Sequential and Swap Mechanism for Public Housing Allocation with Quotas and Neighbourhood-Based Utilities

N. Gross–Humbert, N.Benabbou, A. Beynier and N. Maudet

LIP6, Sorbonne Université

Model

➤ **The model contains:**

- A set $\mathcal{N}$ of $N \in \mathbb{N}$ agents divided into $k \in \mathbb{N}$ subsets $T_1, T_2, ...T_k$ called *types*
- A set $\mathcal{M}$ of $M \in \mathbb{N}$ items divided into $l \in \mathbb{N}$ subsets $B_1, B_2, ...B_l$ called *blocks*
- A matrix $\lambda \in \mathbb{N}^{[\![1;k]\!] \times [\![1;l]\!]}$ of *quotas*

➤ An **allocation** A is a function that maps every agent i to a set $A(i)$ of items.
An allocation is **valid** if and only if:

- each agent receives at most one item
- agents do not share item
- all item are assigned
- quotas are respected: for each type p and each block q, the number of items from the block q assigned to agents of type p is at most $\lambda_{p,q}$

Utility

➤ **Utility function**
In an allocation A, the utility $u_i(A)$ of agent i is:

$$u_i(A) = u_i^I(A) + \varphi \times u_i^N(A)$$

Where:

- $u_i^I(A) \in [0;1]$ is the utility that the agent i has for the item received in the allocation A.
- $u_i^N(A) \in [0;1]$ is the utility that the agent i receives from its neighbours, and is equal to the proportion of agents of the same type as i who have been allocated item of the same block as i. More formally:

$$u_i^N(A) = \frac{\sum_{j \in \mathcal{N}:\, A(j) \in \mathcal{B}(A(i))} \mathbb{I}(\mathcal{T}(i), \mathcal{T}(j))}{|\mathcal{B}(A(i))|}$$

  Where $\mathcal{T}(i)$ is the type of i and $\mathcal{B}(A(i))$ is the block of the item allocated to i.

- If no item is assigned to an agent, the utility for this agent is 0.

➤ **Particular cases**

- an agent is *item-based* if $\varphi = 0$ for its utility
- an agent is *neighbourhood-based* if it receives an utility of 0 for all items.

➤ The social welfare, or global utility, is the sum of the utilities of all the agents.

Stability

➤ **Swap-Deal**

- A swap-deal between two agents i and j is said to be *improving* if and only if: $u_i(A(i)) < u_i(A(j))$ and $u_j(A(j)) < u_j(A(i))$
- A swap-deal between two agents i and j is *valid* if the allocation resulting from the swap of their respective assignment would still respect the quotas.

➤ **Stability:** An allocation is **stable** if there is no valid improving swap.

➤ **Price of Stability**

- Let U_{OPT} be the social welfare of the allocation maximizing the global utility, and let U_{STABLE} be the social welfare of the best stable allocation.
  We define the Price of Stability (PoS) as:

$$PoS = \frac{U_{OPT}}{U_{STABLE}}$$

- **Proposition:** PoS=1 when all the agents are item-based or when all the agents are neighbourhood-based. $PoS \geq 1$ in the general case.

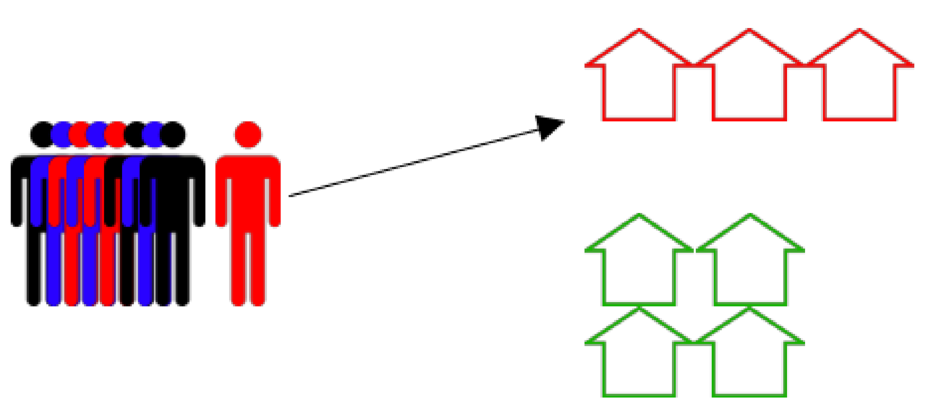
Sequential Mechanism

➤ **How does it work?**

- in some random order, the agents sequentially pick the items that maximize their utilities at the time of their selection, while respecting the diversity constraints.
- If an agent cannot be assigned to any item because of the quotas, it is skipped.
- The algorithm ends when all agents are either assigned or unable to be assigned to any remaining item

➤ **Propositions**

- The Sequential mechanism does not always return a valid allocation
- The Sequential mechanism does not always return a stable allocation, unless all agents are item-focused
- The **worst case error** for the sequential mechanism is the ratio between the social welfare of the optimal allocation and that of the worst allocation that the mechanism can return for a given instance.
- In the general case, the **worst-case error** is unbounded.
  If the number of types k is a constant, the ratio is upper-bounded by $\frac{(1+\varphi)k}{\varphi}$.



Swap Mechanism

➤ **How does it work?**

1. Start from a valid allocation
2. Pick a valid improving swap, and execute it.
3. When there is no more valid improving swap, the algorithm stops.

➤ **Propositions**

- **The swap mechanism will always reach a stable outcome**
- The **worst-case error** for the swap algorithm is unbounded in the general case.
  If the number of types k is a constant, the ratio is upper-bounded by $\frac{(1+\varphi)k}{\varphi}$.