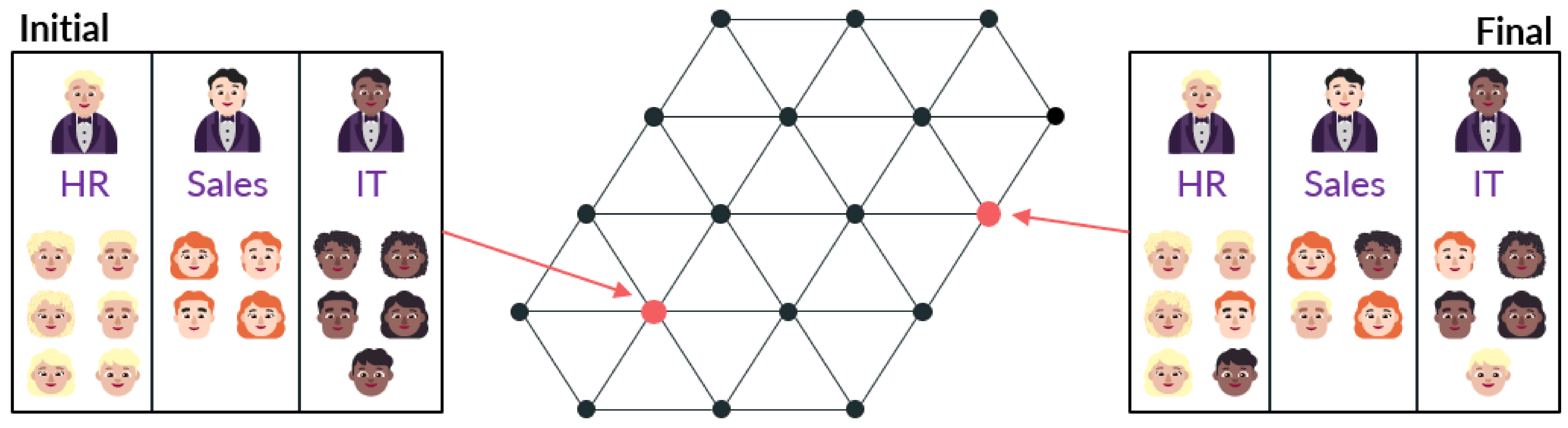
- **Constraint #1.** Every month, two employees from different departments will be selected to exchange positions; the desired final allocation will hopefully be reached after some number of months.
  - Performing the entire redistribution at once instead may excessively disrupt operations.
- **Constraint #2.** EF1 must be maintained throughout the whole process.
  - Otherwise, some head of department would not be very happy!
- **Question.** Can we always start with an initial allocation and reach the desired final allocation in this manner?

Reconfiguration

- Reachability problems are also known as reconfiguration problems.
- **Other Examples.**
  - Minimum spanning tree [Ito et al. '11]
  - Graph coloring [Johnson et al. '16]
  - Perfect matching [Bonamy et al. '19]
  - Voting [Obraztsova et al. '13; Obraztsova et al. '20]

Model: Exchange Graph

- **Vertices:** All allocations with the same size vector.
- **Edges:** Two vertices are adjacent iff they can be reached via an exchange.

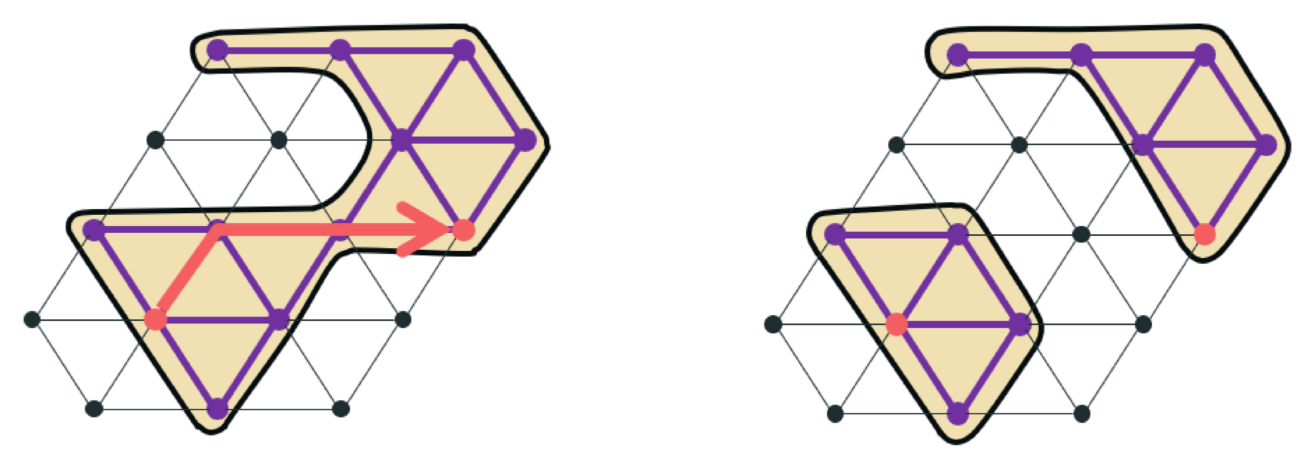


- **Theorem 1.** Computing the distance between two allocations on the exchange graph is NP-hard.

  Proof. Reduce from DIRECTED TRIANGLE PARTITION.

Connectivity of EF1 Exchange Graph

- **EF1 Exchange Graph:** The subgraph induced by all EF1 allocations.



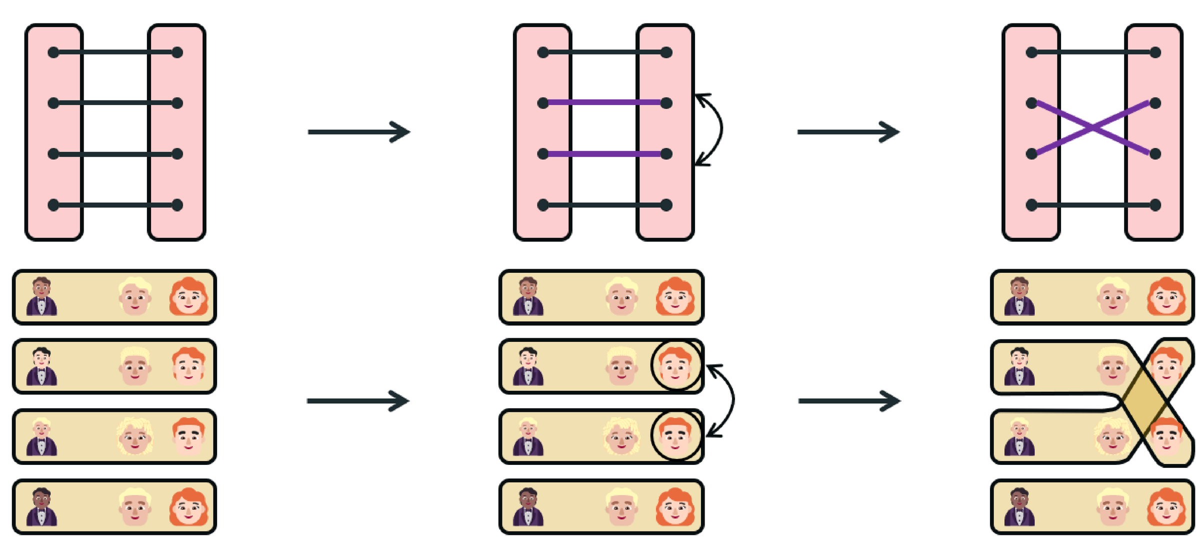
EF1 exchange graph is connected — EF1 exchange graph is not connected

- **Question.** Is the EF1 exchange graph always connected?

utilities	general	identical	binary	identical binary
two agents	✗	✓	✓	✓
$\geq$ three agents	✗ [Thm 2]	✗	✗	✓

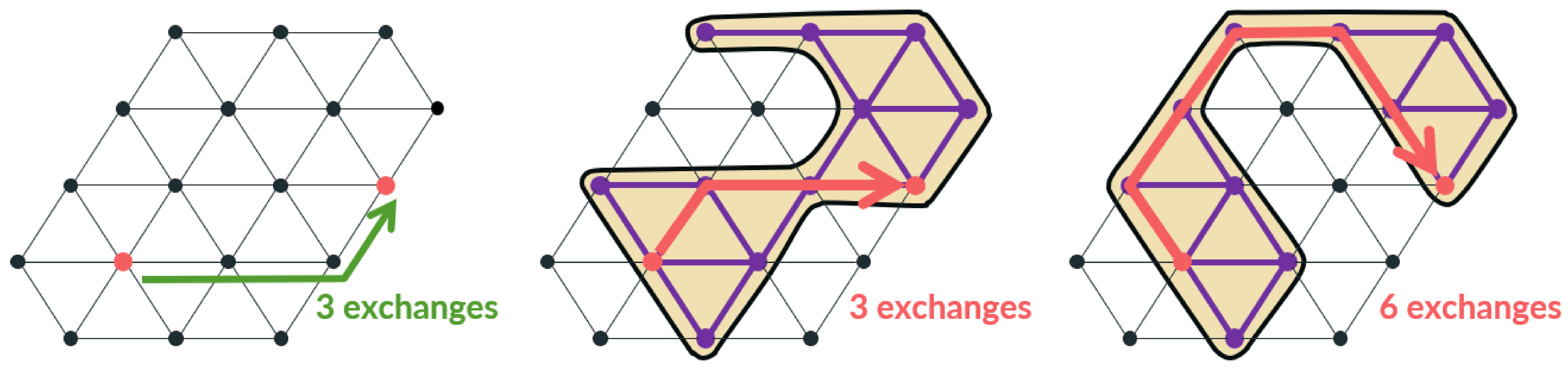
- **Theorem 2.** Determining the existence of an EF1 exchange path between two EF1 allocations is PSPACE-complete.

  Proof. Reduce from PERFECT MATCHING RECONFIGURATION [Bonamy et al. '19].



Optimality of EF1 Exchange Path

- Now, only consider instances where the EF1 exchange graph is connected.
- **Optimal EF1 Exchange Path:** An EF1 exchange path with the same number of exchanges as an exchange path without EF1 constraints.



Exchange graph — Optimal EF1 exchange path exists — Optimal EF1 exchange path does not exist

- **Question.** Does there always exist an optimal EF1 exchange path between two given allocations?

utilities	general	identical	binary	identical binary
two agents	✗	✓	✓	✓
$\geq$ three agents	✗ [Thm 3]	✗ [Thm 3]	✗	✗

- **Theorem 3.** Determining the existence of an optimal EF1 exchange path between two EF1 allocations is NP-hard, even for identical utilities.

  Proof. Reduce from PARTITION.

東京大学 THE UNIVERSITY OF TOKYO
九州大学 KYUSHU UNIVERSITY
NUS National University of Singapore