



# Swap dynamics in single-peaked housing markets

Aurélie Beynier<sup>1</sup> · Nicolas Maudet<sup>1</sup>  · Simon Rey<sup>2</sup> · Parham Shams<sup>1</sup>

Accepted: 22 April 2021 / Published online: 19 May 2021  
© Springer Science+Business Media, LLC, part of Springer Nature 2021

## Abstract

This paper focuses on the problem of fairly and efficiently allocating resources to agents. We consider a specific setting, usually referred to as a *housing market*, where each agent must receive exactly one resource (and initially owns one). In this framework, in the domain of linear preferences, the Top Trading Cycle (TTC) algorithm is the only procedure satisfying Pareto-optimality, individual rationality and strategy-proofness. Under the restriction of single-peaked preferences, Crawler enjoys the same properties. These two centralized procedures might however involve long trading cycles. In this paper we focus instead on procedures involving the shortest cycles: bilateral swap-deals. In such swap dynamics, the agents perform pairwise mutually improving deals until reaching a swap-stable allocation (no improving swap-deal is possible). We prove that in the single-peaked domain every swap-stable allocation is Pareto-optimal, showing the efficiency of the swap dynamics. In fact, this domain turns out to be maximal when it comes to guaranteeing this property. Besides, both the outcome of TTC and Crawler can always be reached by sequences of swaps. However, some Pareto-optimal allocations are not reachable through improving swap-deals. We further analyze the outcome of swap dynamics through social welfare notions, in our context the average or minimum rank of the resources obtained by agents in the final allocation. We start by providing a worst-case analysis of these procedures. Finally, we present an extensive experimental study in which different versions of swap dynamics are compared to other existing allocation procedures. We show that they exhibit good results on average in this domain, under different cultures for generating synthetic data.

**Keywords** Resource allocation · Swap dynamics · Housing markets · Single-peaked domains

---

✉ Nicolas Maudet  
nicolas.maudet@lip6.fr

Aurélie Beynier  
aurelie.beynier@lip6.fr

Simon Rey  
s.j.rey@uva.nl

Parham Shams  
parham.shams@lip6.fr

<sup>1</sup> Sorbonne Université, CNRS, LIP6, 75005 Paris, France

<sup>2</sup> Institute for Logic, Languages and Computation, University of Amsterdam, Amsterdam, The Netherlands

## 1 Introduction

This paper studies the problem of allocating fairly and efficiently a set of indivisible resources to a set of agents. We consider a specific setting, usually referred to as a *housing market* [59], where each agent must receive exactly one resource (and initially owns one). We investigate housing markets by moreover assuming that the preferences are *single-peaked* [4, 13]. Under this assumption, there exists a common ordering of the resources such that the further you go from your most preferred resource—either going left or right according to the ordering—the less preferred the resources are. This domain restriction would for instance correspond to real-world situations where the preferences of agents are related to a one-dimensional space, such as the temperature in a room, the distance of a house to a facility (metro, bike rental station, supermarket), the price of a new product, the size of a cloth [49, 50]. In voting, this restriction domain may be appropriate in political elections where preferences usually decrease when considering candidates further away on the political spectrum from the most preferred candidate.

The design of resource allocation procedures is primarily guided by the the properties we want the final allocation, or the procedure itself, to satisfy. In single-peaked housing markets, the two prominent allocation procedures are *Top Trading Cycle* (TTC) [59] and *Crawler* [8]. They both satisfy a set of key desirable properties: Pareto-optimality, strategy-proofness and individual rationality [8, 43, 53, 59]. These three properties thus provide strong guarantees in terms of efficiency (Pareto-optimality) and incentive compatibility (individual rationality and strategy-proofness). In fact, Crawler has an additional advantage over TTC in terms of strategy-proofness, since it can implemented in obviously dominant strategies [8].

However, both TTC and Crawler require a significant amount of global coordination. Indeed, they are both based on *trading cycles*—cyclic exchanges of resources between several agents—which can potentially involve many agents. Such long cycles may not be acceptable or even feasible in practice, for instance because of the risk of failure they induce.

As an example, kidney exchange programs usually restrict the size of the cycles to two or three [55] because of time constraints.

Limiting the length of these cycles is thus a relevant agenda to reduce the coordination complexity and make procedures more robust. It is in particular a prerequisite to the development of more *decentralized* approaches that rely on the agents autonomously performing simple deals, without being hampered by a prohibitive coordination and communication cost.

Indeed, the use of a central coordinator induces a weak point in the system: the coordinator causes a bottleneck whose default often leads to the failure of the whole allocation process. The system may also be inherently distributed and the use of a central coordinator may not be possible because of limitations in the communication infrastructure or because of privacy requirements. In addition, centralized procedures may be perceived as less fair by the agents who are not allowed to actively take part in the allocation process [14, 40, 63]. While TTC and Crawler are both presented as centralized procedures (*i.e.* a central authority computes each trading cycle that should be implemented, and dictates it to the

agents<sup>1</sup>), distributed procedures [22, 57] take a different perspective. Agents autonomously negotiate over the resources and locally agree on deals, and the outcome of the resource allocation problem results from the sequence of such local deals. Such dynamics may achieve interesting efficiency and fairness results [20, 30]. However, long cycles still pose real challenges for these procedures [52] as they involve distributed coordination among numerous agents.

In this paper, we pursue this line of research and focus on the simplest possible local deals in the context of housing markets: *swap-deals* [24]. Under a *swap dynamics*, agents meet each other, in a pairwise fashion, and exchange their resources if they both benefit from it. The process iterates until a stable state, an equilibrium, is reached. Under this approach, once a trading cycle is selected, very little coordination is required. Nevertheless, selecting *which* swap-deal (with which agent) to perform may still require significant prior coordination, depending on the heuristic used. Hence this approach is best described as a family of dynamics, with different degrees of decentralization depending on the heuristics used to select the improving deals to be implemented. We shall study several of them in this paper.

*Contributions.* We first establish that a large class of swap dynamics is vulnerable to manipulation, unveiling a tension between efficiency and strategy-proofness for such decentralized procedures. On the positive side, we demonstrate that in housing markets with single-peaked preferences, every allocation that is stable with respect to swap-deals is Pareto-optimal. We also prove that the single-peaked domain is maximal in that respect: in other words, any larger domain would fail to offer such a guarantee. Moreover, even though some Pareto-optimal allocations may not be reached by a sequence of improving swap-deals, we show that the allocations returned by TTC and Crawler are both reachable: there exist sequences of swap-deals simulating these procedures, such that (potentially) long trading cycles are not necessary any longer. We further investigate how swap dynamics behave with respect to social welfare, taken as the average or minimum rank of the resource obtained by the agents. After a worst-case analysis, we explore experimentally the influence of different heuristics used to select deals. These experiments highlight that swap dynamics perform particularly well with respect to these objectives.

*Outline of the paper.* We first present some related works (Sect. 2) and introduce our model and some additional definitions in Sect. 3. Section 4 describes the centralized allocation procedures discussed in this paper, while swap dynamics are introduced in Sect. 5. A formal analysis of these procedures is presented in Sect. 6. Section 7 offers a comparison between swap dynamics outcomes and social welfare optimization, in a “price of anarchy” perspective. The experimental analysis is presented in Sect. 8. Sect. 9 concludes.

## 2 Related works

The problem of fair and efficient allocation of indivisible goods is well studied in economics, with original developments impelled by the computer science perspective in recent years [5, 16, 46, 65, 67]. In the present paper we focus on the model defined by [59], called

---

<sup>1</sup> In principle, it is possible to distribute the execution of central procedures, by letting all agents broadcast, compute locally their own version of the central algorithm, and coordinate. This might however induce a significant cost.

*housing market* or *assignment problem*, in which each agent must receive exactly one indivisible resource. Shapley and Scarf [59] defined the *Top-Trading Cycle* (TTC) algorithm which has been extensively studied [53] and shown to be the only mechanism satisfying Pareto-optimality when preferences are strict [43]. Bade [8] explored another direction by restricting preferences to single-peaked domains. In this case she presents the *Crawler* that satisfies the same properties as TTC. This work has been further explored by [42].

Following this line of work, we assume single-peaked preferences in this paper. This domain of preferences has been introduced by [4, 13]. Initially motivated in voting contexts, it is now a well studied domain of preferences [28, 45]. Numerous works have explored single-peaked preferences in the context of fair division. Sprumont [61] for instance characterized the *uniform allocation rule*, the unique strategy-proof, efficient and anonymous allocation procedure with single-peaked preferences and divisible resources. Kasajima [36] investigated probabilistic allocation of indivisible resources with single-peaked preferences. More recently, [34] extended this research area to indivisible resources and considered the problem of assigning agents to a line under single-peaked preferences. On the empirical side, it is not clear whether single-peaked preferences can be observed in real-life scenarios, as mentioned in [50]. On the one hand, [25, 41, 60] argue that interactions between the agents lead towards single-peaked preferences. On the other hand, [27] shows that political preferences might be double-peaked (and not single-peaked) when it comes to some polarizing topics.

Following the development of multi-agent systems, decentralized procedures have been defined through the idea of local exchanges between agents. Sandholm [57] considered the problem of reallocating tasks among individually rational agents. Aziz et al. [7] investigated different complexity problems in this setting. Chevaleyre et al. and Endriss et al. [21, 30] respectively characterized the class of deals and the class of preferences required to reach socially optimal allocations, while communication complexity issues were explored in [26, 29]. Chevaleyre et al. [20] focused on reaching efficient and envy-free allocations. Similar procedures were also introduced in the area of two-sided matching [2, 54]. The idea of using swap-deals was explored for instance by [1], who studied barter exchange networks. Gourvès et al., Huang and Xiao, Saffidine and Wilczynski [32, 35, 56] studied dynamics of swap-deals by considering an underlying social network constraining the possible interactions of the agents, albeit in a centralized perspective, and focusing on complexity issues. These results were recently extended by [10]. Brandt and Wilczynski [18] studied the convergence of swap-deals to Pareto-optimal allocations in housing, marriage and roommate markets.

### 3 Preliminaries

We start by presenting the basic components of our model and the different fairness and efficiency concepts discussed in this paper.

#### 3.1 The model

Let us consider a set  $\mathcal{N} = \{a_1, \dots, a_n\}$  of  $n$  agents and a set  $\mathcal{R} = \{r_1, \dots, r_n\}$  of  $n$  resources. An allocation  $\pi = \langle \pi_{a_1}, \dots, \pi_{a_n} \rangle$  is a vector of  $\mathcal{R}^n$  such that  $\bigcup_{a_i \in \mathcal{N}} \pi_{a_i} = \mathcal{R}$ . A element  $\pi_{a_i} \in \mathcal{R}$  of  $\pi$  represents the unique resource allocated to  $a_i$  (recall that we consider housing

markets where exactly one resource is assigned to each agent). The set of all allocations is denoted  $\Pi$ .

Following previous work [6, 15, 17, 59], agents are assumed to express their preferences over the resources through complete linear orders. We denote by  $\succ_i$  agent  $a_i$ 's preferences, where  $r_1 \succ_i r_2$  means that  $a_i$  prefers  $r_1$  over  $r_2$ .

For a given linear order  $\succ$ , we use  $top(\succ)$  to denote its top-ranked resource:  $\forall r \in \mathcal{R} \setminus \{top(\succ)\}, top(\succ) \succ r$ . Similarly,  $snd(\succ)$  refers to the second most preferred resource in  $\succ$ . With a slight abuse of notation we will write  $top(a_i)$  and  $snd(a_i)$  to refer to  $top(\succ_i)$  and  $snd(\succ_i)$ . When it is not clear from the context we will subscript these notations to specify the resource set. For instance  $top_R(a_i)$  is the most preferred resource for agent  $a_i$  among resources in  $R \subseteq \mathcal{R}$ .

Given a resource  $r \in \mathcal{R}$  and an agent  $a_i \in \mathcal{N}$ , we use  $rank_{a_i}(r)$  to refer to the rank of  $r$  in  $\succ_i$ . We have then  $rank_{a_i}(top(a_i)) = n$ ,  $rank_{a_i}(snd(a_i)) = n - 1$ , etc...

For a given agent  $a_i \in \mathcal{N}$ ,  $a_i$  strictly prefers  $\pi$  to  $\pi'$  (denoted as  $\pi \succ_i \pi'$ ) iff  $\pi_{a_i} \succ_i \pi'_{a_i}$ . We also introduce weak preferences over an allocation. For a given agent  $a_i \in \mathcal{N}$ ,  $a_i$  weakly prefers  $\pi$  to  $\pi'$  (written  $\pi \succeq_i \pi'$ ) iff  $\pi_{a_i} \succeq_i \pi'_{a_i}$  or  $\pi_{a_i} = \pi'_{a_i}$ .

Finally, a preference profile  $L = \langle \succ_i \mid a_i \in \mathcal{N} \rangle$  is a vector of  $n$  linear orders, one for each agent. An instance of a resource allocation problem is then a vector  $I = \langle \mathcal{N}, \mathcal{R}, L, \pi^0 \rangle$  composed of a set of agents  $\mathcal{N}$ , a set of resources  $\mathcal{R}$ , a preference profile  $L$  and an initial allocation  $\pi^0$ .

In some settings, natural properties of the agents' preferences can be identified, thus restricting the set of possible preference orderings. The notion of preference domain formalizes these restrictions. For a set of resources  $\mathcal{R}$ , we denote by  $\mathcal{L}_{\mathcal{R}}$  the set of all linear orders over  $\mathcal{R}$ . Any subset  $D \subseteq \mathcal{L}_{\mathcal{R}}$  is then called a preference domain.

We say that an instance  $I = \langle \mathcal{N}, \mathcal{R}, L, \pi^0 \rangle$  is defined over a preference domain  $D$  if the preferences of the agents are selected inside  $D$ , that is, for every  $\succ_i \in L$ , we have  $\succ_i \in D$ . The set of all instances defined over  $D$  is denoted  $\mathcal{I}_D$ .

### 3.2 Single-peaked preferences

Under single-peaked preferences [4, 13], the agents are assumed to share a common axis  $\triangleleft$  over the resources and individual rankings are defined with respect to this axis.

**Definition 1** Let  $\mathcal{R}$  be a set of resources and  $\triangleleft$  an axis (i.e. a linear order) over  $\mathcal{R}$ . We say that a linear order  $\succ$  is single-peaked with respect to  $\triangleleft$  if we have:

$$\forall (r_1, r_2) \in \mathcal{R}^2 \text{ s.t. } \left. \begin{array}{l} r_2 \triangleleft r_1 \triangleleft top(\succ), \\ \text{or, } top(\succ) \triangleleft r_1 \triangleleft r_2 \end{array} \right\} \Rightarrow r_1 \succ r_2.$$

In words,  $\succ$  is single-peaked with respect to  $\triangleleft$  if  $\succ$  is decreasing on both the left and the right sides of  $top(\succ)$ , where left and right are defined by  $\triangleleft$ .

For a given axis  $\triangleleft$ , we call  $\mathcal{SP}_{\triangleleft}$  the set of all the linear orders single-peaked with respect to  $\triangleleft$ :

$$\mathcal{SP}_{\triangleleft} = \{ \succ \in \mathcal{L}_{\mathcal{R}} \mid \succ \text{ is single-peaked w.r.t. } \triangleleft \}.$$

A preference domain  $D$  is called single-peaked if and only if there exists an axis  $\triangleleft$  such that  $D \subseteq SP_{\triangleleft}$ . An instance  $I$  is said to be single-peaked if it is defined over a single-peaked preference domain.

Based on these definitions, it is straightforward that restricting preference domains preserves single-peakedness.

**Observation 1** *Let  $D$  be a preference domain single-peaked over  $\triangleleft$ . For a subset of resources  $\mathcal{R}' \subseteq \mathcal{R}$ , the domain  $D'$  defined as the restriction of  $D$  to  $\mathcal{R}'$  is a single-peaked domain over  $\triangleleft'$ , the restriction of  $\triangleleft$  to  $\mathcal{R}'$ .*

[9] provided a characterization of single-peaked domains. In particular, they gave a necessary condition for a domain to be single-peaked: it should be worst-restricted [58].

**Definition 2** An instance  $I = \langle \mathcal{N}, \mathcal{R}, L, \pi^0 \rangle$  is worst-restricted if for any triplet of resources  $(r_x, r_y, r_z) \in \mathcal{R}^3$ , one of them is never ranked last in the restriction of  $L$  to these three resources.

**Proposition 1** ([9]) *If an instance is single-peaked then it is worst-restricted.*

Let us illustrate the single-peaked domain with a simple example.

**Example 1** Consider the following four linear orders defined over 3 resources.

$$\begin{aligned} \succ_1 : r_1 >_1 r_2 >_1 r_3 \\ \succ_2 : r_3 >_2 r_2 >_2 r_1 \\ \succ_3 : r_2 >_3 r_1 >_3 r_3 \\ \succ_4 : r_2 >_4 r_3 >_4 r_1 \end{aligned}$$

One can check that these linear orders represent a single-peaked preference profile with respect to  $\triangleleft$  defined as:  $r_1 \triangleleft r_2 \triangleleft r_3$ . In fact these orders exactly correspond to  $SP_{\triangleleft}$ . However, the following preferences are not single-peaked:

$$\begin{aligned} \succ_1 : r_1 >_1 r_2 >_1 r_3 \\ \succ_2 : r_3 >_2 r_1 >_2 r_2 \\ \succ_3 : r_2 >_3 r_3 >_3 r_1 \end{aligned}$$

It can be checked that there is no linear order  $\triangleleft$  over which these preferences are single-peaked. Indeed, they are not worst-restricted: every resource of the triplet  $(r_1, r_2, r_3)$  is ranked last at least once, which violates Proposition 1.

### 3.3 Efficiency and fairness criteria of an allocation

There exists an extensive literature investigating how to define efficiency requirements for an allocation (see [19] and [65] for some surveys). Pareto-optimality is well-known basic efficiency requirement.

**Definition 3** (Pareto-optimality) Let  $\pi$  and  $\pi'$  be two allocations. We say that  $\pi'$  Pareto-dominates  $\pi$  if and only if:

$$\forall a_i \in \mathcal{N}, \pi' \succeq_i \pi, \text{ and } \exists a_j \in \mathcal{N}, \pi' \succ_j \pi.$$

An allocation  $\pi$  is then said to be Pareto-optimal if there is no other allocation  $\pi'$  that Pareto-dominates  $\pi$ .

Efficiency can also be evaluated via social welfare measures. One such measure is the *average rank* (*ark*) of the resources held by the agents, defined as:

$$ark(\pi) = \frac{1}{n} \cdot \sum_{a_i \in \mathcal{N}} rank_{a_i}(\pi_{a_i}).$$

Maximizing the average rank is of course equivalent to maximizing the sum of ranks, and can also be interpreted as the utilitarian social welfare, under the assumption that agents have Borda utilities.

However, maximizing the average rank or searching for Pareto-optimal solutions may not be satisfactory as it can lead to particularly unfair allocations. For this reason, many fairness criteria have been introduced. We will focus here on maximizing the *minimum rank* (*mrk*) of the resources held by the agents, defined as:

$$mrk(\pi) = \min_{a_i \in \mathcal{N}} rank_{a_i}(\pi_{a_i}).$$

Again, if we were to interpret the rank as a cardinal utility function, the minimum rank would be equivalent to the egalitarian welfare. Maximizing the minimum rank follows Rawls' principle of maximizing the welfare of the worst-off [51]. It has been introduced by [48] and is now a very common rule in fair division [47, 62, 64].

### 3.4 Properties of the procedures

Let us now turn our attention to the allocation procedures. An allocation procedure  $M$  is a mapping taking as input an instance  $I$  and returning an allocation  $M(I)$ .

We will say that an allocation procedure is Pareto-optimal if it always return a Pareto-optimal allocation. Given that the agents initially own the resources in our setting, another very common efficiency requirement is that of *individual rationality*. It stipulates that no agent should be worse-off in the final allocation.

**Definition 4** (Individual rationality) A procedure  $M$  is said to be individually rational if for every instance  $I = \langle \mathcal{N}, \mathcal{R}, L, \pi^0 \rangle$  and every agent  $a_i \in \mathcal{N}$ , we have:

$$M(I) \succeq_i \pi_{a_i}^0$$

Let us illustrate through an example the different criteria introduced so far.

**Example 2** Let us consider the following instance with 5 agents and 5 resources. The preferences presented below are single-peaked with respect to  $r_1 \triangleleft r_2 \triangleleft r_3 \triangleleft r_4 \triangleleft r_5$ . The initial allocation  $\pi^0 = \langle r_1, r_2, r_3, r_4, r_5 \rangle$  is represented by the underlined resources.

$$\begin{aligned}
 a_1 &: r_3 \succ_1 r_4 \succ_1 r_5 \succ_1 \boxed{r_2} \succ_1 r_1 \\
 a_2 &: r_3 \succ_2 \boxed{r_4} \succ_2 r_5 \succ_2 \underline{r_2} \succ_2 r_1 \\
 a_3 &: r_4 \succ_3 \boxed{r_5} \succ_3 \underline{r_3} \succ_3 r_2 \succ_3 r_1 \\
 a_4 &: \boxed{r_3} \succ_4 r_4 \succ_4 r_5 \succ_4 r_2 \succ_4 r_1 \\
 a_5 &: \boxed{r_1} \succ_5 r_2 \succ_5 r_3 \succ_5 r_4 \succ_5 \underline{r_5}
 \end{aligned}$$

The allocation  $\pi^0$  is not Pareto-optimal as it is Pareto-dominated by the squared allocation  $\boxed{\pi} = \langle r_2, r_4, r_5, r_3, r_1 \rangle$ . We have  $ark(\boxed{\pi}) = 4$  and  $mrk(\boxed{\pi}) = 2$ .

We conclude this section by introducing another desirable property that is *strategy-proofness*. Informally, an allocation procedure is strategy-proof if no agent could get a strictly better outcome by lying instead of revealing her true preferences.

**Definition 5** Let  $L = \langle \succ_j \rangle_{a_j \in \mathcal{N}}$  be a preference profile. For a given agent  $a_i \in \mathcal{N}$ , an  $i$ -variant  $L_{-i} = \langle \succ'_j \rangle_{a_j \in \mathcal{N}}$  of  $L$  is a preference profile such that:

$$\forall a_j \in \mathcal{N} \setminus \{a_i\}, \succ_j = \succ'_j \text{ and } \succ_i \neq \succ'_i.$$

An allocation procedure  $M$  is said to be strategy-proof if for every instance  $I = \langle \mathcal{N}, \mathcal{R}, L, \pi^0 \rangle$ , every agent  $a_i \in \mathcal{N}$ , every  $i$ -variant  $L_{-i}$  of  $L$ , we have:

$$M(\langle \mathcal{N}, \mathcal{R}, L, \pi^0 \rangle) \succeq_i M(\langle \mathcal{N}, \mathcal{R}, L_{-i}, \pi^0 \rangle).$$

## 4 Centralized allocation procedures for housing markets

This section introduces two centralized allocation procedures that will be used as references in the paper. The first one is the TTC algorithm [59], which can be used without any domain restriction. It is well known to satisfy the three main desirable properties of an allocation procedure: *Pareto-optimality*, *individual rationality* and *strategy-proofness*. The second one, Crawler [8], is specially designed for single-peaked domains. It satisfies the same properties as TTC. Both of these procedures are based on the notion of deals.

### 4.1 Improving deals

A deal is a vector of agents, usually denoted by  $\mu = \langle a_1, \dots, a_k \rangle$ , where  $a_i \in \mathcal{N}, \forall i \in \{1, \dots, k\}$ . It represents an exchange where agent  $a_i$  gives her resource to agent  $a_{i+1}$  for each  $i \in \{1, \dots, k-1\}$  and agent  $a_k$  gives her resource to agent  $a_1$ . With a slight abuse of notations, given a deal  $\mu = \langle a_1, \dots, a_k \rangle$ , agent  $a_{k+1}$  will refer to agent  $a_1$  and agent  $a_0$  (i.e. the agent before  $a_1$ ) to agent  $a_k$ . For the particular case of deals involving only two agents, i.e.  $k = 2$ , we will talk about *swap-deals*.



Interestingly, it can be observed that in housing markets, any reallocation (permutation of resources) can be implemented as a collection of disjoint cycle-deals [59]. This notion of deal is thus sufficient in this context.

Let us now introduce the concept of improving deals.

**Definition 6** Let  $\pi$  be an allocation, and  $\mu = \langle a_1, \dots, a_k \rangle$  a deal involving  $k \geq 1$  agents. The allocation  $\pi[\mu]$  obtained by applying the deal  $\mu$  to  $\pi$  is defined by:

$$\begin{cases} \pi[\mu]_{a_i} = \pi_{a_{i-1}} & \text{if } i \in \{1, \dots, k\}, \\ \pi[\mu]_{a_i} = \pi_{a_i} & \text{otherwise.} \end{cases}$$

A deal is said to be *improving* if  $\pi[\mu] \succ_i \pi$  for every agent  $a_i$  involved in  $\mu$ .

Observe that for a deal  $\mu$  of length 1, we have  $\pi[\mu] = \pi$ . This trivial case consists of an agent giving her resource to herself. It will be useful to simplify the presentation of TTC and Crawler. In particular, when an agent already holds her top resource, she should also hold it in the final allocation because of individual rationality.

It is also straightforward to see that a procedure applying only improving deals trivially satisfies individual rationality.

## 4.2 Gale's top trading cycle algorithm

The TTC algorithm takes as input an instance  $I = \langle \mathcal{N}, \mathcal{R}, L, \pi^0 \rangle$  and proceeds as follows. The algorithm maintains a set of available agents  $N$  and a set of available resources  $R$  where initially  $R = \mathcal{R}$  and  $N = \mathcal{N}$ . At each step of the algorithm, a directed bipartite graph  $G = \langle V, E \rangle$ , with  $V = N \cup R$ , is defined. The nodes of  $G$  represent the agents in  $N$  and the resources in  $R$ , and the set of edges  $E$  is such that:

- there is a directed edge  $(a_i, r_i)$  between  $a_i$  and  $r_i$  if and only if  $r_i = \text{top}_R(a_i)$  i.e. agents are linked to their preferred resource in  $R$ ,
- there is a directed edge  $(r_i, a_i)$  between  $r_i$  and  $a_i$  if and only if  $r_i = \pi_{a_i}^0$  i.e. resources are linked to their owner in  $\pi^0$ .

Note that there always exists at least one cycle in  $G$  and that cycles correspond to improving deals. The cycle-deals constructed can be of size 1 if an agent already owns her top resource in  $R$ . The TTC algorithm selects one of the cycles  $\mu$  in  $G$ . The agents and resources involved in  $\mu$  are then removed from  $N$  and  $R$ . A new graph  $G'$  is computed with the remaining agents and resources. The process is iterated on the new graph  $G'$  and  $\pi^0[\mu]$  until an empty graph has been reached. Note that the outcome of TTC is unique (regardless of the possibly different choices of cycles to implement during the process).

**Algorithm 1:** TTC algorithm

---

**Input:** An instance  $I = \langle \mathcal{N}, \mathcal{R}, L, \pi^0 \rangle$   
**Output:** An allocation  $\pi$

```

1  $\pi \leftarrow$  empty allocation
2  $R \leftarrow \mathcal{R}$ : list of resources
3  $N \leftarrow \mathcal{N}$ : list of agents
4 while  $N \neq \emptyset$  do
5    $E \leftarrow \emptyset$ 
6    $V \leftarrow N \cup R$ 
7   for each agent  $a \in N$  do
8     | Add a directed edge in  $E$  between  $a$  and  $top_R(a)$ 
9   end
10  for each resource  $r \in R$  do
11    | Add a directed edge in  $E$  between  $r$  and her owner in  $\pi^0$ 
12  end
13  Select a cycle  $\mu$  from the graph  $G = \langle V, E \rangle$ 
14   $do(\mu, \pi)$ 
15 end
16 return  $\pi$ 

```

---

A formal description of TTC procedure is given in Algorithm 1. Note that we make use of the sub-procedure  $do(\mu, \pi)$  which simply implements the cycle-deal  $\mu$  and adds the resulting allocation (restricted to the agents involved in the deal) in the allocation  $\pi$ . It then removes the agents and the resources involved in the deal  $\mu$  from the lists of available agents and resources,  $N$  and  $R$  respectively.

**Example 3** Let us consider the instance defined in Example 2. The reader can check that the allocation returned by TTC is  $\pi^{TTC} = \langle r_5, r_2, r_4, r_3, r_1 \rangle$ . Indeed, the first cycle-deal that can be applied is  $\mu_1 = \langle a_3, a_4 \rangle$ . The next one is  $\mu_2 = \langle a_1, a_5 \rangle$ . Finally,  $a_2$  remains with her initial resource which corresponds to the cycle  $\mu_3 = \langle a_2 \rangle$ .

### 4.3 Crawler algorithm

In Crawler algorithm, agents are initially ordered along the single-peaked axis according to the resource they hold. The first agent is then the one holding the resource on the left end side of the axis. As in TTC, the list of available resources is denoted by  $R$  and is ordered according to the single-peaked axis. A list of available agents  $N$  such that the  $i$ th agent of the list is the one who holds the  $i$ th resource in  $R$  is also maintained.

Considering agents sequentially from the first in  $N$  to the last in  $N$ , Crawler checks for each agent  $a_i$  where her top resource,  $top_R(a_i)$ , is on the axis.<sup>2</sup>

- If  $top_R(a_i)$  is on her right, the algorithm moves to the next agent.
- If  $a_i$  holds her top resource  $top_R(a_i)$ , then  $top_R(a_i)$  is allocated to  $a_i$ . Agent  $a_i$  and the resource  $top_R(a_i)$  are removed from  $N$  and  $R$ . The algorithm starts again from the agent on the left end of the axis.

<sup>2</sup> Note that the algorithm can equivalently be executed from the last agent to the first one.

- If  $top_R(a_i)$  is on the left of  $a_i$ , the agent is allocated her top resource  $top_R(a_i)$ . We denote by  $t^*$  the index of  $top_R(a_i)$  and  $t$  the index of  $a_i$  (we have  $t^* < t$ ). Then, every agent between  $t^*$  and  $t - 1$  receives the resource held by the agent on her right (the resources “crawl” towards left). Once again,  $a_i$  and  $top_R(a_i)$  are removed from  $N$  and  $R$  and the algorithm restarts from the first agent.

Once all the resources have been allocated, the algorithm terminates.

A formal description of Crawler procedure is given in Algorithm 2. Note that we make use of the sub-procedure  $pick(a_{t^*}, r, N, R, \pi)$  which simply assigns the resource  $r$  to the given agent  $a_{t^*}$  in the allocation  $\pi$ , and then removes the agent and the resource from the lists of available agents and resources,  $N$  and  $R$  respectively. Since the list of resources is ordered following the single-peaked axis and the  $i$ th agent in  $N$  corresponds to the owner of the  $i$ th resource in  $R$ , the removal of  $r$  and  $a_{t^*}$  is in fact equivalent to assigning  $r$  to agent  $a_{t^*}$  and crawling the resources from right to left.

---

**Algorithm 2:** Crawler algorithm

---

```

Input: An instance  $I = \langle N, \mathcal{R}, L, \pi^0 \rangle$  single-peaked with respect to  $\triangleleft$ 
Output: An allocation  $\pi$ 
1  $\pi \leftarrow$  empty allocation
2  $R \leftarrow \mathcal{R}$ : list of resources sorted accordingly to  $\triangleleft$ 
3  $N \leftarrow \mathcal{N}$ : list of agents such that the  $i^{th}$  agent is the one who initially holds the  $i^{th}$  resource in  $R$ 
4 while  $N \neq \emptyset$  do
5    $t^* \leftarrow |N|$ 
6   for  $i = 0$  to  $|N| - 1$  do
7     if  $r_i \succ_i r_{i+1}$  then
8        $t^* \leftarrow i$ 
9       Break
10    end
11  end
12   $r \leftarrow top_R(a_{t^*})$ 
13   $pick(a_{t^*}, r, N, R, \pi)$ 
14 end
15 return  $\pi$ 

```

---

Let us illustrate the execution of Crawler:

**Example 4** On the instance of Example 2, agent  $a_4$  is the first agent whose top resource is not on her right, she thus receives her top resource  $r_3$ . The second step matches agent  $a_3$  to  $r_4$ . On the third step, agents  $a_1$  and  $a_2$  both have their top resources (among the remaining resources) on the right but the last agent  $a_5$  has hers on her left.  $a_5$  is then matched to  $r_1$ . Resource  $r_2$  crawls to agent  $a_1$  and resource  $r_5$  crawls to agent  $a_2$ . On the fourth step,  $a_2$  picks her current resource  $r_5$ . Finally,  $a_1$  is assigned resource  $r_2$ . The allocation returned by Crawler is  $\pi^C = \langle r_2, r_5, r_4, r_3, r_1 \rangle$ . At each step  $i$  of the procedure, an improving cycle-deal  $\mu_i$  is applied (with the last agent in the cycle picking her top resource among the remaining ones):  $\mu_1 = \langle a_3, a_4 \rangle$ ,  $\mu_2 = \langle a_3 \rangle$ ,  $\mu_3 = \langle a_1, a_2, a_5 \rangle$ ,  $\mu_4 = \langle a_2 \rangle$ ,  $\mu_5 = \langle a_1 \rangle$ . One can observe that on this example the allocation returned by Crawler is not the same as the one returned by TTC. However, both procedures lead to the same minimum rank  $mrk$  and average rank  $ark$ .

Interestingly, a variant of this procedure allows to check Pareto-optimality in single-peaked domains in linear time [12].

## 5 Swap dynamics: a family of procedures based on swap-deals

We will now focus on dynamics based on local exchanges between the agents. Let us first introduce some additional definitions.

Given an allocation  $\pi$ , we denote by  $C_k(\pi)$ ,  $k \geq 2$ , the set of all the improving deals of size at most  $k$  that can be applied from  $\pi$ :

$$C_k(\pi) = \{\mu \mid \mu \text{ is an improving deal and } |\mu| \leq k\}.$$

When investigating procedures based on improving exchanges, we will try to reach allocations that are stable with respect to some deals.

**Definition 7** For a given  $k \in \{1, \dots, n\}$ , an allocation  $\pi$  is said to be stable with respect to  $C_k$  if  $C_k(\pi) = \emptyset$ .

It is obvious from this definition that if an allocation is stable with respect to  $C_k$  for a given  $k$ , it is also stable with respect to any  $k' < k$ . Moreover, since we are considering housing markets, it can be observed that an allocation is Pareto-optimal if and only if it is stable with respect to  $C_n$ .

Dynamics based on local exchanges start from an initial allocation and let the agents negotiate improving cycle-deals involving at most  $k$  agents until reaching an allocation stable with respect to  $C_k$ .

Following this process, to each  $k \in \{1, \dots, n\}$  corresponds a family of allocation procedures based on  $C_k$ . Indeed, at each step, the improving deal to be implemented can be selected in many different ways. We call *selection heuristic* a mapping giving the deal to implement at a given step of the procedure. A dynamic based on local exchanges will thus be defined by a given  $k$  and a specific selection heuristic. Depending on the selection heuristic, it may be the case that the selected deal is not an improving one. Such a deal could then be refused by the agents. We will say that a deal is *successful* if all agents involved in it agree to exchange. The sequence of proposed deals, together with the fact they were successful or not will be called a *history*.

**Algorithm 3:** Cycle-deals dynamics

---

**Input:** An instance  $I = \langle \mathcal{N}, \mathcal{R}, L, \pi^0 \rangle$ ,  
a maximum size  $k$  for the deals,  
a selection heuristic  $\sigma$

**Output:** An allocation  $\pi$  stable with respect to  $C_k$

```

1  $\pi \leftarrow \pi^0$ 
2 while  $\pi$  is not stable with respect to  $C_k$  do
3    $\mu \leftarrow \sigma(\mathcal{N}, k, \dots)$ 
4   if  $\mu$  is successful then
5      $do(\mu, \pi)$ 
6   end
7 end
8 return  $\pi$ 

```

---

A general formal description of cycle-deals dynamics is given in Algorithm 3. The  $do(\mu, \pi)$  method is similar to the one described in Algorithm 1 and implements the deal  $\mu$  (as chosen by the selection heuristic  $\sigma$ ) on the allocation  $\pi$ . Note that the selection heuristic itself takes as input at least the set of agents  $\mathcal{N}$  and the maximum size  $k$  of the cycle-deals (but it may be more informed and take more parameters, as we shall discuss later in this section). Unless stated otherwise we shall assume from now on that  $k = 2$  and omit this parameter, as we will mainly be interested in deals of size 2. We will also use *swap-stability* to refer to stability with respect to  $C_2$ .

We say that the procedure has *reached* the allocation obtained upon termination. Termination occurs when, for every possible swap-deal  $\langle a_x, a_y \rangle$ , there exists a latest unsuccessful proposal, such that there was subsequently no successful swap-deal involving  $a_x$  nor  $a_y$ . This guarantees in particular that the process cannot end as long as there remains an improving deal that has not yet been proposed. To ensure termination, the selection heuristic should not prevent some deals from happening. We will thus require heuristics to satisfy a property of full coverage as defined below.

**Definition 8** A selection heuristic  $\sigma$  has *minimal* (resp. *full*) *coverage* if for any  $a_x, a_y \in \mathcal{N}^2$ , there exists at least one round in the sequence (resp. after the latest successful swap-deal involving  $a_x$  or  $a_y$  if there is one), when the swap-deal  $\langle a_x, a_y \rangle$  is proposed.

$\mathcal{M}_2$  will denote the family of swap dynamics defined with respect to a selection heuristic with full coverage. It is useful to make a further distinction between different types of selection heuristics, depending on the information they take as input. In particular, heuristics may require preferential information (e.g. which resources agents would be happy to swap their current resource with), or on the contrary, be solely based on observable information (e.g. the history of deals). We will mostly focus on the latter in what follows.

A history-based selection heuristic  $\sigma$  is a function taking as input the set of agents  $\mathcal{N}$ , the history of deals  $h$ , and returning a deal  $\sigma(\mathcal{N}, h)$  to be proposed. Swap dynamics which rely on history-based selection heuristics will be called *history-based swap dynamics*.

A swap dynamics equipped with a given history-based selection heuristic  $\sigma$  on the instance  $I = \langle \mathcal{N}, \mathcal{R}, L, \pi^0 \rangle$ , proceeds iteratively as follows. The history  $h$  is initially empty. At each round corresponding to an allocation  $\pi$  and a history  $h$ , the allocation is updated to  $\pi'$  defined as:

$$\pi' = \begin{cases} \pi[\sigma(\mathcal{N}, h)] & \text{if the agents in } \sigma(\mathcal{N}, h) \text{ agree on swapping} \\ \pi & \text{otherwise.} \end{cases}$$

The deal  $\sigma(\mathcal{N}, h)$  together with its “success status” are then added to the history  $h$ .

We now give some examples of selection heuristics which will be studied in this paper. We start with *round-robin* heuristics, which simply specify a predefined way to order the different pairs of agents (i.e. possible swaps), and repeat it until termination. There are several ways to proceed, we give here two prominent examples:

- *Round-Robin over the Agents* ( $M_2$ -RRA): agents are ordered and paired following their name ( $\langle a_1, \dots, a_n \rangle$ ) in a round-robin fashion. The first agent  $a_1$  is paired with each other agent, the second agent  $a_2$  is paired with each other agent  $a_j$  with  $j > 2$  and so on. The agents then go over possible deals by iterating over the following sequence:

$$(a_1, a_2), (a_1, a_3), \dots, (a_1, a_n), (a_2, a_3), \dots, (a_{n-1}, a_n), (a_1, a_2), \dots$$

- *Round-Robin over Pairs of agents* ( $M_2$ -RRP): agents are ordered and paired following their name ( $\langle a_1, \dots, a_n \rangle$ ) in a round-robin fashion. In this case, the first agent is paired with the second agent, the second agent is paired with the third... Hence, the agents go over possible deals by iterating over the following sequence:

$$(a_1, a_2), (a_3, a_4), \dots, (a_{n-1}, a_n), (a_1, a_3), (a_2, a_4), \dots, (a_1, a_n), \dots$$

There is an obvious bias in the way deals are selected, it is thus natural to introduce some stochasticity in the process. One way to do this is as follows:

- *Randomized Round-Robin over deals* ( $M_2$ -RRR): the heuristic picks uniformly at random a permutation among all the possible deals. This permutation defines a round-robin order in which the deals are considered. The agents then go over possible deals by iterating over this permutation.

The three previous heuristics all guarantee by construction that all possible swap-deals were proposed before a swap-deal gets proposed again. The following natural heuristic do not have such guarantee.

- *Uniform* ( $M_2$ -U): a pair of agents (i.e. a swap-deal)  $(a_i, a_j)$  is selected uniformly at random among all possible pairs.
- *Random matchings* ( $M_2$ -RM): this heuristic proceeds in succession of matching steps where at each step, every agent is matched to a unique other agent and all the resulting swap-deals are proposed to the agents. This simulates a natural market where agents are paired randomly and simultaneously try to perform bilateral deals.

Note that these selection heuristics do not all have the same degree of decentralization. In particular,  $M_2$ -U can be easily executed in a fully decentralized way. Round-robin heuristics require a central entity to broadcast the sequence of pairs. Then, the agents can meet in a distributed way. Finally,  $M_2$ -RM can be implemented in a distributed way using a protocol ensuring that an agent can not encounter several agents simultaneously. Each agent  $a_i$  then selects uniformly at random an agent  $a_j$  to encounter and contact her. If agent  $a_j$  is already engaged in another encounter,  $a_i$  selects another agent.

We conclude the section by a straightforward observation: different selection heuristics can lead to different outcomes given the same initial allocation. Let us illustrate this with a simple example.

**Example 5** Consider the instance described in Example 2. We showed that Crawler and TTC return different allocations. In fact, these allocations can be reached by a sequence of swap-deals. Observe first that the cycle-deals applied by TTC are at most of length 2 (Example 3). The allocation  $\pi^{TTC} = \langle r_5, r_2, r_4, r_3, r_1 \rangle$  is obtained through two swap-deals:  $\langle a_3, a_4 \rangle$  and  $\langle a_1, a_5 \rangle$ . The allocation  $\pi^C = \langle r_2, r_5, r_4, r_3, r_1 \rangle$  returned by Crawler also is reachable by swap-deals. It is obtained by applying the following sequence:  $\langle a_3, a_4 \rangle$ ,  $\langle a_2, a_5 \rangle$  and  $\langle a_5, a_1 \rangle$ . Since these two allocations are Pareto-optimal, they are swap-stable. This shows that the way swap-deals are selected may affect the final stable allocation. Notice that these are not the only two swap-stable allocations reachable from  $\pi^0$  of the instance:  $\langle r_3, r_2, r_5, r_4, r_1 \rangle$  and  $\langle r_2, r_3, r_5, r_4, r_1 \rangle$  are the two other ones.

This example suggests that the allocations returned by Crawler and by TTC can both be reached via swap-deals—a point we will make formal in the next section. More generally, as TTC and Crawler both provide desirable guarantees, on Pareto-optimality, individual rationality and strategy-proofness, it is natural to study whether swap dynamics enjoy similar properties.

## 6 Properties of swap dynamics

We now investigate properties of swap dynamics. Recall first the easy observation made in Sect. 4: swap dynamics, because they rely on improving deals, are individually rational. But what about strategy-proofness and Pareto-optimality? We first discuss strategy-proofness, showing that such procedures are in general subject to manipulation. We next prove that any allocation stable with respect to swap-deals is Pareto-optimal and that both the allocation returned by TTC and Crawler can be reached via swap-deals. Finally, we show that the single-peaked domain is a maximal domain when it comes to guaranteeing Pareto-optimality.

### 6.1 Strategy-proofness

Although strategy-proofness is usually defined for centralized procedures, the question is still relevant for swap dynamics.

In a swap dynamic, an agent has the opportunity to behave strategically only when she is asked to accept or reject a proposed deal. As usual, we assume here that a potential manipulator has full knowledge of the preferences of the other agents and is aware of the fact that the selection heuristics has minimal coverage. In other words, the manipulator only needs to know whether she will ever have the opportunity to swap with some other agent or not. A swap dynamic will be called *strategy-proof* when, for every instance, at no point during the procedure an agent can be better off by accepting a non-improving deal or by refusing an improving one. Otherwise, the procedure is *manipulable*.

**Proposition 2** *Any history-based swap dynamic with minimal coverage is manipulable.*

**Proof** We are given an arbitrary history of deals  $h$  starting with  $(a_x, a_y)$ , as produced by the selection heuristic.

We are now going to show that we can build an instance  $I = \langle \mathcal{N}, \mathcal{R}, L, \pi^0 \rangle$  such that agent  $a_x$  will have an incentive to accept the first swap-deal  $(a_x, a_y)$ , even though this is not rational. The instance involves  $a_x, a_y, a_z$ , as well as  $n - 3$  other dummy agents, and assumes preferences to be single-peaked with respect to  $r_1 \triangleleft r_2 \triangleleft r_3 \triangleleft r_{d_1} \triangleleft \dots \triangleleft r_{d_{n-3}}$ .

$$\begin{aligned}
 a_x &: r_2 \succ_1 \underline{r_3} \succ_1 r_1 \succ_1 r_{d_1} \dots \\
 a_y &: r_2 \succ_2 r_3 \succ_2 \underline{r_1} \succ_2 r_{d_1} \dots \\
 a_z &: r_1 \succ_3 \underline{r_2} \succ_3 r_3 \succ_3 r_{d_1} \dots \\
 d_1 &: \underline{r_{d_1}} \succ \dots \\
 &\vdots \\
 d_{n-3} &: \underline{r_{d_{n-3}}} \succ \dots
 \end{aligned}$$

Observe first that all the dummy agents have their top resource from the start, hence they will not be involved in any deal. Now, consider the situation when  $a_x$  is given the opportunity to deal with  $a_y$ . If  $a_x$  is truthful (as all the other agents), no improving swap-deal involving agent  $a_x$  can occur, and she will end up with resource  $r_3$ . Now, suppose instead that  $a_x$  strategically accepts  $\langle a_x, a_y \rangle$ . This deal is improving for  $a_y$  so if  $a_x$  agrees on it, it will be implemented. In that case,  $a_y$  can no longer exchange with  $a_z$ , hence the only improving swap-deal left is  $\langle a_x, a_z \rangle$ . Because the selection heuristic has minimal coverage, this opportunity will occur at some point in the future. In the final allocation, agent  $a_x$  would then hold her top resource  $r_2$  (obtained from the swap-deal  $\langle a_x, a_z \rangle$ ).

□

Now, suppose that for a given selection heuristic, there exists a swap-deal  $\langle a_x, a_y \rangle$  that will never be proposed. Consider then the instance where all the agents have their top resource, except for  $a_x$  and  $a_y$  who would be happy to perform a swap-deal—but will never get a chance to. As no other swap-deal is possible, the outcome is certainly not Pareto-optimal. This leads to the following observation.

**Observation 2** Consider a history-based swap dynamic  $M_2$  defined with respect to a selection heuristic  $\sigma$ . If  $M_2$  is Pareto-optimal, then  $\sigma$  has minimal coverage.

Together with Proposition 2, this tells us that there is a fundamental tension between strategy-proofness and Pareto-optimality for history-based swap-deal procedure.

**Proposition 3** No history-based swap dynamic can be both Pareto-optimal and strategy-proof.

The reason why other types of dynamics may not be concerned by this result is that they can potentially condition the selection of deals to the preferences.

### 6.2 Pareto-optimality of swap dynamics

We show here that any allocation reached by swap dynamics is Pareto-optimal.



**Theorem 1** *Let  $I$  be a single-peaked instance, every allocation  $\pi$  in  $I$  that is stable with respect to swap-deals also is stable with respect to  $C_n$ .*

**Proof** Let us consider an allocation  $\pi$  stable with respect to  $C_2$  but not with respect to  $C_n$ . Since  $\pi$  is not stable with respect to  $C_n$ , there exists at least one improving deal in  $\pi$ . Let  $\mu$  be the smallest improving cycle-deal, i.e., which involves the smallest number of agents. Assume without loss of generality that  $\mu = \langle a_1, \dots, a_k \rangle$ , with  $2 < k \leq n$ .

Since  $\mu$  is an improving deal, every agent in  $\mu$  is happy to exchange with the agent coming before her in  $\mu$ :

$$\pi_{a_{i-1}} \succ_i \pi_{a_i}, \forall a_i \in \mu. \tag{1}$$

Moreover, as there exists no improving swap-deal in  $\pi$  ( $C_2(\pi) = \emptyset$ ) agents do not want to exchange with the agent coming after them in  $\mu$ :

$$\pi_{a_i} \succ_i \pi_{a_{i+1}}, \forall a_i \in \mu. \tag{2}$$

Indeed, because of (1), an improving swap-deal would otherwise exist in  $\pi$ .

We now show by induction on the size of  $\mu$ , denoted by  $k$ , that such an improving deal cannot exist.

If  $k = 3$ , consider without loss of generality that  $\mu = \langle a_1, a_2, a_3 \rangle$ . From (1) and (2), we obtain the following profile where underlined resources indicate the initial allocation:

$$\begin{aligned} a_1 : \pi_{a_3} \succ_1 \underline{\pi_{a_1}} \succ_1 \pi_{a_2}, \\ a_2 : \pi_{a_1} \succ_2 \underline{\pi_{a_2}} \succ_2 \pi_{a_3}, \\ a_3 : \pi_{a_2} \succ_3 \underline{\pi_{a_3}} \succ_3 \pi_{a_1}. \end{aligned}$$

The triplet of resources  $\langle \pi_{a_1}, \pi_{a_2}, \pi_{a_3} \rangle$  is thus a witness of the violation of the worst-restrictedness condition for a profile to be single-peaked (Proposition 1). Indeed, all the three resources are ranked last by an agent when we restrict our attention to these resources. The contradiction is thus established.

Suppose now that  $\pi$  is stable with respect to  $C_{k-1}$ . We will show that no improving deal of size  $k$  exists in  $\pi$ . From the induction hypothesis, we get that:

$$\pi_{a_i} \succ_i \pi_{a_j}, \forall a_i, a_j \in \mu, a_j \notin \{a_{i-1}, a_i\}. \tag{3}$$

Indeed there would otherwise exist two agents  $a_l$  and  $a'_l$ , that are not next to one another in  $\mu$ , such that  $\pi_{a'_l} \succ_l \pi_{a_l}$ . It would then have been possible to “cut”  $\mu$  between those two agents so that  $a_l$  receives  $\pi_{a'_l}$ . The new cycle-deal obtained would also have been improving and then an improving deal of size strictly smaller than  $k$  would exist.

Because the profile is single-peaked, it is also worst-restricted (Proposition 1) and there exist thus at most two resources ranked last by an agent among the ones appearing in  $\mu$ . Call  $\pi_{a_w}$  one such resource holds by agent  $a_w$ , and consider the triplet of resources  $R = \langle \pi_{a_{w-1}}, \pi_{a_w}, \pi_{a_{w+1}} \rangle$ . From (1), (2) and (3) we get:

$$\begin{aligned} a_w : \pi_{a_{w-1}} \succ_w \underline{\pi_{a_w}} \succ_w \pi_{a_{w+1}}, \\ a_{w+1} : \pi_{a_w} \succ_{w+1} \underline{\pi_{a_{w+1}}} \succ_{w+1} \pi_{a_{w-1}}. \end{aligned}$$

Hence when restricting preferences to  $R$ , for every resource in  $R$ , there exists an agent ranking it last among the resources in  $R$ . This violates the worst-restrictedness condition for the single-peaked profile and sets the contradiction.

Overall we have proved that no improving deal exists in  $\pi$  which entails that  $\pi$  is stable with respect to  $C_n$ . □

This theorem states that the  $C_k$ -stability hierarchy collapses at the  $C_2$  level in single-peaked housing markets. Remember that every Pareto-optimal allocation is stable with respect to  $C_n$ . This result provides then a new characterization of Pareto-optimality in our setting.

**Corollary 1** *In a single-peaked housing market, an allocation  $\pi$  is Pareto-optimal if and only if it is stable with respect to  $C_2$ .*

Stating this result in terms of stability with respect to  $C_n$  and not just Pareto-optimality gives us more flexibility. Indeed, in the more general setting where there are more resources to allocate than the number of agents, the result of Theorem 1 still holds [11] but it is no longer the case that Pareto-optimality implies  $C_n$ -stability.

As we have proven that the allocation reached by swap-deals is Pareto-optimal, a natural question is then whether every allocation that Pareto-dominates the initial allocation can be reached by swap-deals. It is not the case.

**Proposition 4** *There exists an instance  $I = \langle \mathcal{N}, \mathcal{R}, L, \pi^0 \rangle$  for which there is an allocation  $\pi$  that Pareto-dominates  $\pi^0$  and that can not be reached by a sequence of improving swap-deals.*

**Proof** Let us consider the following instance with three agents where the initial allocation  $\pi^0 = \langle \underline{r_3}, r_2, r_1 \rangle$  is the one underlined.

$$\begin{aligned}
 a_1 : r_1 & \succ_1 \boxed{r_2} \succ_1 \underline{r_3} \\
 a_2 : \boxed{r_1} & \succ_2 \underline{r_2} \succ_2 r_3 \\
 a_3 : r_2 & \succ_3 \boxed{r_3} \succ_3 \underline{r_1}
 \end{aligned}$$

The squared allocation  $\boxed{\pi} = \langle r_2, r_1, r_3 \rangle$  is Pareto-optimal. However from  $\pi^0$  only two deals are possible:  $\mu_1 = \langle a_1, a_3 \rangle$  that reaches allocation  $\pi' = \langle r_1, r_2, r_3 \rangle$ , or  $\mu_2 = \langle a_2, a_3 \rangle$  that leads to  $\pi'' = \langle r_3, r_1, r_2 \rangle$ . No sequence of improving swap-deals can thus reach  $\pi$ . □

It is however interesting to note that both the allocation returned by TTC and by Crawler can always be reached through improving swap-deals.

**Proposition 5** *Let  $I = \langle \mathcal{N}, \mathcal{R}, L, \pi^0 \rangle$  be an instance and let  $\pi^{TTC}$  be the allocation returned by the TTC algorithm on  $I$ . Then  $\pi^{TTC}$  is reachable by swap-deals from  $\pi^0$ .*

**Proof** We show that any cycle-deal applied by TTC can be implemented as a sequence of improving swap-deals. For a given step of the algorithm, we consider  $R$  the set of resources and  $N$  the set of agents remaining at this step and  $\pi$  the current partial allocation.

Let  $\mu$  be the next cycle-deal to be applied, and let  $R_\mu$  and  $N_\mu$  be respectively the set of resources and the set of agents involved in  $\mu$ . We must prove that the allocation  $\pi[\mu]$  can be reached by a sequence of swap-deals. When  $|R_\mu| = 2$ ,  $\mu$  is actually a swap-deal and the claim is trivially true.

Assume then that  $|R_\mu| > 2$ , let us consider the instance  $I'$  obtained from  $I$  by restricting  $\mathcal{R}$  to  $R_\mu$  and  $\mathcal{N}$  to  $N_\mu$ . In this restricted instance, the allocation  $\pi^*$  where each agent receives her top resource is feasible. Indeed, it is the allocation obtained by applying  $\mu$  (by definition of  $\mu$  in TTC). This allocation is trivially Pareto-optimal as every agent has her top resources and it thus Pareto-dominates every other allocation. It is then the only Pareto-optimal allocation in  $I'$ . Remember from Observation 1 that  $I'$  is single-peaked. Then, by virtue of Theorem 1, this implies that  $\pi^*$  must be reachable by improving swap-deals in  $I'$ . Overall,  $\pi_\mu$  is then reachable by improving swap-deals in  $I$ .

The same argument can be stated for any step of the algorithm. By concatenating the sequences of improving swap-deals for every step of the algorithm, we then obtain a sequence of improving swap-deals leading to  $\pi^{TTC}$  from  $\pi^0$ .  $\square$

**Proposition 6** *Let  $I = \langle \mathcal{N}, \mathcal{R}, L, \pi^0 \rangle$  be an instance and let  $\pi^C$  be the allocation returned by Crawler. Then  $\pi^C$  is reachable by swap-deals from  $\pi^0$ .*

**Proof** We show that every cycle-deal applied by Crawler can be implemented as a sequence of improving swap-deals. For clarity reasons and without loss of generality, we assume that each agent  $a_j$  currently holds resource  $r_j$ .

For a given iteration of the algorithm, consider agent  $a_i$  who picks resource  $r_k$  currently held by agent  $a_k$ . From the definition of the procedure,  $a_k$  is on the left of  $a_i$  (with respect to the single-peaked axis) and  $a_k$  has already been considered at this step before considering  $a_i$ . In fact, all the agents between  $a_k$  (included) and  $a_i$  (excluded) on the left of the single-peaked axis have already been considered at the current iteration before reaching  $a_i$ . Moreover, all these agents have passed their turn because their peak is on their right. In other words, each agent  $a_j$  between  $a_k$  (included) and  $a_i$  (excluded) prefers the resource held by the agent  $a_{j+1}$  on her right, that is:

$$r_{j+1} \succ_j r_j, \forall j \in \{k, \dots, i-1\}.$$

Let  $\mu = \langle a_i, a_{i-1}, \dots, a_{k+1}, a_k \rangle$  be the cycle-deal implemented by Crawler at the current step. In this deal,  $a_k$  gives her resource  $r_k$  to  $a_i$  and all the other agents of the deal give their resource to the next agent in the sequence which is the agent on their left with respect to the single-peaked axis. The decomposition of  $\mu$  into a sequence of swap-deals consists in using agent  $a_i$  as a hub for the swap-deals. Agent  $a_i$  first exchanges with  $a_{i-1}$  then,  $a_i$  exchanges with  $a_{i-2}$  and so on until  $a_i$  performs a swap-deal with  $a_k$ . At the end,  $a_i$  holds  $r_k$  and each other agent  $a_j$  in  $\mu_i$  holds the resource initially held by  $a_{j+1}$ . The sequence of swap-deals is thus equivalent to  $\mu_i$ .

We now show that all these swap-deals are improving. In the first deal  $\langle a_i, a_{i-1} \rangle$ ,  $a_{i-1}$  receives the resource  $r_i$  held by  $a_i$  that she prefers to her current resource  $r_{i-1}$  (as shown previously  $r_i \succ_{i-1} r_{i-1}$ ). Simultaneously, agent  $a_i$  receives the resource  $r_{i-1}$  held by  $a_{i-1}$  that she prefers to her current resource since her peak is on the left of  $a_{i-1}$  (it is held by  $a_k$ ), that is:

$$r_j \succ_i r_{j+1}, \forall j \in \{k, \dots, i-1\}.$$

The first deal is thus mutually improving. Concerning the next swap-deal  $\langle a_i, a_j \rangle$  with  $j \in \{k, \dots, i-2\}$ ,  $a_i$  exchanges  $r_{j+1}$  that she obtained from her previous swap-deal, against

$r_j$  held by  $a_j$ . Agent  $a_j$  receives the resource  $r_{j+1}$  that she prefers to her current resource  $r_j$  (as shown previously  $r_{j+1} \succ_j r_j$ ).  $a_i$  receives the resource  $r_j$  that she prefers to  $r_{j+1}$  (since  $\forall j \in \{k, \dots, i-1\}, r_j \succ_i r_{j+1}$ ). All the swap-deals are thus mutually improving.  $\square$

These two results show that both the outcome of TTC and Crawler could be implemented as sequences of mutually beneficial swap-deals in single-peaked domains. This attenuates a bit a critical feature of these procedures. However, even though these long trading cycles can be broken down into swap-deals, they would still need to be pre-computed, and carefully coordinated.

The proof of Proposition 6 shows how to compute the set of swap-deals that achieves Crawler outcome: for each cycle-deals  $\mu = \langle a_i, a_{i-1}, \dots, a_{k+1}, a_k \rangle$  implemented,  $a_i$  is used as a hub to decompose the deal into mutually beneficial swap-deals. On the contrary, the proof of Proposition 5 does not provide explicitly the corresponding sequence of deals. We show now how to compute the set of swap-deals that achieves the TTC outcome using Crawler. Note that for each cycle-deal selected by TTC, we can build a sub-instance of the initial instance and apply Crawler on this sub-instance to obtain the set of swap-deals that implements the cycle-deal. Since Crawler is Pareto-optimal, the following observation is straightforward:

**Observation 3** *In a single-peaked housing market, if the allocation where each agent has her top resource is feasible (i.e. each agent has a different top resource), then the allocation is returned by Crawler.*

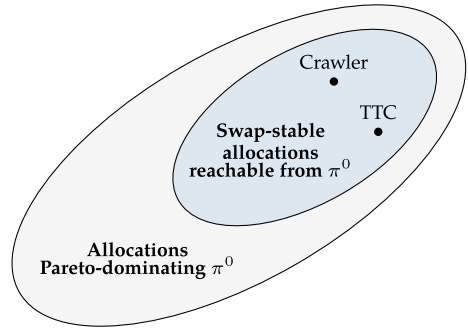
We can now compute the set of swap-deals that achieves the TTC outcome using Crawler. For each deal  $\mu$  implemented by TTC:

- If  $|\mu| \leq 2$ , it is already a swap-deal and we are done.
- If  $|\mu| > 2$ , the deal has to be decomposed into a set of swap-deals. Let  $I'$  be the sub-instance restricted to the agents and the resources involved in  $\mu$ . This instance is guaranteed to remain single-peaked (see Observation 2). By definition of TTC, in  $I'$ , each agent obtains her top resource. By Observation 3, the Crawler outcome will be the same as the outcome of  $\mu$  when applying Crawler to  $I'$ . The swap-deal decomposition of  $\mu$  is then obtained by applying Crawler on  $I'$  and decomposing each cycle-deal  $\mu'$  of Crawler as explained above.

Again, it should be kept in mind that in practice, a coordination mechanism would have to ensure that the agents indeed execute the correct sequence of swap-deals (and thus the desired allocation would be obtained).

Figure 1 concludes this section by summarizing the different findings related the properties of swap dynamics.

**Fig. 1** Summary of the results of Theorem 1 and Propositions 4, 5 and 6 for a given initial allocation  $\pi^0$ . Remember from Example 2 that TTC and Crawler can return different allocations



### 6.3 Maximality of the single-peaked domain

We now show that the single-peaked domain is maximal for the Pareto-optimality of swap dynamics: For every preference domain  $D$  such that  $\mathcal{SP}_{\triangleleft}$  is a proper subset of  $D$  (for a given linear order  $\triangleleft$ ) there exists an instance such that no swap dynamics can reach a Pareto-optimal allocation.

Before going through the proof, let us first start with a simple example. Consider the following profile and let  $\triangleleft$  be the order  $r_1 \triangleleft r_2 \triangleleft r_3$ .

$$\begin{aligned}
 a_1 &: r_1 \succ_1 \underline{r_2} \succ_1 r_3 \\
 a_2 &: r_3 \succ_2 \underline{r_1} \succ_2 r_2 \\
 a_3 &: r_2 \succ_3 \underline{r_3} \succ_3 r_1
 \end{aligned}$$

It is clear that the profile is not single-peaked with respect to  $\triangleleft$  (it actually is not for any order) as it is clearly not worst-restricted. Note that the preferences of agents  $a_1$  and  $a_3$  are single-peaked with respect to  $\triangleleft$  but not those of agent  $a_2$ .

Take the underlined resources to form the initial allocation. It has been constructed as follows. Agents  $a_1$  and  $a_3$  receive their second best resource (respectively  $r_2$  and  $r_3$ ). For agent  $a_2$ , the violation of single-peakedness comes from her ranking of resources  $r_1$  and  $r_2$ . She is thus allocated her most preferred resource among these two, i.e.  $r_1$ . It can easily be checked that this allocation is stable with respect to  $C_2$  but is Pareto-dominated by  $\langle r_1, r_3, r_2 \rangle$ .

This construction is generalized to an arbitrary number of agents below.

**Theorem 2** *Let  $\mathcal{R}$  be a set of resources and  $\triangleleft$  a linear order over  $\mathcal{R}$ . For every preference domain  $D$  such that  $\mathcal{SP}_{\triangleleft} \subsetneq D$ , there exists an instance  $I = \langle \mathcal{N}, \mathcal{R}, L, \pi^0 \rangle$  defined over  $D$  such that swap dynamics on  $I$  do not reach a Pareto-optimal allocation.*

**Proof** Let us construct an instance  $I = \langle \mathcal{N}, \mathcal{R}, L, \pi^0 \rangle$  defined over  $D$  such that swap dynamics on  $I$  do not reach a Pareto-optimal allocation.

Without loss of generality and for the ease of the reader, let us assume that  $\triangleleft$  is such that  $r_1 \triangleleft r_2 \triangleleft \dots \triangleleft r_n$ .

Since  $\mathcal{SP}_{\triangleleft} \subsetneq D$ , there exists a linear order  $\succ^* \in D$  that is not single-peaked with respect to  $\triangleleft$ . There exist then two resources  $(r_{s-1}, r_s) \in \mathcal{R}^2$  such that:

$$\left. \begin{array}{l} r_s \triangleleft r_{s-1} \triangleleft \text{top}(\succ^*), \\ \text{or, } \text{top}(\succ^*) \triangleleft r_{s-1} \triangleleft r_s \end{array} \right\} \text{ and } r_s \succ_i^* r_{s-1}.$$

Without loss of generality, let us assume that  $\text{top}(\succ^*) \triangleleft r_{s-1} \triangleleft r_s$ . We will use index  $t$  to refer to the top ranked resource in  $\succ^*$ , i.e.,  $r_t = \text{top}(\succ^*)$ .

Moreover, as  $SP_{\triangleleft} \subset D$ , for every resource  $r_i \in \mathcal{R}$  there exist in  $D$  two linear orders  $\succ_i^1$  and  $\succ_i^2$  that are single-peaked with respect to  $\triangleleft$  such that:

$$\text{top}(\succ_i^1) = \text{top}(\succ_i^2) = r_i \text{ and } \text{snd}(\succ_i^2) = r_{i-1}.$$

Thus, in the preference order  $\succ_i^1$  the top resource is  $r_i$  and the other resources are not constrained. For  $\succ_i^2$ , the top resource is  $r_i$  and the second best must be  $r_{i-1}$ .

We introduce the preference profile  $L = \{\succ_i \mid a_i \in \mathcal{N}\}$  defined as follows:

$$\begin{aligned} \succ_1 &= \succ^*, \\ \succ_i &= \succ_{i-1}^1, \quad \forall a_i \in \mathcal{N}, i \in \{2, \dots, t\}, \\ \succ_i &= \succ_i^2, \quad \forall a_i \in \mathcal{N}, i \in \{t + 1, \dots, s\}, \\ \succ_i &= \succ_i^1, \quad \forall a_i \in \mathcal{N}, i \in \{s + 1, \dots, n\}. \end{aligned}$$

The initial allocation  $\pi^0$  is then defined as:

$$\begin{aligned} \pi_{a_1}^0 &= r_s, \\ \pi_{a_i}^0 &= \text{top}(a_i) = r_{i-1}, \quad \forall a_i \in \mathcal{N}, i \in \{2, \dots, t\}, \\ \pi_{a_i}^0 &= \text{snd}(a_i) = r_{i-1}, \quad \forall a_i \in \mathcal{N}, i \in \{t + 1, \dots, s\}, \\ \pi_{a_i}^0 &= \text{top}(a_i) = r_i, \quad \forall a_i \in \mathcal{N}, i \in \{s + 1, \dots, n\}. \end{aligned}$$

To get a better understanding of the instance  $I = \langle \mathcal{N}, \mathcal{R}, L, \pi^0 \rangle$  constructed in this proof, Fig. 2 presents the preference profile  $L$  and the initial allocation  $\pi^0$ .

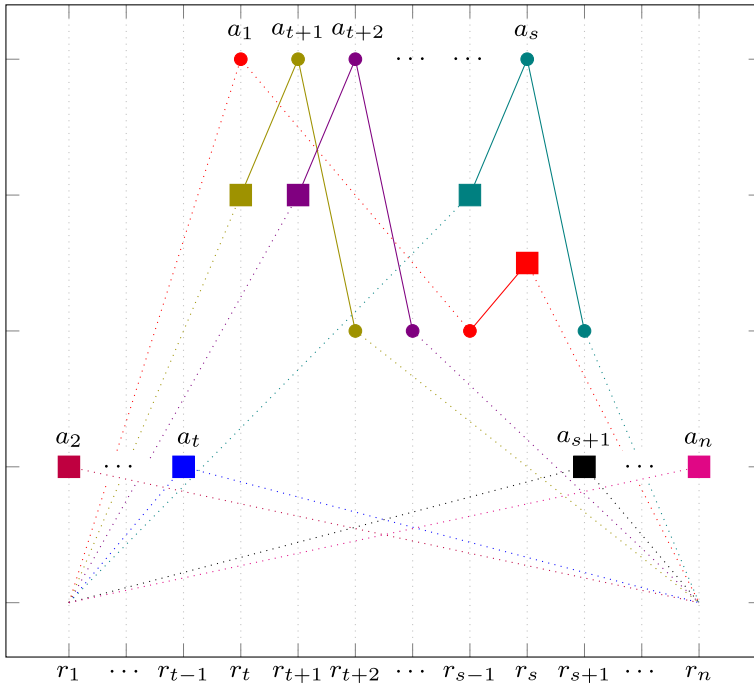
We claim that  $\pi^0$  is stable with respect to  $C_2$  but not Pareto-optimal. Allocation  $\pi^0$  is clearly not Pareto-optimal as the allocation in which every agent receives her top resource is feasible—because no two agents have the same top resource—and this allocation clearly Pareto-dominates  $\pi^0$ . Let us now show that  $C_2(\pi^0) = \emptyset$ .

First observe that every agent  $a_i, i \in \{2, \dots, t\} \cup \{s + 1, \dots, n\}$  owns her top resource, hence can not be involved in an improving swap-deal.

Consider now agent  $a_i, i \in \{t + 1, \dots, s - 1\}$ . She owns her second most preferred resource which she would only trade against her top resource that is owned by agent  $a_{i+1}$ . However, agent  $a_{i+1}$  is not interested in  $\pi_{a_i}$ , hence no improving swap-deal is possible.

Finally, let us consider agent  $a_s$  who owns her second most preferred resource  $r_{s-1}$  and whose top resource  $r_s$  is owned by agent  $a_1$ . By the hypothesis that  $a_1$ 's preferences are not single-peaked, we have  $r_s \succ_1 r_{s-1}$ . Once again, there is no improving swap-deal involving agent  $a_s$ .

Overall, we have proved that  $C_2(\pi^0) = \emptyset$ , hence any swap dynamic returns  $\pi^0$  on  $I$ . As  $\pi^0$  is Pareto-dominated, this concludes the proof. □



**Fig. 2** The instance constructed in the proof of Theorem 2. The dotted lines represent the trend of the utilities, dots fixed points and squares the resource allocated to each agent

This result is particularly interesting since it shows that the single-peaked domain captures in a “tight” way the domain under which swap dynamics are Pareto-optimal (in the vein of similar results obtained by [21] in different settings).

Note that this is not a characterization result: there are some domains that are not single-peaked but for which swap dynamics return Pareto-optimal allocations.

**Example 6** Let us consider the following preference profile:

$$a_1 : r_1 \succ_1 r_2 \succ_1 r_3$$

$$a_2 : r_1 \succ_2 r_3 \succ_2 r_2$$

$$a_3 : r_3 \succ_3 r_2 \succ_3 r_1$$

This profile is not single-peaked over any linear order  $\triangleleft$ : the triplet  $\langle r_1, r_2, r_3 \rangle$  is a witness of the violation of the worst-restrictedness condition (Proposition 1), however we can show that for every initial allocation, swap dynamics return a Pareto-optimal allocation. Let us consider the different initial allocations:

1.  $\pi^0 = \langle r_1, r_2, r_3 \rangle$ :  $C_2(\pi^0) = \emptyset$  and  $\pi^0$  is Pareto-optimal.
2.  $\pi^0 = \langle r_1, r_3, r_2 \rangle$ :  $C_2(\pi^0) = \emptyset$  and  $\pi^0$  is Pareto-optimal.
3.  $\pi^0 = \langle r_2, r_1, r_3 \rangle$ :  $C_2(\pi^0) = \emptyset$  and  $\pi^0$  is Pareto-optimal.
4.  $\pi^0 = \langle r_2, r_3, r_1 \rangle$ : two swap-deals are possible:

- (a)  $\langle a_1, a_3 \rangle$  which leads to case 2.
  - (b)  $\langle a_2, a_3 \rangle$  which leads to case 3.
5.  $\pi^0 = \langle r_3, r_1, r_2 \rangle$ : one swap-deal is possible:  $\langle a_1, a_3 \rangle$  which leads to case 3.
6.  $\pi^0 = \langle r_3, r_2, r_1 \rangle$ : three swap-deals are possible:
- (a)  $\langle a_1, a_3 \rangle$  which leads to case 1.
  - (b)  $\langle a_1, a_2 \rangle$  which leads to case 4.
  - (c)  $\langle a_2, a_3 \rangle$  which leads to case 5.

This example does not contradict Theorem 2: Swap dynamics are Pareto-optimal on the domain  $D = \{>_1, >_2, >_3\}$  because  $D$  does not include every single-peaked linear order over  $\triangleleft$ . For instance take  $\triangleleft$  to be  $r_1 \triangleleft r_2 \triangleleft r_3$ , extend  $D$  by adding the two following linear orders:  $r_2 > r_1 > r_3$  and  $r_2 > r_3 > r_1$  so that  $SP_{\triangleleft} \subset D'$  and Theorem 2 will apply.

### 7 The “price of” swap dynamics

It is natural to now ask to what extent swap dynamics induce a cost in terms of social welfare. In that perspective we will first discuss the “standard” *price of anarchy* ( $PoA$ ) [3, 38], that is, the (worst-case, over all instances) ratio between the worst stable outcome and the social welfare optimum. More formally, for an allocation procedure  $M$ :

$$PoA_v(M) = \max_{I \in \mathcal{I}_D} \frac{\max_{\pi} v(\pi)}{\min_{\pi \in M(I)} v(\pi)}, \text{ with } v \in \{ark, mrk\}$$

But, as we have discussed, swap dynamics constitute a family of procedures. In that context, a perhaps more relevant metric is the ratio between the best and worst stable outcome which can be obtained by such procedures. In particular, this would tell us the price to pay for not being guided by a central planner (in the selection of deals) towards maximizing our social welfare notion. We call this notion the *price of dynamics* ( $PoD$ ), and for a family of dynamics  $\mathcal{M}$ , we define it as follows:

$$PoD_v(\mathcal{M}) = \max_{\substack{I \in \mathcal{I}_D \\ M, M' \in \mathcal{M}}} \frac{\max_{\pi \in M'(I)} v(\pi)}{\min_{\pi \in M(I)} v(\pi)}, \text{ with } v \in \{ark, mrk\}.$$

Note that this notion is not specific to swap dynamics and can be applied to any family of dynamics. Also, observe that an upper bound on the  $PoA$  also applies to the  $PoD$ .

These definitions are parametrized by the social welfare notion considered. In terms of average rank, we recall that [24] established that the price of anarchy is 2 for swap dynamics *in the general domain*. While the upper bound remains valid in our restricted domain,



the instance exhibited to show this bound to be tight in [24] violates single-peakedness. We now show that this result still holds under our domain restriction.

**Proposition 7** *For any  $M \in \mathcal{M}_2$ ,  $PoA_{ark}(M)$  is 2 in the single-peaked domain.*

**Proof** Let us consider the following single-peaked instance involving  $n$  agents, and assume without loss of generality  $n$  to be odd. Take for now the shaded allocation as the initial allocation.<sup>3</sup>

$$\begin{array}{cccccccccccc}
 a_1 : & \boxed{r_{n-1}} & \succ & r_n & \succ & r_{n-2} & \succ & r_{n-3} & \cdots & r_3 & \succ & \underline{r_2} & \succ & r_1 \\
 a_2 : & \boxed{r_1} & \succ & \boxed{r_2} & \succ & \underline{r_3} & \succ & r_4 & \cdots & r_{n-2} & \succ & r_{n-1} & \succ & r_n \\
 a_3 : & \boxed{r_2} & \succ & r_1 & \succ & \boxed{r_3} & \succ & \underline{r_4} & \cdots & r_{n-2} & \succ & r_{n-1} & \succ & r_n \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 a_{n-2} : & \boxed{r_{n-3}} & \succ & r_{n-4} & \succ & r_{n-5} & \succ & r_{n-4} & \cdots & \boxed{r_{n-2}} & \succ & \underline{r_{n-1}} & \succ & r_n \\
 a_{n-1} : & \boxed{r_{n-2}} & \succ & r_{n-3} & \succ & r_{n-4} & \succ & r_{n-5} & \cdots & r_{n-1} & \succ & \boxed{r_n} & \succ & \underline{r_1} \\
 a_n : & \boxed{r_1} & \succ & r_2 & \succ & r_3 & \succ & r_4 & \cdots & r_{n-2} & \succ & r_{n-1} & \succ & \boxed{r_n}
 \end{array}$$

Let  $\pi$  be the shaded allocation, and  $\pi^*$  be the squared allocation ( $a_1$  holds  $r_{n-1}$  in both cases). Observe that both  $\pi$  and  $\pi^*$  are Pareto-optimal allocations. For  $\pi^*$  this is obvious as only  $a_n$  does not hold her top resource. For  $\pi$ , notice that  $a_2$  would only wish to swap for  $r_1$ , which is held by an agent who ranks it first. Then  $a_3$  would only wish to swap for  $r_1$  or  $r_2$ , and so on until  $a_{n-2}$ . Finally,  $a_{n-1}$  may swap with anyone but  $a_n$ , but no one wants to swap with her.

We see that  $ark(\pi^*) = [(n - 1) \cdot n + 1]/n$ , while  $ark(\pi) = [n \cdot (n + 1)/2 + n - 1]/n$ , thus this instance shows that (asymptotically) the  $PoA$  is at least 2. In [24] it is shown that 2 is an upper bound for the  $PoA$  of any swap-deal procedure. We thus conclude that the  $PoA$  is 2 here, as in the general domain.  $\square$

Following [24], we can also make some further observations. In fact, as both allocations are Pareto-optimal, this instance shows that the  $PoA$  of any procedure satisfying individual rationality must be at least 2. Indeed, taking the shaded allocation  $\pi$  as the initial allocation, any individually rational procedure would output  $\pi$ . Furthermore, as any Pareto-optimal allocation is stable for swap-deals, our upper bound remains valid.

**Observation 4** *For any individually rational allocation procedure  $M$ ,  $PoA_{ark}(M)$  is 2 in the single-peaked domain. This holds in particular for TTC and Crawler.*

To address the price of dynamics, we must now make use of a different initial allocation, and show that both allocations are reachable by sequences of swaps. This would demonstrate that this gap could also only be due to the selection heuristics.

**Proposition 8** *For the family of swap dynamics  $\mathcal{M}_2$ ,  $PoD_{ark}(\mathcal{M}_2)$  is 2 in the single-peaked domain.*

<sup>3</sup> The underlined allocation will only be relevant in forthcoming proofs.

**Proof** Consider again the instance presented in the proof of Proposition 8. Take now the initial allocation to be  $\pi^0 = \langle r_2, r_3, r_4, \dots, r_{n-1}, r_1, r_n \rangle$ , the underlined allocation. We first show that the shaded allocation  $\pi$  can be reached from  $\pi^0$  via improving swap-deals. First the deal  $\langle a_n, a_{n-1} \rangle$  is implemented. Then  $a_1$  acts as a hub, and the following sequence is implemented  $\langle a_1, a_2 \rangle \dots \langle a_1, a_{n-2} \rangle$ .

Now, the squared allocation  $\pi^*$  can also be reached with swap-deals. Agent  $a_1$  will act as a hub for odd agents, while  $a_{n-1}$  will do the same for even agents. In other words, the following sequence is implemented:  $\langle a_1, a_3 \rangle, \langle a_1, a_5 \rangle \dots \langle a_1, a_{n-2} \rangle$  (note that  $a_n$  gets her final resource from the start), followed by the sequence of swaps  $\langle a_{n-1}, a_2 \rangle, \langle a_{n-1}, a_4 \rangle \dots \langle a_{n-1}, a_{n-3} \rangle$ .

The result follows then from the computations of the average rank presented in Proposition 8. □

Finally, we turn our attention to the minimum rank counterparts of the same notions. Note that  $n$  is certainly an upper bound in that case, as this is highest possible ratio between two allocations.

**Proposition 9** For any  $M \in \mathcal{M}_2$ ,  $PoA_{mrk}(M)$  is  $\Theta(n)$  in the single-peaked domain.

**Proof** Let us consider the following single-peaked instance involving  $n$  agents. Take the shaded allocation as the initial allocation.<sup>4</sup> Let  $\pi$  be the shaded allocation, and  $\pi^*$  be the squared allocation.

$a_1 :$	$r_1$	$\succ$	$r_2$	$\succ$	$r_3$	$\succ$	$r_4$	$\dots$	$r_{n-1}$	$\succ$	$r_n$
$a_2 :$	$r_2$	$\succ$	$r_3$	$\succ$	$r_1$	$\succ$	$r_4$	$\dots$	$r_{n-1}$	$\succ$	$r_n$
$a_3 :$	$r_3$	$\succ$	$r_4$	$\succ$	$r_2$	$\succ$	$r_1$	$\dots$	$r_{n-1}$	$\succ$	$r_n$
$a_4 :$	$r_4$	$\succ$	$r_5$	$\succ$	$r_3$	$\succ$	$r_2$	$\dots$	$r_{n-1}$	$\succ$	$r_n$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$a_n :$	$r_2$	$\succ$	$r_1$	$\succ$	$r_3$	$\succ$	$r_4$	$\dots$	$r_{n-1}$	$\succ$	$r_n$

It is easy to see that  $mrk(\pi^*) = n - 1$ , while  $mrk(\pi) = 1$ , thus  $PoA$  is  $\Omega(n)$ . Clearly, the  $PoA$  cannot be worse, thus it is  $\Theta(n)$ . □

**Proposition 10**  $PoD_{mrk}(\mathcal{M}_2)$  is  $\Theta(n)$  in the single-peaked domain.

**Proof** Let us consider the single-peaked instance of Proposition 9 and let us assume that the initial allocation is now  $\pi^0 = \langle r_{n-1}, r_1, r_2, \dots, r_{n-2}, r_n \rangle$ , the underlined allocation.

From this initial allocation  $\pi^0$ , it is possible to reach the shaded allocation  $\pi = \langle r_1, r_2, r_3, \dots, r_{n-1}, r_n \rangle$  with improving swap deals. This allocation is reached by performing  $n - 2$  swap-deals  $\langle a_1, a_i \rangle, \forall i \in \{n - 1, \dots, 2\}$  starting with  $\langle a_1, a_{n-1} \rangle$  and finishing with  $\langle a_1, a_2 \rangle$ .

From  $\pi^0$ , it is also possible to reach  $\pi^* = \langle r_1, r_3, r_4, \dots, r_n, r_2 \rangle$  (the squared allocation) which is both Pareto-optimal and optimal in terms of minimum rank. If  $n$  is even,

<sup>4</sup> Again, the underlined allocation will only be relevant in a later proof.

we perform the following sequence of couples of swap-deals  $\langle a_{i-1}, a_n \rangle$  and  $\langle a_{i-2}, a_1 \rangle$   $\forall i \in \{n, n-2, \dots, 6, 4\}$ . If  $n$  is odd, we perform the same sequence of couples of swap-deals but  $\forall i \in \{n, n-2, \dots, 7, 5\}$  and add two final swap-deals  $\langle a_2, a_n \rangle$  and  $\langle a_1, a_n \rangle$ .

The result follows from the previous computations of the minimum rank in these two allocations.  $\square$

These different theoretical results remain a worst-case analysis. To complete the picture, it would be valuable to know how well these swap dynamics do in practice. This is what we investigate in the next section.

## 8 Experimental study

Our objective in this section is to get some insights about the empirical behaviour of swap dynamics. Our study includes swap dynamics based on the different history-based selection heuristics introduced in Sect. 5, as well as two other selection heuristics added for comparison: one which takes as input more than the mere history of deals ( $M_2$ -PW), and another one which relaxes the constraint of using swaps only ( $M_3$ -U). For completeness, we also compare their performance to that of TTC and Crawler (keeping in mind though that these two centralized procedures are not specifically designed to optimize our social welfare measures).

We now give the full detail of our protocol by specifying (1) how deals are selected in the swap dynamics, (2) how preferences are generated, as well as (3) the full specification of the parameters used in the experiments.

### 8.1 Experimental protocol

#### 8.1.1 Selection heuristics for swap dynamics

Concerning swap-deal procedures, our study includes the history-based selection heuristics already introduced in Sect. 5:  $M_2$ -RRA,  $M_2$ -RRP,  $M_2$ -RRR,  $M_2$ -U and  $M_2$ -RM. We also include for comparison a preference-based heuristics ( $M_2$ -PW) and a procedure allowing cycles involving up to 3 agents ( $M_3$ ). More precisely:

- *Priority to the worst-off agent* ( $M_2$ -PW): agents are ordered considering the rank of the resource they own from the one with the lowest rank to the one with the highest rank. Agents are then paired in a round-robin fashion like  $M_2$ -RRA does following this ordering. Note that  $M_2$ -PW is more demanding than history-based heuristics as it requires some information about the agent's current rank which can only be collected via a central entity.
- *Uniform up to three agents* ( $M_3$ -U): a deal is selected uniformly at random among all possible deals involving 2 or 3 agents. If the deal is rational for all the agents involved, it is implemented.

### 8.1.2 Generation of single-peaked preferences

Different methods can be envisioned to generate single-peaked preferences. We consider impartial culture for single-peaked domain (IC-SP) and uniform peak for the single-peaked domain (UP-SP).

Single-peaked preferences under impartial culture (IC-SP) are drawn using the method proposed by [66]. Given an axis, single-peaked preferences are built recursively from the end (i.e. the worst resource of the agent) to the top resource. At each iteration, the next resource in the preference order is randomly selected between the two extremes of the axis. The selected resource is then removed from the axis and so on until the axis is empty.

In the uniform peak culture (UP-SP), presented by [23], preferences are constructed by first picking uniformly at random a resource to be the peak. The second-highest ranked resource is chosen with equal probability from the two adjacent alternatives, and so on until a full order is obtained.

As already mentioned by [66], the probabilities of the preference orders significantly differ from one method to another. Under IC-SP, each single-peaked preference order has a uniform probability  $1/2^{n-1}$  to be selected. On the contrary, under UP-SP, probabilities over preference orders are not uniform. In fact, the peak is uniformly drawn (with probability  $1/n$ ) and single-peaked preferences are then built from this peak. Since there is only one preference order with its peak at one end of the axis, these orders are more likely to be drawn than preference orders with a peak in the middle of the axis, for instance.

Let us consider the case where  $n = 5$  and the axis is  $r_1 \triangleleft r_2 \triangleleft r_3 \triangleleft r_4 \triangleleft r_5$ . Preference orders  $r_1 \succ r_2 \succ r_3 \succ r_4 \succ r_5$  and  $r_5 \succ r_4 \succ r_3 \succ r_2 \succ r_1$  both have a probability  $1/5$  to be generated whereas the probability to generate some preference order with the peak  $r_3$  is  $1/5$ , and there are 6 orders with this peak. Under IC-SP, the probability to generate the preference order  $r_1 \succ r_2 \succ r_3 \succ r_4 \succ r_5$  is  $1/16$ .

Closer inspection reveals that the frequency of each rank for a given resource is more evenly distributed with UP-SP than with IC-SP. For instance, under IC-SP, resources on the end-sides of the single-peaked axis have a high probability to be ranked among the top resources of an agent but they have a very low probability to be ranked in the second half of the preference linear order. Under UP-SP, these resources have a high probability to be selected as the top resource of an agent but they also have a more uniform probability distribution among all other possible ranks. A resource on the end-side of the single-peaked axis is thus more likely to obtain a low ranking under UP-SP.

This suggests that profiles under UP-SP are more diverse than under IC-SP. To quantify this more precisely, we have computed two diversity indices suggested by [33]: the sum of Kendall's tau and the sum of Spearman distances.<sup>5</sup> The results confirm that UP-SP generates more diverse profiles on average compared to IC-SP. Take for instance the case of seven resources ( $n = 7$ ). We generated 10000 preference profiles under both UP-SP and IC-SP. With respect to the Kendall's tau index, profiles under UP-SP achieved an average diversity score of 75% (normalized over the empirically observed maximum) while this value reaches only 48% in the case of IC-SP. With the Spearman distance, the numbers are still in favour of UC-SP: 85% versus 64%.

<sup>5</sup> The Kendall's tau distance between two preference order  $\succ_i$  and  $\succ_{i'}$  over  $\mathcal{R}$  is the number of pairs  $(r, r') \in \mathcal{R}^2$  such that  $\succ_i$  and  $\succ_{i'}$  do not rank  $r$  and  $r'$  in the same order. The Spearman distance between two preference order  $\succ_i$  and  $\succ_{i'}$  over  $\mathcal{R}$  is defined as  $\sum_{r \in \mathcal{R}} |\text{rank}_{a_i}(r) - \text{rank}_{a_{i'}}(r)|$ . In both cases, the diversity index of a profile  $L$  is the sum of the distance between every pair of preference orders  $(\succ_i, \succ_{i'})$  of  $L$ .

### 8.1.3 Experiments conducted

We conducted two types of experiments:

- (i) A study of the average efficiency and fairness of swap dynamics. The *efficiency* (respectively *fairness*) ratio is defined as the ratio between the average (respectively minimum) rank realized by the procedure and the optimal average (respectively min) rank achievable for the instance (disregarding the individual rationality constraint). The optimal values are obtained using matching techniques [31]. These investigations can be viewed as the empirical counterpart of the worst-case analysis of Sect. 7. To isolate the influence of the individual rationality (IR) constraints alone, we also include the rules “max *ark* IR” and “max *mrk* IR” which respectively returns the allocation maximizing the average and the minimum ranks under the constraint of being individually rational. We also study the performance of swap dynamics with respect to TTC and Crawler.
- (ii) A study of the number of swaps performed by swap dynamics with respect to the number of cycles performed by TTC and Crawler. Of course this comparison needs to acknowledge that the size of cycles may be very different in both approaches. We thus also investigate the size of the cycles computed by TTC and Crawler.

### 8.1.4 Parameters of the experiments

For every experiment, we considered several number of resources, varying from  $n = 2$  to  $n = 60$  and we randomly generated 1000 instances in each case. For a given instance, the initial allocation is selected uniformly at random among all possible allocations. For every instance, we ran the different procedures and report the results for a fixed number of resources, averaged over the 1000 instances.

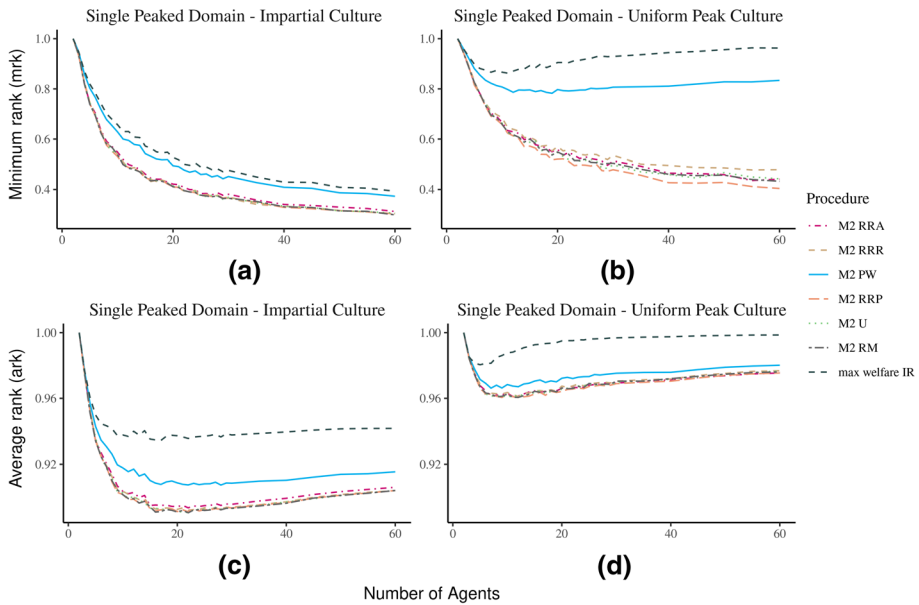
## 8.2 Analysis of the results

### 8.2.1 Efficiency and fairness of swap dynamics

*Comparison among swap dynamics.* Figure 3 presents the efficiency and fairness ratio for each swap dynamic for preferences generated both under the IC-SP model (left side) and under the UP-SP model (right side). Results for max *mrk* IR and max *ark* IR are also presented for comparison purposes.

Regarding the minimum rank—Figs. 3a, b— $M_2$ -PW reaches significantly better allocations than other dynamics. This is expected: this heuristic favours deals between agents holding low ranked resources and thus tends to improve the satisfaction of the poorest agents. On the contrary, round-robin heuristics with their fixed ordering tend to always favour the same agents and thus often leads to lower minimum rank. However, notice that the randomized round-robin version ( $M_2$ -RRR) slightly corrects this.

Interestingly, the performance of  $M_2$ -PW is highly sensitive to the culture considered. This seems to be due to the way preferences are generated. Indeed, the results for max *mrk* IR follow similar patterns, indicating that IC-SP (Fig. 3a) is much more constrained by the initial allocation than UP-SP (Fig. 3b) is. Still, under UP-SP,  $M_2$ -PW returns close to optimal allocations, while other dynamics remain with a fairness ratio of about 50%. Following



**Fig. 3** Average efficiency and fairness ratios for each heuristic of the swap-deal procedure and for each preference culture. For comparison baseline, the average maximum welfare achievable under individual rationality (max *mrk* IR for the upper part and max *ark* IR for the lower part) is also presented. The number of agents varies from 2 to 60

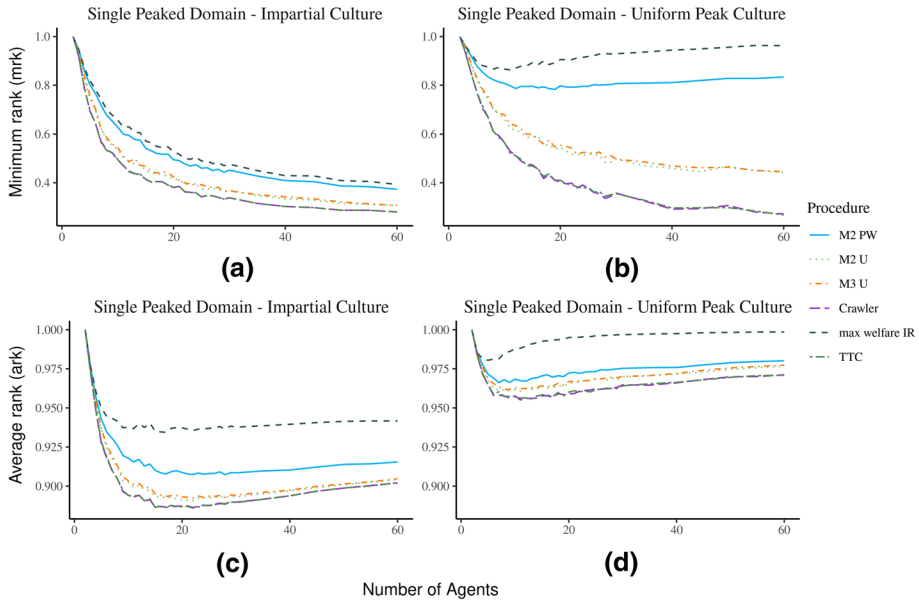
this observation, as the allocation returned by  $M_2$ -PW can be taken as a witness of a good allocation reachable by swap dynamics, we see that the empirical minimum rank price of dynamics for our studied family of dynamics is much higher in UP-SP than in IC-SP.

Regarding the average rank of the outcomes—Figs. 3c, d—all the heuristics obtain very good results [(above 90% under IC-SP (Fig. 3c) and above 96% under UP-SP (Fig. 3d)]. It can be observed that all heuristics give very similar values.

Recall that our price of anarchy results informed us that in principle, swap dynamics can return allocation with an efficiency ratio as low as 50%. Given these high performances, it is not surprising that very little room is left to observe difference among selection heuristics. Still, it is noteworthy that  $M_2$ -PW provides, again, the best results (while this selection heuristics doesn't seem designed to optimize this measure of welfare at first sight). Note that our remark about the relative performance of P2-PW under the two cultures does not hold here: this suggests that only a few agents are left with low-ranked resources with our swap dynamics, even under UP-SP.

*Comparison with TTC and Crawler.* We now compare swap dynamics to other procedures. For the sake of readability, we only keep  $M_2$ -PW (which offered the best performance) and  $M_2$ -U (the most decentralized selection heuristics, requiring hardly any communication between the agents). Results are presented in Fig. 4.

Regarding the minimum rank—Fig. 4a, b—a first observation is that  $M_2$ -U performs significantly better than TTC and Crawler algorithm (which give similar results). The poor performances of these procedures, not designed for this purpose, are not surprising: they typically leave some agents with very little opportunities to be involved in a cycle-deal. For instance, by selecting top-trading cycles, TTC can remove all the agents involved once a deal is implemented, thus significantly limiting the range of possible deals for the



**Fig. 4** Average efficiency and fairness ratios for each procedure and for each preference domain. Curves for “max welfare IR” correspond to max *mrk* IR for the upper part of the figure and to max *ark* IR for the lower part. The number of agents varies from 2 to 60

remaining agents. On the other hand, as discussed,  $M_2$ -PW favours low ranked agents and gives more opportunities to these agents to exchange their initial resources.

Regarding the average rank of the outcomes obtained by the different procedures—Fig. 4c, d—swap dynamics slightly outperform TTC and Crawler under both cultures.

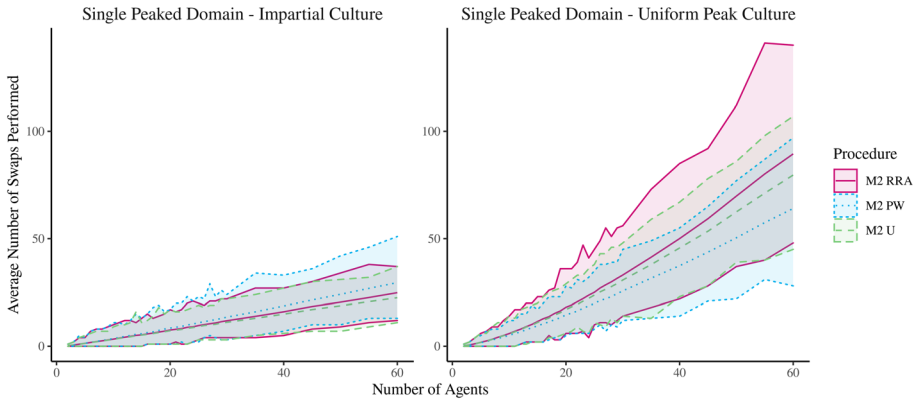
Finally, it can also be observed that  $M_3$ -U shows no significant difference with  $M_2$ -U under the same heuristics (uniform selection of the exchanges), either for the average or minimum rank: slightly increasing the size of the deals leads raises complex coordination issues with no evidence of improvements in terms of welfare.

These experiments promote the relevance of swap dynamics: besides being simple to implement, they also provide very good results both in terms of average and minimum rank, close to the optimal when individual rationality is enforced. For the minimum rank and under UP-SP, we observed that our history-based swap dynamics may incur a significant cost, even though better allocations would be reachable by swaps (this was shown by comparison to a more specifically designed, not history-based, selection heuristic).

### 8.2.2 Length of swap dynamics

Figure 5 represents the mean number of swap-deals performed (solid lines) when varying the size of the instances. For the sake of readability, we only show some selection heuristics (the other ones performed a number of swaps similar to either  $M_2$ -RRA or  $M_2$ -U). Dotted lines represent the highest and the lowest numbers of swaps registered for an instance of a given size (averaged over 1000 randomly generated instances).

It can be observed that the different selection heuristics are close in terms of the average number of swaps. However, the number of swaps performed under UP-SP is significantly



**Fig. 5** Average number of swap-deals performed, the filled area represents the range, from the minimum to the maximum. The number of agents varies from 2 to 60

higher than under IC-SP. This phenomenon is related to the method used to generate single-peaked preferences. According to the discussion of Sect. 8.1.2, UP-SP offers a greater diversity of profiles, which leads to more opportunities of exchanges. However, even though we noted in previous experiments that the results in terms of welfare were better in UP-SP than in IC-SP, note that the proportion is of a different order here: there are on average three times more swap-deals performed under UP-SP.

The higher diversity of profiles under UP-SP can also explain the higher variance in the number of swaps. Typically, when agents have completely opposite preference orders, the number of deals can greatly vary depending on which agents are encountered. This is much more likely to happen under UP-SP.

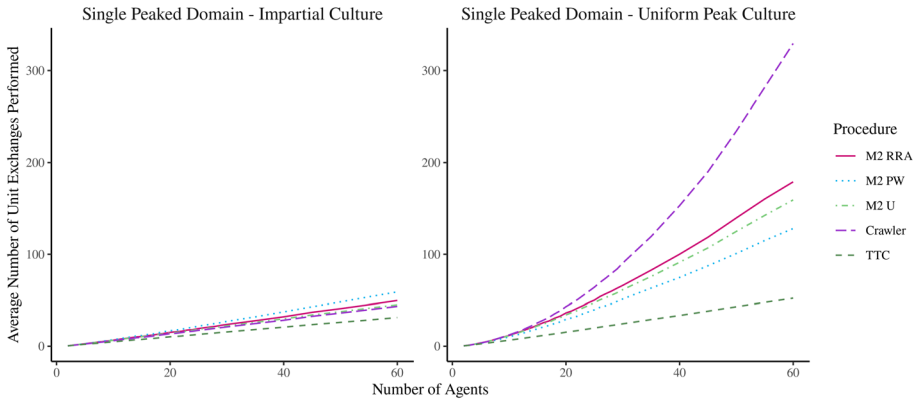
*Comparison with TTC and Crawler.* The number of deals performed by swap dynamics can be compared with the number of deals induced by TTC or Crawler. To do so we introduce the number of unit exchanges performed by a procedure. It corresponds to the sum of the length of the cycle-deals applied. For a swap dynamics, it is twice the number of swap-deals performed for instance. This measure allows us to compare all the procedures even though they perform cycle-deals of different sizes. As shown in Fig. 6, the procedure can be clustered in three distinct classes. The swap dynamics can be grouped together, offering a middle way between TTC which performs particularly well and Crawler which performs particularly badly in terms of number of unit exchanges.

The difference between Crawler and TTC can be explained by the fact that the former performs much more deals on average than TTC. As shown in Fig. 7, this can be observed for any number of agents and the gap increases as the number of agents increases. The size of the deals is also larger when implementing Crawler (Fig. 8). Both of these facts naturally lead to a higher number of unit exchanges.

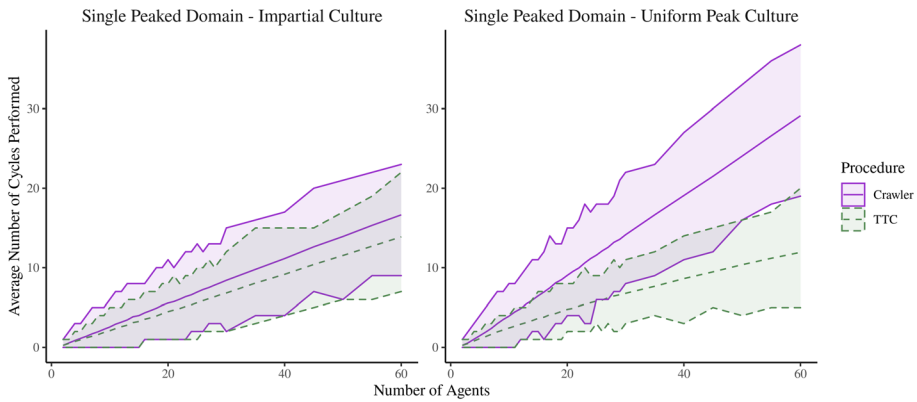
The larger number of deals and the larger sizes of deals performed by Crawler are related to the fact that agents are ordered with respect to the resource they initially hold and with respect to the order of these resources on the single-peaked axis. Based on this order, Crawler only considers deals  $\mu = \langle a_1, \dots, a_k \rangle$  such that  $a_i$  and  $a_{i+1}$  (with  $i \in \{1, 2, \dots, k - 1\}$ ) are owners of adjacent resources on the axis. Hence, TTC allows for considering a larger range of cycle-deals than Crawler.

Putting aside the comparison between Crawler, and TTC, the size of the deals to implement can be extremely large for both procedures as depicted in Fig. 8. A cycle-deal may involve more than half (respectively 35%) of the agents under UP-SP and a

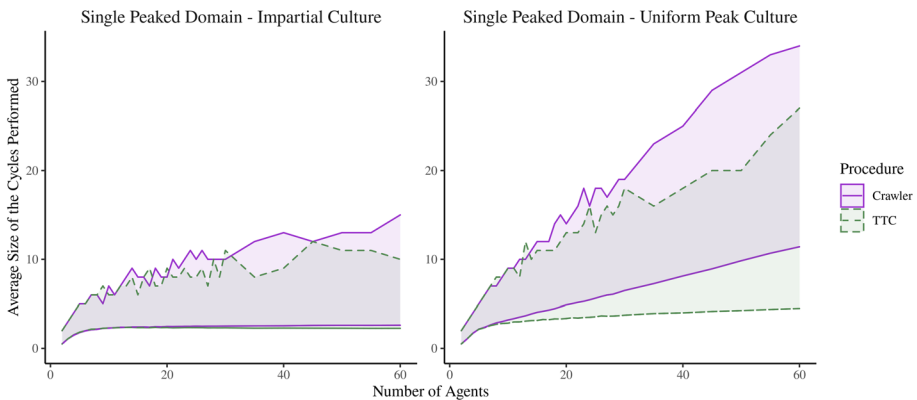




**Fig. 6** Average number of unit exchanges performed by some swarm dynamics, TTC and Crawler. A cycle-deal of length  $k$  counts for  $k$  unit exchanges. The number of agents varies from 2 to 60



**Fig. 7** Average number of cycles performed by TTC and Crawler, the filled area represents the range, from the minimum to the maximum. The number of agents varies from 2 to 60



**Fig. 8** Average size of the cycles performed by TTC and Crawler, the filled area represents the maximum. The minimum, not shown, is constantly equal to zero. The number of agents varies from 2 to 60

**Table 1** Linear regression of the maximum size of the deals over the number of agents.  $\beta_0$  and  $\beta_1$  are the coefficients of the regression:  $maxSize = \beta_0 + \beta_1 * n$  where  $n$  is the number of agents,  $R^2$  the coefficient of determination. The p-value is omitted as it is not meaningful for simulations (see, e.g., [39])

	TTC IC-SP	TTC UP-SP	Crawler IC-SP	Crawler UP-SP
$\beta_0$	4.64	4.73	4.27	2.86
$\beta_1$	0.15	0.36	0.20	0.56
$R^2$	0.7872	0.9032	0.883	0.9862

fifth (respectively 15%) of the agents under IC-SP for Crawler (respectively TTC). The linear regressions explaining these values are presented in Table 1. Overall, even though TTC performs less unit exchanges than our swap dynamics, it can still need to perform extremely large cycle-deals.

These experiments show that the number of swap-deals can be significant, especially under UP-SP where long sequences of slightly improving steps are more likely to occur. On the other hand, we show empirically that TTC and Crawler are indeed prone to require the implementation of cycle-deals of large size. This illustrates the trade-off which occurs between the coordination requirements and the length of the procedure.

## 9 Conclusion

This paper studied the property of swap dynamics for the allocation of indivisible resources in the restricted setting of single-peaked housing markets. The basic principle of the procedure is to let agents perform pairwise improving exchanges. We showed in particular that the single-peaked domain happens to be maximal for guaranteeing convergence to Pareto-optimal outcomes with such dynamics. We also showed that the outcomes of TTC and of Crawler are reachable by swap-deal sequences. To refine our analysis, we have studied two further notions: the average rank and the minimum rank of the resources obtained by the agents. None of the procedures discussed in this paper are specifically designed for optimizing these ranks, even though these notions capture very natural criteria of efficiency (for the average rank) and fairness (for the minimum rank). It thus seems important to study how these allocation procedures behave on that respect. To complement worst-case theoretical bounds on the loss of social welfare induced by swap dynamics, we ran experiments which demonstrated that they actually provide good results in practice.

Our focus on swap-deals is motivated by their minimal coordination requirements. In the context of this paper, our experimental results suggest that there is little gain to expect when allowing deals involving three agents. A complementary relaxation is to permit deals possibly not improving for some of the agents involved (but at least a majority).

This may unveil interesting connections with other notions: [37] recently established for instance that an allocation is popular in a housing market if and only if no (majority) improving exchanges between three agents exist.

To go further with the experiments, it would be interesting to use real data. Our attempt to use data from Preflib [44] was not successful as there is no dataset that is single-peaked when there are more than 5 agents. Getting such preferences would be an interesting way to confirm our results. Regarding the model itself, [8] extended Crawler to single-peaked domains with indifferences. Whether our results with swap dynamics could be similarly generalized is an avenue for future research. Overall, this paper raises the exciting issue of giving a characterization of rules that are efficient, individually rational and strategy-proof for the single-peaked domain. Such a characterization would also provide more insights about the type of selection heuristic for which the swap dynamics are strategy-proof. It would also be interesting to tackle the characterization of the swap dynamics efficiency, that is, identifying the domain on which the swap dynamics are Pareto-optimal. Our maximality result is a significant step in this sense, it would be nice to complete the picture.

**Acknowledgements** We thank Sophie Bade, Yann Chevaleyre, Anastasia Damamme, and Julien Lesca, for discussions related to this topic as well as the anonymous reviewers for their comments and suggestions which significantly improved the paper.

## References

1. Abbasi, Z., Lakshmanan, L. V., & Xie, M. (2013). Fair recommendations for online barter exchange networks. In: Proceedings of the 16th International Workshop on the Web and Databases (WebDB), pp 43–48.
2. Ackermann, H., Goldberg, P. W., Mirrokni, V. S., Röglin, H., & Vöcking, B. (2011). Uncoordinated two-sided matching markets. *SIAM Journal on Computing*, 40(1), 92–106.
3. Anshelevich, E., Das, S., & Naamad, Y. (2013). Anarchy, stability, and utopia: creating better matchings. *Autonomous Agents and Multi-Agent Systems*, 26(1), 120–140.
4. Arrow, K. J. (1951). *Social choice and individual values*. Wiley.
5. Aziz, H. (2020). Developments in multi-agent fair allocation. In: Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI), pp 13563–13568.
6. Aziz, H., Gaspers, S., Mackenzie, S., & Walsh, T. (2015). Fair assignment of indivisible objects under ordinal preferences. *Artificial Intelligence*, 227, 71–92.
7. Aziz, H., Biró, P., Lang, J., Lesca, J., & Monnot J. (2016). Optimal reallocation under additive and ordinal preferences. In *Proceedings of the 15th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pp. 402–410.
8. Bade, S. (2019). Matching with single-peaked preferences. *Journal of Economic Theory*, 180, 81–99.
9. Ballester, M. A., & Haeringer, G. (2011). A characterization of the single-peaked domain. *Social Choice and Welfare*, 36(2), 305–322.
10. Bentert M, Chen J, Froese V, & Woeginger GJ. (2019). Good things come to those who swap objects on paths. CoRR abs/1905.04219.
11. Beynier, A., Bouveret S, Lemaître M, Maudet N, Rey S, & Shams P. (2019). Efficiency, sequenceability and deal-optimality in fair division of indivisible goods. In: *Proceedings of the 18th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pp 900–908.
12. Beynier, A., Maudet, N., Rey, S., & Shams, P. (2020). An optimal procedure to check pareto-optimality in house markets with single-peaked preferences. *ArXiv*, 2002, 11660.
13. Black, D. (1948). On the rationale of group decision-making. *Journal of Political Economy*, 56(1), 23–34.
14. Van den Bos, K., Lind, E. A., Vermunt, R., & Wilke, H. A. (1997). How do I judge my outcome when I do not know the outcome of others? the psychology of the fair process effect. *Journal of Personality and Social Psychology*, 72(5), 1034.
15. Bouveret, S., Endriss, U., & Lang, J. (2010). Fair division under ordinal preferences: Computing envy-free allocations of indivisible goods. In: *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI)*, pp 387–392.

16. Bouveret, S., Chevaleyre, Y., & Maudet, N. (2016). Fair allocation of indivisible goods. In V. Conitzer, U. Endriss, J. Lang, & A. D. E. Procaccia (Eds.), *Brandt F* (pp. 284–310). Cambridge University Press: Handbook of Computational Social Choice.
17. Brams, S. J., Edelman, P. H., & Fishburn, P. C. (2003). Fair division of indivisible items. *Theory and Decision*, 55(2), 147–180.
18. Brandt F, & Wilczynski A. (2019). On the convergence of swap dynamics to pareto-optimal matchings. In: *Proceedings of the 15th International Workshop on Internet and Network Economics (WINE)*, pp. 100–113.
19. Chevaleyre, Y., Dunne, P. E., Endriss, U., Lang, J., Lemaître, M., Maudet, N., et al. (2006). Issues in multiagent resource allocation. *Informatica*, 30(1).
20. Chevaleyre, Y., Endriss, U., Estivie, S., & Maudet N. (2007). Reaching envy-free states in distributed negotiation settings. In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, vol 7, pp 1239–1244.
21. Chevaleyre, Y., Endriss, U., & Maudet, N. (2010). Simple negotiation schemes for agents with simple preferences: Sufficiency, necessity and maximality. *Autonomous Agents and Multi-Agent Systems*, 20(2), 234–259.
22. Chevaleyre, Y., Endriss, U., & Maudet, N. (2017). Distributed fair allocation of indivisible goods. *Artificial Intelligence*, 242, 1–22.
23. Conitzer, V. (2009). Eliciting single-peaked preferences using comparison queries. *Journal of Artificial Intelligence Research*, 35, 161–191.
24. Damamme A, Beynier A, Chevaleyre Y, & Maudet N. (2015). The power of swap deals in distributed resource allocation. In: *Proceedings of the 14th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pp 625–633.
25. DeMarzo, P. M., Vayanos, D., & Zwiebel, J. (2003). Persuasion bias, social influence, and unidimensional opinions. *The Quarterly Journal of Economics*, 118(3), 909–968.
26. Dunne, P., Wooldridge, M., & Laurence, M. (2005). The complexity of contract negotiation. *Artificial Intelligence*, 164(1), 23–46.
27. Egan, P. J. (2014). “do something” politics and double-peaked policy preferences. *The Journal of Politics*, 76(2), 333–349.
28. Elkind E, Lackner M, & Peters D. (2017). Structured preferences. In: E U (ed) Trends in Computational Social Choice, Lulu.com, pp 187–207.
29. Endriss, U., & Maudet, N. (2005). On the communication complexity of multilateral trading. *Autonomous Agents and Multi-Agent Systems*, 11(1), 91–107.
30. Endriss U, Maudet N, Sadri F, & Toni F. (2006). Negotiating socially optimal allocations of resources. *Journal of Artificial Intelligence Research*.
31. Garfinkel, R. S. (1971). An improved algorithm for the bottleneck assignment problem. *Operations Research*, 19(7), 1747–1751.
32. Gourvès L, Lesca J, & Wilczynski A. (2017). Object allocation via swaps along a social network. In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pp 213–219.
33. Hashemi V, & Endriss U. (2014). Measuring diversity of preferences in a group. In: *Proceedings of the 21st European Conference on Artificial Intelligence*, pp 423–428.
34. Hougaard, J. L., Moreno-Terreno, J. D., & Østerdal, L. P. (2014). Assigning agents to a line. *Games and Economic Behavior*, 87, 539–553.
35. Huang S, & Xiao M. (2019). Object reachability via swaps along a line. In: *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, vol 33, pp 2037–2044.
36. Kasajima, Y. (2013). Probabilistic assignment of indivisible goods with single-peaked preferences. *Social Choice and Welfare*, 41(1), 203–215.
37. Kondratev, A.Y., & Nesterov, A. S. (2019). Minimal envy and popular matchings. arXiv preprint [arXiv:1902.08003](https://arxiv.org/abs/1902.08003).
38. Koutsoupias, E., & Papadimitriou, C. (1999). Worst-case equilibria. In: Meinel, C., Tison, S. (eds) *Proceedings of the 16th International Symposium on Theoretical Aspects of Computer Science (STACS)*, pp. 404–413.
39. Lee, J. S., Filatova, T., Ligmann-Zielinska, A., Hassani-Mahmooui, B., Stonedahl, F., & Lorscheid, I., et al. (2015). The complexities of agent-based modeling output analysis. *Journal of Artificial Societies and Social Simulation*, 18(4).
40. Leventhal, G. S. (1980). *What should be done with equity theory Social exchange* (pp. 27–55). Berlin: Springer.

41. List, C., Luskin, R. C., Fishkin, J. S., & McLean, I. (2013). Deliberation, single-peakedness, and the possibility of meaningful democracy: evidence from deliberative polls. *The Journal of Politics*, 75(1), 80–95.
42. Liu P. (2018). A large class of strategy-proof exchange rules with single-peaked preferences. Research Collection School of Economics, mimeo.
43. Ma, J. (1994). Strategy-proofness and the strict core in a market with indivisibilities. *International Journal of Game Theory*, 23(1), 75–83.
44. Mattei N, & Walsh T. (2013). Preflib: A library for preferences <http://www.preflib.org>. In: Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT), Springer, pp 259–270.
45. Moulin, H. (1991). *Axioms of cooperative decision making* (Vol. 15). Cambridge: Cambridge University Press.
46. Moulin, H. (2019). Fair division in the age of internet. *Annual Review of Economics*, 11, 407–441.
47. Nguyen, N. T., Nguyen, T. T., Roos, M., & Rothe, J. (2014). Computational complexity and approximability of social welfare optimization in multiagent resource allocation. *Autonomous Agents and Multi-agent Systems*, 28(2), 256–289.
48. Pazner, E. A., & Schmeidler, D. (1978). Egalitarian equivalent allocations: A new concept of economic equity. *The Quarterly Journal of Economics*, 92(4), 671–687.
49. Peters D, & Lackner M. (2017). Preferences single-peaked on a circle. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI Press, p 649–655.
50. Puppe, C. (2018). The single-peaked domain revisited: A simple global characterization. *Journal of Economic Theory*, 176(C), 55–80.
51. Rawls, J. (1971). *A theory of justice*. Harvard University Press.
52. Rosenschein JS, & Zlotkin G. (1994). Rules of encounter: designing conventions for automated negotiation among computers. MIT press.
53. Roth, A. E. (1982). Incentive compatibility in a market with indivisible goods. *Economics letters*, 9(2), 127–132.
54. Roth, A. E., & Vate, J. H. V. (1990). Random paths to stability in two-sided matching. *Econometrica*, 58, 1475–1480.
55. Roth, A. E., Sönmez, T., & Ünver, M. U. (2005). Pairwise kidney exchange. *Journal of Economic theory*, 125(2), 151–188.
56. Saffidine A, & Wilczynski A. (2018). Constrained swap dynamics over a social network in distributed resource reallocation. In: *Proceedings of the 11th International Symposium on Algorithmic Game Theory (SAGT)*, pp 213–225.
57. Sandholm T. (1998). Contract types for satisficing task allocation. In: *Proceedings of the Spring Symposium of AAAI Conference on Artificial Intelligence (AAAI)*, pp 23–25.
58. Sen, A. K. (1966). A possibility theorem on majority decisions. *Econometrica*, 34, 491–499.
59. Shapley, L., & Scarf, H. (1974). On cores and indivisibility. *Journal of Mathematical Economics*, 1(1), 23–37.
60. Spector, D. (2000). Rational debate and one-dimensional conflict. *The Quarterly Journal of Economics*, 115(1), 181–200.
61. Sprumont, Y. (1991). The division problem with single-peaked preferences: a characterization of the uniform allocation rule. *Econometrica*, 59, 509–519.
62. Sprumont, Y. (1996). Axiomatizing ordinal welfare egalitarianism when preferences may vary. *Journal of Economic Theory*, 68(1), 77–110.
63. Thibaut JW, & Walker L. (1975). Procedural justice: A psychological analysis. L. Erlbaum Associates.
64. Thomson, W. (1983). Problems of fair division and the egalitarian solution. *Journal of Economic Theory*, 31(2), 211–226.
65. Thomson, W. (2016). Introduction to the theory of fair allocation. In V. Conitzer, U. Endriss, J. Lang, & A. D. Procaccia (Eds.), *Brandt F* (pp. 261–283). Cambridge University Press: Handbook of Computational Social Choice.
66. Walsh, T. (2015). Generating single peaked votes. [arXiv:1503.02766](https://arxiv.org/abs/1503.02766).
67. Walsh, T. (2020). Fair division: The computer scientist’s perspective. In: Bessiere C (ed) *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pp 4966–4972.