Available online at www.sciencedirect.com
SCIENCE DIRECT
Journal of Economic Theory 127 (2006) 155–171
JOURNAL OF Economic Theory
www.elsevier.com/locate/jet

On two competing mechanisms for priority-based allocation problems

Onur Kesten*
Department of Economics, University of Rochester, NY 14627, USA

Received 6 May 2004; final version received 15 November 2004
Available online 11 February 2005

Abstract

We consider the *priority-based allocation problem*: there is a set of indivisible objects with multiple supplies (e.g., schools with seats) and a set of agents (e.g., students) with priorities over objects (e.g., proximity of residence area). We study two well-known and competing mechanisms. The *agent-optimal stable mechanism* (AOSM) allots objects via the *deferred acceptance algorithm*. The *top trading cycles mechanism* (TTCM) allots objects via *Gale's top trading cycles algorithm*. We show that the two mechanisms are equivalent, or TTCM is *fair* (i.e., respects agents' priorities), or *resource monotonic*, or *population monotonic*, if and only if the priority structure is *acyclic*. Furthermore, if AOSM fails to be *efficient* (*consistent*) for a problem, TTCM also fails to be *fair* (*consistent*) for it. However, the converse is not necessarily true.

JEL classification: C78; C79; D61; D78; I20

Keywords: Indivisible goods; Agent-optimal stable mechanism; Top trading cycles mechanism; Acyclicity

1. Introduction

Resource allocation based on priorities is a commonly observed real-life problem. Among many situations one can think of, an important example is the placement of students to public schools in the US. This application is also known as the *school choice problem*. In a school choice problem, there is a set of students each of whom needs to be placed to a school from a set of schools. Each school has a *supply* of seats. Each student has preferences over schools

* Fax: +1 585 256 2309.
E-mail address: kstn@troi.cc.rochester.edu.

0022-0531/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jet.2004.11.001

and each school has a *priority order* of students. These priorities are usually imposed by law and a student's priority for a school is determined according to criteria such as whether he/she is handicapped, the proximity of his/her residence to the school, whether he/she has a sibling attending the same school *etc*. The collection of priority orders is called a *priority structure*.

The school choice problem is closely related to the well-known *college admissions problem* (Gale and Shapley [12]). There is one key difference however: In a college admissions problem, the priority order of each school is replaced by the preferences of that school over students. Unlike the case with preferences, when there are priorities, one does not consider welfare issues or strategic behavior for the school side of the problem. In our study, a priority structure is given as a primitive of the model. Adopting a general resource allocation terminology, we refer to a problem as a *priority-based allocation problem*, to students as *agents*, and to schools as *objects*.

At an *allocation*, no agent is allotted more than one object and the number of agents a particular object is allotted to does not exceed the supply of that object. An allocation is *fair* if no agent envies any other agent whose allotment he has higher priority for. A *mechanism* is a function that associates an allocation to each problem. A mechanism is *fair* (*efficient*) if it always selects *fair* (*efficient*) allocations.

Two important and competing mechanisms have been proposed in the literature: First is the *agent-optimal stable mechanism* (AOSM) (Gale and Shapley [12]). Second is the *top trading cycles mechanism* (TTCM) (Abdulkadiroğlu and Sönmez [3]).

The outcome of AOSM is calculated via the well-known *deferred acceptance algorithm*. AOSM is *fair*. Furthermore, the allocation selected by AOSM Pareto dominates any other *fair* allocation (Balinski and Sönmez [4]). While AOSM enjoys nice properties such as *strategy-proofness*,[1] *fairness*, *resource monotonicity*,[2] and *population monotonicity*,[3] it violates other appealing properties such as *efficiency*, *consistency*,[4] and *group strategy-proofness*.[5]

Gale's top trading cycles procedure has attracted much attention in the recent literature. Due to its various desirable features, a number of mechanisms based on this procedure have been proposed and characterized. (See for example, Pápai [16], Abdulkadiroğlu and Sönmez [1–3], Ehlers et al. [7], Ehlers and Klaus [8], Ehlers [9], and Kesten [13–15]). TTCM is one such mechanism and an important competitor of AOSM. It, too, enjoys nice properties such as *efficiency* and *group strategy-proofness*.[6] However, it lacks *fairness*, *consistency*, *resource monotonicity*, and *population monotonicity*.

[1] No agent ever benefits by misrepresenting his preferences.

[2] All agents are affected in the same direction (in welfare terms) whenever the set of available objects shrinks or expands.

[3] All agents are affected in the same direction (in welfare terms) whenever some agents leave without their allotments.

[4] The recommendation for any given problem does not change after the departure of some of the agents with their allotments.

[5] No group of agents ever benefit by jointly misrepresenting their preferences.

[6] Ehlers [9] considers efficient and group strategy-proof rules for house allocation problems and establishes a maximal domain result.

In a recent paper, Ergin [11] introduces a notion of "acyclicity" for priority structures. Loosely speaking, a priority structure is acyclic if it never gives rise to situations in which an agent can block a potential settlement between any other two agents under AOSM. The acyclicity restriction enables AOSM to recover many desirable properties which are lacking in the absence of this restriction. One can say more: AOSM is *efficient*, or *group strategy-proof*, or *consistent* if and only if the priority structure is *acyclic* (Ergin [11]).

In this paper, we introduce stronger notions of acyclicity on priority structures. We show that *AOSM and TTCM are equivalent*, *or TTCM is fair* (*or*, *stable*), *or resource monotonic*, *or population monotonic*, *if and only if the priority structure satisfies our first notion of acyclicity* (Theorem 1). We also show that *TTCM is consistent if and only if the priority structure satisfies our second and an even stronger form of acyclicity* (Theorem 2). In addition, *if AOSM selects an inefficient allocation for a problem*, *then TTCM selects an unfair allocation for it*. *Yet*, *the converse is not necessarily true* (Proposition 2). *If AOSM is not consistent for a problem*, *then TTCM is not consistent for it either*. *Yet, the converse is again not necessarily true* (Proposition 3).

The paper is organized as follows: In Section 2 we introduce the model and describe the two mechanisms. We introduce further properties of mechanisms in Section 3. In Section 4 we present the main results. We give a brief conclusion in Section 5. All the proofs are deferred to the Appendix.

2. Two competing mechanisms

2.1. The model

Let $N \equiv \{1, 2, \dots, n\}$ denote the finite set of agents. Let X denote the finite set of objects. We use the letters of the alphabet such as x and y to represent typical elements of X. If an agent is not allotted any object in X, we say that he is allotted the *null object*. Let $\varnothing$ denote the null object. Let $s_x \geqslant 1$ be the number of units available of object x or, the *supply* of x, and $s \equiv (s_x)_{x \in X}$ the supply vector. Each agent $i \in N$ is equipped with a complete, transitive, and strict preference relation R_i over $X \cup \{\varnothing\}$. Let $\mathcal{R}$ denote the class of all such preferences. Let P_i denote the strict relation associated with R_i.

Also given are "priorities" of agents over objects.[7] Let $\mathcal{S}^N$ denote the set of all bijections from N to N. Given $x \in X$, the *priority order* $f_x \in \mathcal{S}^N$ assigns ranks to agents according to their priority for object x. The *rank* of agent i with respect to f_x is $f_x(i)$. Then, $f_x(i) < f_x(j)$ means that *agent i has higher priority* (*or*, *lower rank*) *for object x than agent j*. The collection of priority orders is the *priority structure*. We denote it by $f \equiv (f_x)_{x \in X}$. We assume that a priority structure is given as a primitive of the model.

A (priority-based allocation) *problem* is described by a preference profile $R \equiv (R_i)_{i \in N} \in \mathcal{R}^N$. An *allocation* $\alpha \equiv (\alpha_i)_{i \in N}$ is a list of *allotments* such that no agent is allotted more than one object and the number of agents a particular object is allotted to does not exceed

[7] For example, consider a real-life application of the model where agents are *students* and objects are *schools*. Then, these priorities could be enforced by local or state laws to ensure equity among students with different backgrounds, life standards, etc.

the supply of that object. The null object can be allotted to any number of agents. Let $\mathcal{A}$ denote the set of all allocations. A *mechanism* ϕ is a function that associates with each problem $R \in \mathcal{R}^N$ an allocation $\phi(R) \in \mathcal{A}$. Given $R \in \mathcal{R}^N$, let $\phi_i(R)$ denote agent i's allotment at $\phi(R)$.

Given $M \subseteq N$ and $R \in \mathcal{R}^N$, let R_M denote the profile $(R_i)_{i \in M}$. Let $\phi(R)|_M$ denote $(\phi_i(R))_{i \in M}$. Also, let s_{-x} denote $(s_y)_{y \in X \setminus \{x\}}$.

2.2. Two central properties

A very closely related model to ours is the two-sided matching model of college admission (Gale and Shapley [12]). In that model, each student (agent) has preferences over schools (objects) and each school has preferences over students. To relate the two models, here preferences of schools are replaced by priority orders for these schools. As opposed to preferences, priority orders are not determined according to welfare criteria, and therefore in our setting strategic and welfare issues are relevant only for agents (and not objects). Also, here we allow only preferences to change. Hence, we assume throughout that f is a priority structure that does not vary with the problem. We next introduce two central properties of mechanisms.

At an *efficient allocation* it is not possible to make an agent better off without making another agent worse off. An *efficient mechanism* always selects efficient allocations. Let ϕ be a mechanism.

Efficiency: There do not exist $R \in \mathcal{R}^N$ and $\alpha \in \mathcal{A}$ such that for all $i \in N$, $\alpha_i \; R_i \; \phi_i(R)$, and for some $j \in N$, $\alpha_j P_j \phi_j(R)$.

In priority-based allocation problems, it is critical that a mechanism *respects* agents' priorities for objects. For this purpose, Balinski and Sönmez [4] introduce a "fairness" criterion. We say agent i *envies* agent j at an allocation α if $\alpha_j P_i \alpha_i$. At a *fair allocation* no agent envies any other agent whose allotment he has higher priority for. A *fair mechanism* always selects fair allocations.

Fairness: [8] For all $i, j \in N$ and all $R \in R^N$, $\phi_j(R) \; P_i \; \phi_i(R) \Rightarrow f_{\phi_j(R)}(j) < f_{\phi_j(R)}(i)$.

Unfortunately, no mechanism is both fair and efficient (Balinski and Sönmez [4]). [9] So far, two competing mechanisms have received most of the attention in priority-based allocation problems. First is the well-known *agent-optimal stable mechanism* (AOSM) (Gale and Shapley [12]). Second is the *top trading cycles mechanism* (TTCM) (Abdulkadiroğlu and Sönmez [3]).

---

[8] This notion is also referred as *adoptedness* to or *respecting* a priority structure.

[9] Here we consider a deterministic indivisible good allocation model in which monetary compensations are not allowed. For such models, equity notions are often hard to achieve along with other desirable properties. One way to get around this problem is to introduce randomization into the model. See Bogomolnaia and Moulin [5] and Abdulkadiroğlu and Sönmez [1] for two such examples and further references.

2.3. The agent-optimal stable mechanism

The central concept for college admissions problems is "stability."[10] At a stable allocation (using our terminology): (1) each agent prefers his own allotment to being allotted the null object; (2a) there is no agent-object pair such that both prefer being allotted to each other to their current allotment, (2b) there is no unallotted object that an agent prefers to the object he is allotted.

It is easy to see that fairness is equivalent to condition (2a) and thus, is weaker than stability. For a given problem, the following *deferred acceptance algorithm* (Gale and Shapley [12]) is used to find the stable allocation that is most preferred by each agent:

Algorithm I (*The deferred acceptance algorithm*).

Step 1: *Each agent "applies to" his favorite object. If the null object is the favorite object of an agent, then he is allotted the null object. If the number of agents who apply to an object say, x, is greater than the supply s_x of x, then those s_x agents among them with the highest priority for x temporarily hold x. The remaining agents are rejected.*

Step k, $k \geqslant 2$: Each agent who is "rejected by" an object at step $k - 1$ applies to his next favorite object. If the null object is the next favorite object of an agent, then he is allotted the null object. If the total number of agents who apply to an object say, x, and who are temporarily holding x, is greater than the supply s_x of x, then those s_x agents among them with the highest priority for x temporarily hold x. The remaining agents are rejected.

Stop when each agent is either holding an object or has been allotted the null object.

At termination, each agent who is temporarily holding an object is allotted that object. We call the mechanism that associates the outcome calculated via the deferred acceptance algorithm with each problem, the *agent-optimal stable mechanism*. We denote it by A^f, where f is the priority structure.

It is well-known that A^f is fair. Furthermore, *the allocation chosen by A^f Pareto dominates any other fair allocation* (Balinski and Sönmez [4]). Also, A^f is strategy-proof (Dubins and Freedman [6]). However, it violates efficiency and group strategy-proofness.

Ergin [11] introduces a notion of "acyclicity" for priority structures. For A^f, this restriction prevents situations in which an agent may block a potential settlement between any other two, thereby causing inefficiency. Given an agent i and an object x, let $U_x^f(i) \equiv \{j \in N | f_x(j) < f_x(i)\}$ be the set of agents who have higher priority than agent i for object x. A *strong-cycle* of f constitutes if the following two conditions are met:

Loop condition: There are $i, j, k \in N$ and $x, y \in X$ such that $f_x(i) < f_x(j) < f_x(k)$ and $f_y(k) < f_y(i)$.

Scarcity condition: There are (possibly empty) disjoint sets $N_x, N_y \subset N \backslash \{i, j, k\}$ such that $N_x \subset U_x^f(j)$, $N_y \subset U_y^f(i)$, $|N_x| = s_x - 1$, and $|N_y| = s_y - 1$.

The priority structure f is *weakly acyclic* if it has no strong-cycles.

[10] See Roth and Sotomayor [18] for a comprehensive account on college admissions problems and other two-sided matching applications.

The agent-optimal stable mechanism A^f *is efficient*, *or group strategy-proof*, *if and only if f is weakly-acyclic* (Ergin [11]). [11]

2.4. The top trading cycles mechanism and an equivalence

The top trading cycles mechanism (Abdulkadiroğlu and Sönmez [3]) is based on *Gale's top trading cycles algorithm*. This algorithm yields an allocation which is in the core [12] of a "house exchange market" [13] (Shapley and Scarf [19]). The algorithm works as follows: initially, each object is *assigned* to a different agent. Each object "points to" the agent it is assigned to and each agent points to his favorite object. If the null object is the favorite object of an agent, then he points to himself and constitutes a *self-cycle*. Since the number of agents and the number of objects are finite, there is at least one cycle. Then in each cycle, the corresponding trades are performed, i.e., each agent in the cycle is *allotted* the object he points to and these agents and objects are removed. Then the same procedure is applied to the reduced market and so on. The algorithm terminates when there are no agents or objects left.

The top trading cycles mechanism is an adaptation of Gale's top trading cycles algorithm to the priority based allocation context. Since now there may be multiple units of a particular object, a counter is assigned to each object to denote the number of units available of that object at each step. Again, in each cycle trades are performed, i.e., the corresponding agents are allotted their favorite objects. Then these objects and agents are removed. Upon the removal of objects, the counters for these objects are reduced. The outcome of the top trading cycles mechanism is calculated via the following algorithm for a given problem:

Algorithm II (*The top trading cycles algorithm*).

Step 1: *Assign a counter to each object. Its initial value is the supply of that object. Each agent points to his favorite object and each object points to the agent who has the highest priority for that object. If the null object is the favorite object of an agent, then he forms a self-cycle. There is at least one cycle. Each agent in a cycle is allotted the object he points to and is removed. The counter of each object in a cycle is reduced by one and if it becomes zero, the object is also removed. The counters of all other objects remain the same.*

Step k, $k \geqslant 2$: *Each remaining agent points to his favorite object among the remaining objects and each remaining object points to the agent who has the highest priority for that object among the remaining agents. If the null object is the favorite object of an agent, then he forms a self-cycle. There is at least one cycle. Each agent in a cycle is allotted the object he points to and is removed. The counter of each object in a cycle is reduced by one and if it becomes zero, the object is also removed. The counters of all other objects remain the same.*

Stop when no agent or object is left.

[11] For the two-sided matching context, Kesten [15] shows that this mechanism is immune to two kinds of manipulation of the "object side" if and only if the priority structure satisfies a similar form of acyclicity.

[12] If preferences are strict, this allocation is unique (Roth and Postlewaite [17]).

[13] In the literature, house exchange markets are usually referred as *housing markets*.

We call the mechanism that associates the outcome calculated via the above algorithm with each problem the *top trading cycles mechanism*. We denote it by T^f, where f is the priority structure. By making priorities transferable, T^f achieves efficiency and group strategy-proofness, two important properties A^f lacks. However, these come with an important price: fairness.

It is possible to further simplify Algorithm II. Instead of having an object point to the agent with the highest priority for that object and keeping track of a counter, we can simply assign *all* available units of an object to the agent with the highest priority for that object and let *only* agents point to each other (instead of having both objects and agents point to each other). This can be achieved by the following simple adaptation of Pápai's [16] "fixed endowment inheritance rules" considered in the "house allocation" context: assign all available units of each object to the agent with the highest priority for that object. An agent may be assigned more than one object. Let each agent point to the agent (possibly himself) who is assigned his favorite object. Then apply the top trading cycles procedure. To each agent in a cycle, allot one unit of his favorite object the agent he points to is assigned. Since an agent can initially be assigned more than one object, after the trades are carried out, some of these objects may remain unallotted. The unallotted objects of each agent who is part of a cycle are *inherited* according to the priority structure by agents who have not yet been allotted any objects. The top trading cycles procedure is again applied to the new problem and so on. This is summarized in the following algorithm for a given problem: [14]

Algorithm III (*The top trading cycles algorithm with inheritance*).

Step 1: *Each agent who has the highest priority for an object is assigned all units available of that object. Each agent points to the agent* (*possibly himself*) *who is assigned* (*all units of*) *his favorite object.* [*If the null object is the favorite object of an agent, then he forms a self-cycle.*] *There is at least one cycle. Each agent in a cycle is allotted one unit of his favorite object and is removed.* [*Some units of the objects that were assigned to an agent who is part of a cycle may remain unallotted.*]

Step k, $k \geqslant 2$: *All the unallotted units of each object that was assigned to an agent who is part of a cycle at step* $k-1$ *are inherited by the agent who has the highest priority for that object among the remaining agents.* [*Hence, each such agent is now assigned all the remaining units of that object* (*in addition to his assignments from previous steps*).] *Each agent points to the agent* (*possibly himself*) *who is assigned his favorite object.* [*If the null object is the favorite object of an agent, then he forms a self-cycle.*] *There is at least one cycle. Each agent in a cycle is allotted one unit of his favorite object and is removed.* [*Some units of the objects that were assigned to an agent who is part of a cycle may remain unallotted.*]

Stop when no agent is left.

We call the mechanism that associates the outcome calculated via the above algorithm with each problem the *inheritance mechanism*. We denote it by I^f, where f is the priority structure. The following example illustrates how a typical inheritance mechanism works.

[14] See Pápai [16] for a formal definition of inheritance rules. Also see Ehlers et al. [7] for a formal definition of fixed endowment inheritance rules.

Example 1. Let $N \equiv \{1, 2, 3, 4, 5, 6, 7\}$, $X \equiv \{a, b, c, d, e\}$, $s_a = s_d = 2$, and $s_b = s_c = s_e = 1$. Let f and R be as given below. (The supply of each object is indicated above the priority order for that object.):

(2) f_a	(1) f_b	(1) f_c	(2) f_d	(1) f_e
6	1	7	3	3
4	2	2	7	5
3	3	6	5	7
7	7	3	6	1
5	4	1	2	6
1	6	5	1	2
2	5	4	4	4

R_1	R_2	R_3	R_4	R_5	R_6	R_7
a	b	b	e	c	e	a
$\emptyset$	a	e	c	b	b	b
c	$\emptyset$	a	a	e	$\emptyset$	$\emptyset$
b	d	d	$\emptyset$	d	d	d
d	e	c	b	$\emptyset$	c	e
e	c	$\emptyset$	d	a	a	c

Step 1: The first row of the priority structure f describes the initial assignments. For ease of notation, if the number of units of an object type inherited by an agent is greater than one, then we write that number in parenthesis next to that object. Let $A_t(i)$ denote the set of objects agent i is assigned at step t. Hence, $A_1(1) = \{b\}$, $A_1(2) = \emptyset$, $A_1(3) = \{d(2), e\}$, $A_1(4) = \emptyset$, $A_1(5) = \emptyset$, $A_1(6) = \{a(2)\}$, and $A_1(7) = \{c\}$ are the initial assignments. Let $f_t(i)$ denote the agent who is assigned all the available units of agent i's favorite object at step t. (We have $f_t(i) = i$ if agent i's favorite object is the null object.) Given R and f, these are: $f_1(1) = 6$, $f_1(2) = 1$, $f_1(3) = 1$, $f_1(4) = 3$, $f_1(5) = 7$, $f_1(6) = 3$, and $f_1(7) = 6$. Thus, agent 1 points to 6, agent 2 points to 1, and so on. Note that the only cycle is formed by agents 1, 3, and 6. Agents 1, 3, and 6 are allotted objects a (one of the two units available), b, and e, respectively, and are removed. Thus, $I_1^f(R) = a$, $I_3^f(R) = b$, and $I_6^f(R) = e$.

Step 2: Objects a and d (2) are inherited by the remaining agents 4 and 7, respectively, according to f, i.e., $A_2(2) = \emptyset$, $A_2(4) = \{a\}$, $A_2(5) = \emptyset$, and $A_2(7) = \{c, d\ (2)\}$. Also, $f_2(2) = 4$, $f_2(4) = 7$, $f_2(5) = 7$, and $f_2(7) = 4$. Hence, agents 4 and 7 point to each other and form a cycle. Thus, $I_4^f(R) = c$ and $I_7^f(R) = a$.

Step 3: The only agents and objects left are agents 2, 5, and two units of object d. According to f, $A_3(2) = \emptyset$ and $A_3(5) = \{d(2)\}$. Since $d\ P_5\ \emptyset$ and there is no other object left, agent 5 points to himself and forms a self-cycle, i.e., $f_3(5) = 5$. Also, since $\emptyset\ P_2\ d$, agent 2, too, forms a self-cycle, i.e., $f_3(7) = 7$. Hence, $I_5^f(R) = d$ and $I_2^f(R) = \emptyset$. One unit of object d remains unallotted.

Two important observations about the inheritance mechanism are in order. First, no agent who is allotted an object at some step envies any other agent who is allotted an object at the same step or at a later step. (Since this agent is pointing to his favorite object at that step.) Second, the allotment of an agent is no worse for him than an object he is assigned at some step. (Since he always has the option of forming a self-cycle.) We call this property the *individual rationality* of I^f. In fact, both being based on variants of Gale's top trading cycles procedure, the inheritance mechanism is equivalent to the top trading cycles mechanism. We omit the straightforward proof.

Proposition 1. *Given a priority structure f, we have $T^f = I^f$.*

3. More properties

3.1. Resource monotonicity

Our first property is a solidarity requirement. We now allow for changes in resources. Imagine a decrease in the supply of an object. We require that all agents are affected in the same direction (in welfare terms) from such a change. Let N be the set of agents. Let X be the *initial* set of objects and s the *initial* supply vector. A *problem* is now a triplet (s'_x, s_{-x}, R), $s'_x \leqslant s_x$. A *mechanism* is a function ϕ that associates with each problem (s'_x, s_{-x}, R) an allocation $\phi(s'_x, s_{-x}, R)$.

Resource monotonicity:[15] For all $x \in X$, all $s'_x \leqslant s_x$, and all $R \in \mathcal{R}^N$, either for all $i \in N$, $\phi_i(s, R)\ R_i\ \phi_i(s'_x, s_{-x}, R)$, or for all $i \in N$, $\phi_i(s'_x, s_{-x}, R)\ R_i\ \phi_i(s, R)$.

We denote the extension of T^f to this context by $\widehat{T}^f$. Formally, given $s' = (s'_x, s_{-x})$, $s'_x \leqslant s_x$, let $\widehat{X} \equiv \{x \in X : s'_x > 0\}$. Let $\widehat{f} \equiv f|_{\widehat{X}} = (f_x)_{x \in \widehat{X}}$ be the induced priority structure for $\widehat{X}$. Also, given $R \in \mathcal{R}^N$, let $R|_{\widehat{X}}$ be the induced preference profile for $\widehat{X}$. Then $\widehat{T}^f(s', R) \equiv T^{\widehat{f}}(R|_{\widehat{X}})$. We also denote the analogous extension of A^f to this context by $\widehat{A}^f$. It is easy to check that while $\widehat{T}^f$ violates resource monotonicity (for instance, if $s'_a = 0$ in Example 1), $\widehat{A}^f$ does satisfy it.

3.2. Population monotonicity

The next property is another solidarity requirement. Now we imagine that some of the agents leave (without their allotments). We require that all agents are affected in the same direction (in welfare terms) from such a change. We now fix the set of objects X and the supply vector s. Let N be the *initial* population of agents. A *problem* is now a pair $(N', (R_i)_{i \in N'})$, $N' \subseteq N$. A *mechanism* is a function ϕ that associates with each problem $(N', (R_i)_{i \in N'})$ an allocation $\phi(N', (R_i)_{i \in N'})$.

Population monotonicity:[16] For all $N' \subseteq N$ and all $R \in \mathcal{R}^N$, either for all $i \in N'$, $\phi_i(N, R)\ R_i\ \phi_i(N', R_{N'})$, or for all $i \in N'$, $\phi_i(N', R_{N'})\ R_i\ \phi_i(N, R)$.

We denote the extension of T^f to this context by $\widetilde{T}^f$. Formally, given $N' \subseteq N$ let $\widetilde{f} \equiv f|_{N'} = (f_x|_{N'})_{x \in X}$ be the induced priority structure for the smaller population. Then given $R \in \mathcal{R}^N$, $\widetilde{T}^f(R_{N'}, N') \equiv T^{\widetilde{f}}(R_{N'})$. We also denote the analogous extension of A^f to this context by $\widetilde{A}^f$. As it is the case for resource monotonicity, while $\widetilde{T}^f$ violates population monotonicity (for instance, if $N' = N \backslash \{3\}$ in Example 1), $\widetilde{A}^f$ does satisfy it.

15 See Thomson [22] for resource monotonic rules on various other economic domains. Also see Ehlers and Klaus [8] and Kesten [13] for related results.

16 See Thomson [21] for various applications of population monotonicity. Also see Ehlers et al. [7] for another related result on population monotonicity

4. The main results

We first introduce a new notion of acyclicity for priority structures. A *cycle* of *f* constitutes if the following two conditions are satisfied:

Loop condition: There are $i, j, k \in N$ and $x, y \in X$ such that $f_x(i) < f_x(j) < f_x(k)$ and $f_y(k) < f_y(i),\ f_y(j)$.

Scarcity condition: There is a (possibly empty) set $N_x \subset N\backslash\{i, j, k\}$ such that $N_x \subset U_x^f(i) \cup (U_x^f(j)\backslash U_y^f(k))$ and $|N_x| = s_x - 1$.

The priority structure *f* is *acyclic* if it has no cycles.

Acyclicity restrictions (the loop condition and the scarcity condition) apply jointly on both the priority structure and the supply vector. Roughly speaking, for T^f, an acyclic priority structure prevents situations in which three agents compete for two objects for which there is excess demand. For example, if the supply of each object is $|N|$, then the resources are abundant enough (hence the scarcity condition is not satisfied) and there are no cycles. At the other extreme, if priority orders are the same for each object, then the loop condition is not satisfied and the structure is acyclic regardless of supplies. As resources become more scarce, acyclicity becomes more restrictive. The next lemma states that our acyclicity condition is stronger than that of Ergin's [11].

Lemma 1. *If a priority structure contains a strong-cycle, then it also contains a cycle.*

All the proofs are given in the Appendix. We are ready to state our main result.

Theorem 1.[17] *Given an initial set of agents, an initial set of objects, an initial supply vector, and a priority structure f, the following statements are equivalent:*

(i) T^f *is fair (or, stable).*
(ii) $T^f = A^f$.
(iii) $\widehat{T}^f$ *is resource monotonic.*
(iv) $\widetilde{T}^f$ *is population monotonic.*
(v) *f is acyclic.*

Theorem 1 clearly shows the importance of the acyclicity restriction on a priority structure for T^f. Acyclicity not only enables this mechanism to recover two important properties of fairness and monotonicity, it also serves as a sufficient and necessary condition for the two mechanisms to always choose the same allocation.

Fairness and efficiency are probably the most desirable two properties that one would hope a good mechanism to have. However, they are incompatible (Balinski and Sönmez [4]). If one considers fairness and efficiency equally important putting all other issues aside, then as the next result suggests, A^f has the edge over its competitor.

Proposition 2. *Given a priority structure f, if A^f selects an inefficient allocation for a problem, then T^f selects an unfair allocation for it. However, the converse is not necessarily true.*

[17] I am indebted to an insightful anonymous referee for pointing out an error in an early version of this theorem.

Example 2. Let $N \equiv \{1, 2, 3\}$, $X \equiv \{x, y\}$, $s_x = 1$, and $s_y = 2$. Let f be such that $f_x(1) < f_x(2) < f_x(3)$ and $f_y(3) < f_y(1) < f_y(2)$. Let for all $i \in \{2, 3\}$, $x P_i y P_i \emptyset$ and $y\ P_1\ \emptyset\ P_1\ x$. The priority structure f is weakly acyclic but not acyclic. Note that $T^f(R) = (y, y, x)$, whereas $A^f(R) = (y, x, y)$. Also $T^f(R)$ is *unfair* [18] (and *efficient*), but $A^f(R)$ is *efficient* (and *fair*).

4.1. Consistency

Our last property is a stability requirement. After an allocation is determined for a given initial problem, suppose some agents leave with their allotments. Once the new problem, or the so-called "reduced problem," is re-evaluated and an allocation is determined for it, we require that the allotments of the remaining agents are the same as those they receive at the initial problem.

We now allow for changes in both resources and population. Let N be the *initial* set of agents, X the *initial* set of objects, and s the *initial* supply vector. A *problem* is now a triplet $(N', s', (R_i)_{i\in N'})$ where $N' \subseteq N$ and $s' \leqslant s$. A *mechanism* is a function ϕ that associates with each problem $(N', s', (R_i)_{i\in N'})$ an allocation $\phi(N', s', (R_i)_{i\in N'})$. Given a mechanism ϕ, an *initial* problem (N, s, R), and a subpopulation $N' \subsetneq N$, let $r^{\phi}_{N'}(N, s, R)$ denote the *reduced problem* with respect to the mechanism ϕ that is faced after agents in $N \backslash N'$ leave with their allotments. Formally, given $N' \subseteq N$ and $R \in \mathcal{R}^N$, $r^{\phi}_{N'}(N, s, R)$ is the problem $(N', s', R_{N'})$ where for all $x \in X$, $s'_x \equiv s_x - |\{j \in N\backslash N' : T_j(R, N, s) = x\}|$.

Consistency: [19] For all $\emptyset \neq N' \subsetneq N$ and all $R \in \mathcal{R}^N$, we have $\phi(N, s, R)|_{N'} = \phi(r^{\phi}_{N'}(N, s, R))$.

We denote the extension of T^f to this context by $\overline{T}^f$. Formally, given $s' \leqslant s$, let $\overline{X} \equiv \{x \in X : s'_x > 0\}$. Also let $\overline{f} \equiv f|_{(N',s')} = (f_x|_{N'})_{x\in\overline{X}}$ be the induced priority structure for the smaller problem. Then given $R \in \mathcal{R}^N$, $\overline{T}^f(N', s', R_{N'}) \equiv T^{\overline{f}}(R_{N'})$. We also denote the analogous extension of A^f to this context by $\overline{A}^f$. It is easy to check that both $\overline{T}^f$ and $\overline{A}^f$ violate consistency. Ergin [11] shows that $\overline{A}^f$ *is consistent if and only if* f *is weakly acyclic*. It turns out that even acyclicity is not restrictive enough to make $\overline{T}^f$ consistent.

Example 3. Let $N \equiv \{1, 2, 3, 4\}$, $X \equiv \{x, y\}$, $s_x = 2$, and $s_y = 2$. Let f be such that $f_x(1) < f_x(2) < f_x(3) < f_x(4)$ and $f_y(3) < f_y(4) < f_y(1) < f_y(2)$. For all $i \in \{2, 3, 4\}$, let $x\ P_i\ \emptyset\ P_i\ y$ and $y\ P_1\ \emptyset\ P_1 x$. Note that f is acyclic. It is easy to calculate that at this problem, $\overline{T}^f$ allots agent 4 the null object and agent 3 object x. Suppose now agent 3 leaves with x. In the reduced problem, agent 4 is allotted x.

[18] Because, although $f_x(2) < f_x(3)$, 2 envies 3 at this allocation.

[19] See Ergin [10] for characterizations of the class of consistent rules for house allocation problems. Also see Thomson [20] for a survey on consistency.

Next we introduce an even stronger form of acyclicity. A *weak-cycle* of *f* constitutes if the following two conditions are satisfied:

Loop condition: There are $i, j, k \in N$ and $x, y \in X$ such that $f_x(i) < f_x(j) < f_x(k)$ and $f_y(k) < f_y(i), f_y(j)$.

Scarcity condition: There is a (possibly empty) set $\overline{N}_x \subset N \backslash \{i, j, k\}$ such that $\overline{N}_x \subset U_x^f(k)$ and $|\overline{N}_x| = s_x - 1$.

The priority structure *f* is *strongly acyclic* if it has no weak-cycles.

It is easy to see that any strongly acyclic priority structure is also acyclic. The following theorem shows that strong-acyclicity is a sufficient and necessary condition for $\overline{T}^f$ to gain consistency.

Theorem 2. *Given an initial set of agents, an initial set of objects, an initial supply vector, and a priority structure* f, *the following statements are equivalent*:

(i) $\overline{T}^f$ *is consistent.*
(ii) *f is strongly acyclic.*

The next result shows that $\overline{A}^f$ once again has the advantage over $\overline{T}^f$ in terms of consistency.

Proposition 3. *Given a priority structure* f, *if* $\overline{A}^f$ *is not consistent for a problem, then* $\overline{T}^f$ *is not consistent for it either. However, the converse is not necessarily true.*

In closing, we present examples of weakly acyclic, acyclic, and strongly acyclic priority structures.[20]

Example 4. Let $N \equiv \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, $X \equiv \{a, b, c\}$, and $s_a = s_b = s_c = 2$. Consider the following three priority structures f, g, and h.

f: a	b	c	g: a	b	c	h: a	b	c
9	2	3	9	3	9	9	5	9
5	3	9	5	2	3	5	9	5
3	5	2	3	9	2	3	2	2
2	9	5	2	5	5	2	3	3
6	1	6	6	1	6	6	1	6
1	6	1	1	6	1	1	6	1
10	10	10	10	10	10	10	10	10
7	4	7	7	4	7	7	4	7
4	7	4	4	7	4	4	7	4

[20] We refer the keen reader to Ergin [11] for characterizations of weakly acyclic structures and to the working paper version of this paper (which is available upon request) for characterizations of acyclic and strongly acyclic structures.

The priority structure f is weakly acyclic but not acyclic (since $f_a(5) < f_a(3) < f_a(2)$ and $f_b(2) < f_b(3), f_b(5)$ with $N_a = \{9\}$). The priority structure g is acyclic but not strongly acyclic (since $f_a(9) < f_a(5) < f_a(2)$ and $f_b(2) < f_b(9), f_b(5)$ with $N_a = \{3\}$). Finally, the priority structure h is strongly acyclic. Loosely speaking, one can observe that as the acyclicity notion becomes stronger, the "correlation" between priority orders of different objects increases. Given a pair of objects a and b, weak-acyclicity partitions the set of agents into two classes with respect to their positions in the priority orders for the two objects: the *upper class* on whose members weak-acyclicity imposes no restrictions and the *lower class* whose members can differ only by one in their ranks across the pair of priority orders for objects a and b (see Ergin [11] for more). Our acyclicity notions, on the other hand, impose the same restrictions on the lower class members and they also bring restrictions on the upper class members as well.

For the three priority structures, consider first the agents whose ranks are greater than four with respect to any priority order. These agents constitute the lower class agents: $L = \{1, 4, 6, 7, 10\}$. Note that the ranks of these agents do not differ by more than one across any pair of objects. Since f is weakly acyclic, no change in the positions of the upper class agents would cause a strong-cycle to appear. As for g and h, this is not the case however. For example, consider g. Had we switched the positions of agents 2 and 3 in any one of the priority orders, a cycle would appear.

5. Conclusion

A social planner will find our results as important guidelines to help him choose between the agent-optimal stable mechanism (AOSM) and the top trading cycles mechanism (TTCM). If the priority structure is weakly-acyclic, then AOSM satisfies all of the desirable properties mentioned in this paper. However, TTCM may still violate consistency and the monotonicity properties. If the priority structure is acyclic, no matter which mechanism is used, the outcome is the same and all properties mentioned except consistency are satisfied by both mechanisms.

Furthermore, in any problem AOSM fails to be efficient or consistent, TTCM would also fail to be fair and consistent. We have shown however, that AOSM may continue to be efficient and consistent for a problem TTCM is unfair or inconsistent.

Acknowledgments

I am very much indebted to Professor William Thomson for his invaluable guidance and support. I am grateful to an anonymous referee for his/her useful suggestions and especially for pointing out an error in an earlier version of Theorem 1. I have benefited from the insightful comments of Haluk Ergin. I wish to thank Bettina Klaus for her detailed comments. Special thanks to Utku Unver, Atila Abdulkadiroğlu, Tayfun Sönmez, Hakan Gemici, Ayse Yazici, and Hervé Moulin for useful discussions. Any possible errors are my own responsibility.

Appendix

In what follows, for convenience we interpret T^f as an inheritance mechanism.

Proof of Theorem 1. We first show (i) $\Leftrightarrow$ (v). Then we prove Lemma 1. Next we show (i) $\Leftrightarrow$ (ii). Finally we show (iii) $\Leftrightarrow$ (v) and (iv) $\Leftrightarrow$ (v).

Proof of $(i) \Leftrightarrow (v)$.

($\Leftarrow$): Suppose by contradiction that there are an acyclic priority structure f, a preference profile $R \in \mathcal{R}^N$, agents $j, k \in N$, and an object $x \in X$ such that at allocation $T^f(R)$, j envies k with $T_k^f(R) = x$ and $f_x(j) < f_x(k)$. This means j does not inherit x at any step (of the algorithm). Then there are $i \in N$ with $f_x(i) < f_x(j)$, a step t, and a cycle C such that at step t, agents i and k are contained in cycle C in which k points to i for object x he is assigned. Hence, $i \neq k$. Furthermore, since j does not inherit x at any step, there exists a set $N_x \subset U_x^f(j)\backslash\{i\}$ of $s_x - 1$ agents each of whom is contained in a cycle in which some agent (possibly himself) points to him for x. Thus, $|N_x| = s_x - 1$. Also, since j envies k, agent j is not contained in a cycle at step t or at any earlier step. Let y be the object agent k is assigned at step t such that some agent (possibly i) in cycle C points to him for it. Then $f_y(k) < f_y(i), f_y(j)$, since otherwise i or j is removed before step t.

To show that f has a cycle, all that is left to show is that the set $N_x \cap (U_x^f(j)\backslash U_x^f(i)) \cap U_y^f(k)$ is empty [and hence, $N_x \subset U_x^f(i) \cup (U_x^f(j)\backslash U_y^f(k))$]. Suppose this set contains an agent $m \neq i, j, k$. Since $m \in N_x$, agent m is contained in a cycle in which some agent (possibly himself) points to him for x. Since $m \in U_x^f(j)\backslash U_x^f(i)$ and hence $f_x(i) < f_x(m)$, this means m is removed after i (that is, after step t). On the other hand, since $m \in U_y^f(k)$ and hence $f_y(m) < f_y(k)$, this means m is removed before k (that is, before step t). A contradiction.

($\Rightarrow$): Suppose f has a cycle. Then, there are agents $i, j, k \in N$, objects $x, y \in X$ such that $f_x(i) < f_x(j) < f_x(k)$ and $f_y(k) < f_y(i), f_y(j)$ and a set $N_x \subset N\backslash\{i, j, k\}$ such that $N_x \subset U_x^f(i) \cup (U_x^f(j)\backslash U_y^f(k))$ and $|N_x| = s_x - 1$. Let $R \in \mathcal{R}^N$ be as follows: for all $m \in N_x$ and all $a \in \varnothing \cup X\backslash\{x\}$, we have $x P_m a$. For all $t \in \{j, k\}$ and all $a \in X\backslash\{x\}$, we have $x P_t \varnothing P_t a$. For all $a \in X\backslash\{y\}$, $y P_i \varnothing P_i a$. Finally, for all $l \in N\backslash(N_x \cup \{i, j, k\})$ and all $a \in X$, we have $\varnothing P_l a$. It is easy to calculate that $T_j^f(R) = \varnothing$ and $T_k^f(R) = x$. But then, j envies k, contradicting the *fairness* of T^f.

Proof of Lemma 1. Let f be a priority structure that contains a strong-cycle but not a cycle. By Theorem 1 of Ergin [11], there is $R \in \mathcal{R}^N$ such that $A^f(R)$ is not *efficient*. On the other hand, $T^f(R)$ is clearly *efficient*. Thus, $A^f(R) \neq T^f(R)$. Furthermore, since (v)$\Rightarrow$(i) and f is acyclic, $T^f(R)$ is *fair*. But, by Balinski and Sönmez [4], the allocation selected by A^f Pareto dominates any other *fair* allocation. This contradicts the *efficiency* of $T^f(R)$. $\square$

Proof of $(i) \Leftrightarrow (ii)$.

($\Leftarrow$): This follows from the *fairness* of A^f.

($\Rightarrow$): By Balinski and Sönmez [4], the allocation selected by A^f Pareto dominates any other *fair* allocation. Then, the *efficiency* of T^f implies $T^f = A^f$.

Proof of $(iii) \Leftrightarrow (v)$.

($\Leftarrow$): Let f be an acyclic priority structure. Suppose by contradiction that $\widehat{T}^f$ is not *resource monotonic*. Then there are $d \in X$, $s'_d < s_d$, and $R \in \mathcal{R}^N$ such that at $\widehat{T}^f(s'_d, s_{-d}, R)$, an agent is better off and another is worse off as compared to $\widehat{T}^f(s, R)$. Furthermore, since $\widehat{T}^f$ is *efficient*, this means that an object changes hand between two agents making one better off and the other worse off. Let $\alpha \equiv \widehat{T}^f(s, R)$ and $\beta \equiv \widehat{T}^f(s'_d, s_{-d}, R)$. Hence, there are $x \in X$ and $j, k \in N$ such that $\alpha_j = \beta_k = x \neq \beta_j$, α_k and $x\, P_k \alpha_k$ and $x P_j \beta_j$. We first note that $x \neq d$. This is because the calculation of the outcome of $\widehat{T}^f$ for problems (s, R) and (s'_d, s_{-d}, R) are identical until the s'_d-th unit of d is allotted. (Because same cycles form and same agents and objects are removed until the s'_d-th unit of d is allotted.) Therefore, at allocation β, s'_d units of d are allotted to the same agents who used to be allotted this object at allocation α. By (v)$\Rightarrow$(i), α is *fair*. Thus, $x = \alpha_j P_k \alpha_k$ implies $f_x(j) < f_x(k)$. This together with $x = \beta_k P_j \beta_j$ implies that j envies k at β. Since $x \neq d$, by the very same argument used in the proof of the step (v)$\Rightarrow$(i), f has a cycle.

($\Rightarrow$): Suppose f has a cycle. Then, there are $i, j, k \in N$ and $x, y \in X$ such that $f_x(i) < f_x(j) < f_x(k)$ and $f_y(k) < f_y(i)$, $f_y(j)$ and $N_x \subset N\backslash\{i, j, k\}$ such that $N_x \subset U_x^f(i) \cup (U_x^f(j)\backslash U_y^f(k))$ and $|N_x| = s_x - 1$. Let $R \in \mathcal{R}^N$ be the same preference profile we considered above. Recall that $\widehat{T}_j^f(R, s) = \emptyset$ and $\widehat{T}_k^f(R, s) = x$. Let $s' \equiv (s'_a)_{a \in X}$ be such that for all $a \in X\backslash\{y\}$, $s'_a \equiv s_a$ and $s'_y \equiv 0$. It is easy to calculate that $\widehat{T}_j^f(s', R) = x$ and $\widehat{T}_k^f(s', R) = \emptyset$. But then, j is better off while k is worse off, contradicting the *resource monotonicity* of $\widehat{T}^f$.

Proof of $(iv) \Leftrightarrow (v)$.

($\Leftarrow$): Let f be an acyclic priority structure. Note that given $N' \subseteq N$, $f|_{N'}$ is also acyclic. By (v)$\Rightarrow$(i)$\Rightarrow$(ii), $T^f = A^f$ and $T^{f|_{N'}} = A^{f|_{N'}}$. Then the result follows from the *population monotonicity* of $\widetilde{A}^f$.

($\Rightarrow$): Suppose f has a cycle. Then, there are $i, j, k \in N$ and $x, y \in X$ such that $f_x(i) < f_x(j) < f_x(k)$ and $f_y(k) < f_y(i)$, $f_y(j)$ and $N_x \subset N\backslash\{i, j, k\}$ such that $N_x \subset U_x^f(i) \cup (U_x^f(j)\backslash U_y^f(k))$ and $|N_x| = s_x - 1$. Let $R \in \mathcal{R}^N$ be again the same preference profile we considered above. As before, $\widetilde{T}_j^f(N, R) = \emptyset$ and $\widetilde{T}_k^f(N, R) = x$. Suppose now i leaves. Let $N' \equiv N\backslash\{i\}$. It is easy to calculate that $\widetilde{T}_j^f(N', R_{N'}) = x$ and $\widetilde{T}_k^f(N', R_{N'}) = \emptyset$. But then, j is better off while k is worse off, contradicting the *population monotonicity* of $\widetilde{T}^f$. □

Proof of Proposition 2. Let f and $R \in \mathcal{R}^N$ be such that $A^f(R)$ is *inefficient* but $T^f(R)$ is *fair*. By Balinski and Sönmez [4], $A^f(R)$ Pareto dominates $T^f(R)$, contradicting the *efficiency* of T^f. See Example 2 as a counter example for the converse statement. □

Proof of Theorem 2. *Proof of* $(ii) \Rightarrow (i)$. Let f be a strongly acyclic priority structure. Suppose by contradiction that there are $\emptyset \neq N' \subsetneq N$ and $R \in \mathcal{R}^N$ such that $\overline{T}^f(N, s, R)|_{N'}$

$\neq \overline{T}^f(r_{N'}^{\overline{T}^f}(N,s,R))$. Let $e \equiv (N,s,R)$ and $e' \equiv r_{N'}^{\overline{T}^f}(N,s,R)$. Also, let $\alpha \equiv \overline{T}^f(e)$ and $\beta \equiv \overline{T}^f(e')$. Since $\alpha|_{N'} \neq \beta$, by the *efficiency* of $\overline{T}^f$, after the agents in $N \backslash N'$ leave, an object changes hand between two agents making one better off and the other worse off, i.e., there are $j,k \in N'$ and $x \in X$ such that $\alpha_j = \beta_k = x \neq \beta_j$, α_k and $x P_k \alpha_k$ and $x P_j \beta_j$.

Since f is also acyclic, by (v)⇒(i) of Theorem 1, α is *fair*. Then, $x = \alpha_j P_k \alpha_k$ implies $f_x(j) < f_x(k)$. Then, since $\beta_k = x P_j \beta_j$, agent j envies k at allocation β. Let s' be the supply vector for problem e'. Then by (v)⇒(i) of Theorem 1, $f|_{(N',s')}$ has a cycle. In particular, by the argument in the proof of (v)⇒(i) of Theorem 1, there are $i \in N' \backslash \{j,k\}$ and $y \in X \backslash \{x\}$ such that $f_x(i) < f_x(j) < f_x(k)$ and $f_y(k) < f_y(i)$, $f_y(j)$ and $N'_x \subset N' \backslash \{i,j,k\}$ such that $N'_x \subset U_x^{f|_{(N',s')}}(j) \subset U_x^f(k)$ and $|N'_x| = s'_x - 1$. We shall show that there is a set $\overline{N}_x \subset N \backslash \{i,j,k\}$ such that: (1) $\overline{N}_x \subset U_x^f(k)$ and (2) $|\overline{N}_x| = s_x - 1$, i.e., f has a weak-cycle. Let $\overline{N}_x$ be the set obtained from N'_x by including those agents in $N \backslash N'$ who are allotted x at problem e, i.e., $\overline{N}_x \equiv N'_x \cup \{m \in N \backslash N' : \overline{T}_m^f(e) = x\}$. Since α is *fair* and $x P_k \alpha_k$, for all $m \in \overline{N}_x \backslash N'_x$, we have $f_x(m) < f_x(k)$. This together with $N'_x \subset U_x^f(k)$ proves (1). Next, note that $|\{m \in N \backslash N' : \overline{T}_m^f(e) = x\}| = s_x - s'_x$. Then, this together with $|N'_x| = s'_x - 1$ proves (2).

Proof of $(i) \Rightarrow (ii)$. Suppose f has a weak-cycle. Then, there are $i,j,k \in N$ and $x,y \in X$ such that $f_x(i) < f_x(j) < f_x(k)$ and $f_y(k) < f_y(i)$, $f_y(j)$ and $\overline{N}_x \subset N \backslash \{i,j,k\}$ such that $\overline{N}_x \subset U_x^f(k)$ and $|\overline{N}_x| = s_x - 1$. Let $L_x^f(i) \equiv \{j \in N |\ f_x(j) > f_x(i)\}$. We consider two cases:

Case 1: $\overline{N}_x \cap L_x^f(i) \cap U_y^f(k) = \emptyset$: Let $R \in \mathcal{R}^N$ be the same preference profile we considered in the proof of Theorem 1 with $N_x = \overline{N}_x$. One can easily calculate that $\overline{T}_k^f(N,s,R) = x$. Suppose i leaves with his allotment. Let $N' \equiv N \backslash \{i\}$. It is again easy to calculate that $\overline{T}_k^f(r_{N'}^{Tf}(N,s,R)) = \emptyset$. A contradiction to the *consistency* of $\overline{T}^f$.

Case 2: $\overline{N}_x \cap L_x^f(i) \cap U_y^f(k) \neq \emptyset$: Let $m \in N \backslash \{i,j,k\}$ be the agent with the highest priority for y among the agents in $\overline{N}_x \cap L_x^f(i) \cap U_y^f(k)$. Let $R \in \mathcal{R}^N$ be the same preference profile as in Case 1. It is easy to calculate that $\overline{T}_k^f(N,s,R) = \emptyset$ (because i forms a cycle with m and is allotted y and k does not inherit x at any step), $\overline{T}_i^f(N,s,R) = y$, and $\overline{T}_j^f(N,s,R) = x$ (because j forms a self-cycle). Suppose all agents but i, j, and k leave with their allotments. Let $N' \equiv \{i,j,k\}$. We now have $\overline{T}_k^f(r_{N'}^{Tf}(N,s,R)) = x$ (because i forms a cycle with k in which he exchanges x for y). A contradiction to the *consistency* of $\overline{T}^f$. □

Proof of Proposition 3. Let f and (N,s,R) be such that $\overline{A}^f$ is not *consistent* but $\overline{T}^f$ is *consistent* for (N,s,R). Then by Ergin [11] (see the proof of (i)⇒(iii) in Theorem 1 of Ergin [11]), $\overline{A}^f(N,s,R)$ is also *inefficient*. Then by Proposition 2, $\overline{T}^f(N,s,R)$ is *unfair*. Hence there are $j,k \in N$, and $x \in X$ such that $x = \overline{T}_k^f(N,s,R)\ P_j\ \overline{T}_j^f(N,s,R)$ and $f_x(j) < f_x(k)$. Now let all agents but j and k leave with their allotments. For the reduced problem, $\overline{T}^f$ allots x to j contradicting the *consistency* of $\overline{T}^f$ for (N,s,R). For the converse statement consider Example 3. Note that f is weakly acyclic. Hence by Ergin [11], $\overline{A}^f$ is *consistent* for the given problem, but $\overline{T}^f$ is not. □

References

[1] A. Abdulkadiroğlu, T. Sönmez, Random serial dictatorship and the core from random endowments in house allocation problems, Econometrica 66 (1998) 689–701.
[2] A. Abdulkadiroğlu, T. Sönmez, House allocation with existing tenants, J. Econ. Theory 88 (1999) 233–260.
[3] A. Abdulkadiroğlu, T. Sönmez, School choice: a mechanism design approach, Amer. Econ. Rev. 93 (2003) 729–747.
[4] M. Balinski, T. Sönmez, A tale of two mechanisms: student placement, J. Econ. Theory 84 (1999) 73–94.
[5] A. Bogomolnaia, H. Moulin, A new solution to the random assignment problem, J. Econ. Theory 100 (2001) 295–328.
[6] L.E. Dubins, D.A. Freedman, Machiavelli and the Gale–Shapley algorithm, Amer. Math. Monthly 88 (1981) 485–494.
[7] L. Ehlers, B. Klaus, S. Pápai, Strategy-proofness and population monotonicity for house allocation problems, J. Math. Econ. 38 (2002) 329–339.
[8] L. Ehlers, B. Klaus, Resource monotonicity for house allocation problems, Int. J. Game Theory 32 (2004) 545–560.
[9] L. Ehlers, Coalitional strategy-proof house allocation, J. Econ. Theory 105 (2002) 298–317.
[10] H. Ergin, Consistency in house allocation problems, J. Math. Econ. 34 (2000) 77–97.
[11] H. Ergin, Efficient resource allocation on the basis of priorities, Econometrica 70 (2002) 2489–2497.
[12] D. Gale, L.S. Shapley, College admissions and the stability of marriage, Amer. Math. Monthly 69 (1962) 9–15.
[13] O. Kesten, Coalitional strategy-proofness and resource monotonicity for house allocation problems, Mimeo, University of Rochester, 2003.
[14] O. Kesten, A new principle for indivisible good allocation problems, Mimeo, University of Rochester, 2003.
[15] O. Kesten, On two kinds of manipulation in two-sided matching markets, Mimeo, University of Rochester, 2004.
[16] S. Pápai, Strategy-proof assignment by hierarchical exchange, Econometrica 68 (2000) 1403–1433.
[17] A. Roth, A. Postlewaite, Weak versus strong domination in a market with indivisible goods, J. Math. Econ. 4 (1977) 131–137.
[18] A. Roth, M. Sotomayor, Two-sided Matching, Cambridge University Press, New York, 1990.
[19] L. Shapley, H. Scarf, On cores and indivisibility, J. Math. Econ. 1 (1974) 23–28.
[20] W. Thomson, The consistency principle, in: T. Ichiishi, A. Neyman, Y. Tauman (Eds.), Game Theory and Applications, Academic Press, New York, 1990, pp. 187–215.
[21] W. Thomson, Population monotonic allocation rules, in: W.A. Barnett, H. Moulin, M. Salles, N. Schofield (Eds.), Social Choice, Welfare and Ethics, Cambridge University Press, Cambridge, 1995.
[22] W. Thomson, The theory of fair allocation, book manuscript, 2000.