



LUND UNIVERSITY
School of Economics and Management

Intertemporal Allocation of Indivisible and Durable Goods

MASTER ESSAY II

Jörgen Kratz

May 30, 2013

Abstract

This paper presents a futures mechanism, as defined by Kurino (2009), allocating a set of indivisible and durable goods among a set of agents over multiple time periods. The mechanism is shown to satisfy individual rationality, Pareto efficiency and non-bossiness for allocation problems with or without endowments. The paper also shows that the mechanism does not satisfy strategy-proofness and presents the conditions under which the mechanism is manipulable.

Keywords: intertemporal allocation; futures mechanism; indivisible goods; endowments

Acknowledgements

I would like to thank Tommy Andersson for his support as a supervisor during the production of this paper.

Contents

1	Introduction	3
1.1	Purpose	6
1.2	Related literature	6
1.3	Overview	7
2	The model: Sets and definitions	7
2.1	Agents and objects	7
2.2	Matching plans and assignments	8
2.3	Preferences	9
2.4	Allocation problems and mechanisms	9
3	Mechanism design	10
3.1	Classes of mechanisms	10
3.2	The φ mechanism	13
3.3	The algorithm	16
3.4	Example	21
4	Properties	25
4.1	Positive results	29
4.2	Negative results	35
5	Concluding remarks	38
6	References	40

1 Introduction

Consider two siblings, Yngve and Ingvor. Yngve and Ingvor are playing in their backyard. In the backyard, there is a short skipping rope and a small trampoline. Only one person may use the skipping rope or the trampoline at a time and no one may use the skipping rope while jumping on the trampoline. Dinner is ready in 20 minutes. If Yngve and Ingvor divide these 20 minutes into two 10 minute periods, they have a number of different alternatives to consider. Yngve could, for example, jump on the trampoline in both periods, while Ingvor spends 20 minutes skipping. Another option would be for Yngve to spend the first period jumping on the trampoline and the second period skipping, while Ingvor is skipping in the first period and jumping on the trampoline in the second period. Yngve and Ingvor cannot agree on what to do, so they ask their mother to decide for them. Which of the many alternatives should she choose, and why? Solutions to this kind of problem are studied in a field known as mechanism design.

The above is an example of the problem of how to allocate a number of goods, such as skipping ropes and trampolines, among a group of agents, such as Yngve and Ingvor. *Allocation problems* of this kind have been studied extensively and have many different applications, ranging from school choice and housing to kidney exchange programs. A *mechanism designer*, represented by Yngve's and Ingvor's mother in the example above, is interested in solving allocation problems and does so by selecting a mechanism. A *matching mechanism* is a rule that, given its inputs, selects a *matching* between agents and goods. For allocation problems with indivisible goods, the mechanism selects a matching such that each good is assigned to at most one agent. Whenever there is more than one time period, such as in the example above, the mechanism must select one matching for each period. Such a sequence of matchings is referred to as a *matching plan*. For instance, a mechanism could select a matching plan, under which Yngve is assigned the skipping rope in period one and the trampoline in period two, while Ingvor is assigned the trampoline in period one and the skipping rope in period two. Imagine that Yngve has called dibs on the trampoline. Yngve is then already assigned the trampoline when their mother is making her decision. Whenever some goods start out assigned to agents, the allocation problem is known as an allocation problem *with endowments*. This paper will limit its focus to a class of mechanisms known as *direct mechanisms*. Under this framework, the agents are of different *types* and each agent is asked to reveal his or her type. The mechanism

makes use of this information when selecting matchings or matching plans between agents and goods. Suppose that Yngve is the type that likes to climb trees, while Ingvor is of a more down-to-earth type. Their mother, who does not know as much about her children as she probably should, will ask Yngve and Ingvor to reveal their types. She will then use their responses to select a matching plan under which, for example, Yngve jumps on the trampoline in both periods and Ingvor is skipping in both periods. An agent's type is often synonymous with his or her preferences over the different goods. If this were the case, Yngve and Ingvor would simply be asked if they prefer skipping to jumping on the trampoline or vice versa.

Mechanisms are normally evaluated by examining which *properties* they satisfy. A mechanism is *individually rational* if no agent is ever made worse off by the mechanism. For example, suppose that Yngve has called dibs on the trampoline, because he really likes to jump on the trampoline. He would then be made worse off if his mother chose not to respect his dibs and let Ingvor jump on the trampoline in both periods. A matching or matching plan is *Pareto efficient* if no agent can be made better off without making some other agent worse off and a mechanism is Pareto efficient if it always selects Pareto efficient matchings or matching plans. Suppose that Yngve wants to jump on the trampoline and that Ingvor prefers skipping. If their mother decides that Yngve will be skipping in both periods and Ingvor will be jumping on the trampoline in both periods, this clearly does not represent a Pareto efficient matching plan. Both Yngve and Ingvor would be made better off if she changed her mind and decided that Yngve should be jumping on the trampoline in both periods and Ingvor should be skipping in both periods. However, if both Yngve and Ingvor hate skipping ropes and only want to jump on the trampoline, then anything their mother decides will represent a Pareto efficient matching plan. A mechanism is *non-bossy* if it is impossible for any agent to affect any other agent's assignment without affecting his or her own assignment as well. Assume that there are two skipping ropes, one red and one blue skipping rope. Further assume that Yngve has reported his true type to his mother and that he was assigned the trampoline in both periods, while Ingvor was assigned the blue skipping rope in both periods. The mechanism is non-bossy if there is nothing Yngve could have reported to his mother that would have made her decide to assign Yngve the trampoline in both periods and Ingvor the red skipping rope in any of the two periods. A mechanism is *strategy-proof* if no agent can benefit from reporting a type that is not his or her true type. In the framework employed in this paper, this is equivalent to no

agent being able to benefit by misrepresenting his or her preferences. For example, suppose Yngve knows that his mother is so mean that she will always choose the activity he likes the least for him. Yngve could then benefit by lying about his preference for trampoline jumping and report that he prefers playing with the skipping rope.

Most of the literature focuses on allocation problems where all reallocations take place at a single point in time. Some authors have introduced time in their models, studying *intertemporal* allocation problems such as the one faced by Yngve's and Ingvor's mother. Time is typically discrete, consisting of a finite number of periods. Kurino (2009) defines two classes of mechanisms solving intertemporal allocation problems. A *spot mechanism* is a one-period mechanism that is repeated in each time period. In each period, agents report their preferences over the goods in the economy and the mechanism selects a matching between agents and goods for this period. A *futures mechanism* is a mechanism where agents report their preferences over the entire time interval during which they participate in the economy. The futures mechanism uses this information to select a matching plan, under which each agent is assigned one good for each time period in which he or she participates in the economy.

While deciding who will jump on the trampoline and who will use the skipping rope might seem like a trivial problem, there are several important real world applications that motivate the study of intertemporal allocation problems. Consider, for example, a company in charge of providing housing for students at a university. Suppose that there are three types of student housing: dorm rooms, small apartments and large apartments; and that each student stays at the university for three years. Assuming that the company is interested in finding an allocation of rooms and apartments that the students are happy with, it will not want to force students to live in the same room or apartment for three years. One solution would be to divide the three years into three periods and match each student with a room or apartment each year. The easiest way to accomplish this would be to repeat a one-period mechanism once a year. That way, each student would have the option to change his or her preferences and possibly be assigned a different room or apartment in the second or third year. Suppose, however, that students know before the start of the first year how their preferences for different types of housing will change over the course of the three years. Perhaps some students know that they want to live in a dorm room the first year to meet new people and that they will want their own apartment in the second and third years. Other students might have families and children and require large

apartments for the full duration of their studies. The company could then implement a futures mechanism to take such preferences into account. The students would report their preferences over sequences of rooms and apartments of the form (dorm room in year 1, small apartment in year 2, large apartment in year 3) or (small apartment in year 1, small apartment in year 2, dorm room in year 3). The company would finally use these reports to assign such a sequence of rooms and apartments to each student.

1.1 Purpose

The purpose of this paper is to present a futures mechanism that solves intertemporal allocation problems with endowments. Furthermore, the purpose is to prove that this mechanism satisfies Pareto efficiency, individual rationality and non-bossiness, and to prove that it does not satisfy strategy-proofness as well as under which conditions it is manipulable.

1.2 Related literature

This paper is related to the literature on intertemporal allocation problems with indivisible goods. One of the earliest papers dealing with a single-period matching problem is Gale and Shapley's (1962) paper on college admissions and the stability of marriage. The use of trading cycles in section 3 is heavily influenced by the paper by Shapley and Scarf (1974) introducing the concept of top trading cycles, as well as the top trading cycles mechanism for single-period house allocation problems in Abdulkadiroglu and Sönmez (1999). The mechanism described in section 3.2 differs from the mechanism introduced by Abdulkadiroglu and Sönmez in the sense that it selects trading cycles, but not necessarily *top* trading cycles. Furthermore, the mechanism in section 3.2 assumes that their preferences in subsequent periods are known by the agents in the first period, in contrast to Abdulkadiroglu and Loertscher (2007) who study a two-period allocation problem where the agents' types in the second period are unknown. Kurino (2009, 2013) and Bloch and Cantala (2011) study a multi-period house allocation problem in which agents enter and exit the model over time, as opposed to the fixed set of agents assumed in this paper. Ünver (2009) considers a dynamic kidney allocation problem where time is introduced as a factor and Dur (2011) considers a dynamic two-period school choice problem, where it is considered preferable to assign siblings to the same schools. Zou, Gujar and Parkes (2010) examine a dynamic intertemporal allocation problem, where agents are ranked in

accordance with their point of arrival. In this setting, they examine the efficiency gains of adopting manipulable mechanisms whenever a sufficiently large share of the agents fail to act strategically and instead report their preferences truthfully.

All of the papers mentioned above that deal with intertemporal allocation problems present some type of spot mechanism. One of the benefits of considering futures mechanisms, such as the futures mechanism presented in section 3.2, is their ability to take more advanced preferences into account. If a spot mechanism is used, the mechanism designer only has access to agents' preferences over objects in the present period. If a futures mechanism is used, the mechanism designer has access to agents' preferences over several periods. As shown in section 3.1, such information could be used to find better matching plans.

1.3 Overview

Section 2 presents the sets and definitions used throughout the paper. Section 3 begins by motivating the study of futures mechanisms. It then goes on to define and describe the mechanism. Finally, an algorithm that finds the matching plan selected by the mechanism is presented. Section 4 shows that the mechanism satisfies individual rationality, Pareto efficiency and non-bossiness. Furthermore, it shows that the mechanism does not satisfy strategy-proofness and presents the conditions under which an agent may benefit from misrepresenting his or her preferences. Section 5 provides some concluding remarks.

2 The model: Sets and definitions

2.1 Agents and objects

Time consists of an interval of T discrete *periods* $t \in [1, T] \subset \mathbb{N}$. There are N^t *agents* $i^t \in I^t \subseteq \mathcal{I}$, where I^t is the set of agents existing in period $t \in [1, T]$ and \mathcal{I} is the *agent space*. An agent may exist in several periods. Define the set of agents existing in any period $t \in [1, T]$ as $I = \{i \mid i^t \in I^t \text{ for some } t \in [1, T]\}$. Agent i^t should be interpreted as the period t representation of agent i , such that $i^t \in I^t \iff i \in I^t$. There are M^t indivisible and durable goods $a^t \in A^t \subseteq \mathcal{A}$ called *objects*, where A^t is the set of objects existing in period $t \in [1, T]$ and \mathcal{A} is the *object space*. The durability of objects indicates that objects may exist in several periods and the indivisibility of objects indicates that an object may

be consumed by at most one agent in each period. Define the set of objects existing in any period $t \in [1, T]$ as $A = \{a \mid a^t \in A^t \text{ for some } t \in [1, T]\}$. As with agents, a^t should be interpreted as the period t representation of a , such that $a^t \in A^t \iff a \in A$. An *economy* is defined as a three-tuple $\langle T, I, A \rangle \in \mathbb{N} \times \mathcal{I} \times \mathcal{A}$. In this paper, it is assumed that for all $t, t' \in [1, T]$, $I^t = I^{t'} = I$ and $A^t = A^{t'} = A$, and by implication, $N^t = N^{t'} = N$ and $M^t = M^{t'} = M$. In other words, no agents enter or exit the model and there is no production or destruction of objects. In addition, denote the *null object* by a_0 . The null object is interpreted as "no object." A sequence of T objects $a \in A \cup a_0$ consisting of one object in each period $t \in [1, T]$ is called a *consumption path*, defined by the ordered set $x \equiv \{x(t) \mid \forall t \in [1, T] x(t) \in A^t \cup a_0\}$. The *consumption path space* X is defined by $X \equiv \prod_{t=1}^T \{A^t \cup a_0\}$, where \prod denotes the Cartesian product. Throughout this paper $J(j)$ will denote the j 'th element in J whenever J is an ordered set.

2.2 Matching plans and assignments

A *matching plan* is an injective function $\mu: I \rightarrow X$. It could also be written as $\mu = \{\mu_t \mid t \in [1, T]\}$, where the *period t matching*, μ_t , is an injective function $\mu_t: I^t \rightarrow A^t \cup a_0$. In other words, a matching plan maps each agent $i \in I$ to some object $a \in A \cup a_0$ in each period, together constituting a consumption path $x \in X$. The consumption path agent i is mapped to is referred to as agent i 's *assignment* and the object agent i is mapped to in period t is referred to as agent i 's *period t assignment*. The *matching plan space* is denoted by \mathcal{M} . The indivisibility of objects implies that an object $a^t \in A^t$ may be assigned to at most one agent $i^t \in I^t$, or equivalently that an object $a \in A$ may be assigned to at most one agent $i \in I$ in each period $t \in [1, T]$. The null object may, however, be assigned to any number of agents in each period. The assignment of agent i under μ is denoted by $\mu(i)$ and the period t assignment of i is denoted by $\mu_t(i)$. The sets of assignments of some $I' \subseteq I$ and $I'^t \subseteq I^t$ are denoted by $\mu(I')$ and $\mu_t(I'^t)$ respectively. Even though a period t matching is not a bijective function and technically not invertible, for notational ease, the agent assigned object a in period t under μ will be referred to as $\mu_t^{-1}(a)$. The *endowment* of agent i is denoted by $\lambda(i) \in X$ and the endowments of $I' \subseteq I$ are denoted by $\lambda(I')$, where $\lambda \in \mathcal{M}$ is the original matching plan before any reassignments have been made. $\lambda(i)$ always contains the same object $a \in A \cup a_0$ in all periods. This means that $\forall t, t' \in [1, T] \lambda_t(i) = a \iff \lambda_{t'}(i) = a$. Let $A_A^t \subseteq A^t$ denote the set of objects assigned

to some agent $i \in I^t$ and let $A_U^t \subseteq A^t$ denote the set of objects unassigned to any agent in period t . By construction, $A_A^t \cap A_U^t = \emptyset$.

2.3 Preferences

Each agent $i \in I$ has a complete and transitive strict *preference relation* $P_i \in \Omega$ on X , where Ω is the *preference domain*. In other words, P_i is an ordered set containing all elements in X . $xR_i x'$ means that x is weakly preferred to x' and $xP_i x'$ means that x is strictly preferred to x' . A consumption path $x \in X$ is said to be *acceptable* to i if and only if $xR_i \lambda(i)$. Note that as $P_i \in \Omega$ is a strict preference relation, $xR_i x' \wedge \neg(xP_i x') \iff x = x'$. Ω is restricted such that, everything else equal, all agents prefer being assigned some $a^t \in A^t$ to being assigned a_0 in any period. A *preference profile* is defined by $P = \{P_i \mid i \in I\} \in \prod_{i=1}^N \Omega$. If $\mu, \mu' \in \mathcal{M}$, then $\mu P_i \mu' \iff \mu(i)P_i \mu'(i)$.

2.4 Allocation problems and mechanisms

An *allocation problem* is defined as a five-tuple $\langle T, I, A, P, \lambda \rangle \in \mathbb{N} \times \mathcal{I} \times \mathcal{A} \times \prod_{i=1}^N \Omega \times \mathcal{M}$. If $\forall i \in I, \forall t \in [1, T] \lambda^t(i) = a_0$, it is called an allocation problem *without endowments*. Otherwise, it is called an allocation problem *with endowments*. If $T \geq 2$, it is called an *intertemporal* allocation problem. There is a mechanism designer who is interested in finding a solution to some allocation problem. The mechanism designer creates or inherits a ranking of all agents and agents are allowed to send some message to the mechanism designer, typically communicating their preferences for different objects or consumption paths. This information is used to select a solution to the allocation problem, in the form of a matching plan $\mu \in \mathcal{M}$.

The ranking of the agents is an externally determined strict *priority structure* given by the bijective function $f: ([1, N] \subset \mathbb{N}) \rightarrow I$. However, for notational ease, the priority structure f will be treated as an ordered subset of I , such that $f(1)$ denotes the agent of highest priority and $f(\#f)$ denotes the agent of lowest priority. The $\#$ sign denotes the number of elements in the set it precedes. If $i, i' \in f$, then $if i'$ is interpreted as i being of higher priority than i' under f . Let F denote the *priority structure space*, given some $I \in \mathcal{I}$. Priority structures can be determined in a number of different ways. For example, the priority structure could reflect the preferences of the owner of the objects, it could reflect the time an agent has been waiting for an object or it could simply be randomized.

A *strategy* of agent i , which can be thought of as a message sent by agent i to the mechanism designer, is denoted by $S_i \in \mathcal{S}_i$, where \mathcal{S}_i is agent i 's *strategy space*. A *strategy profile* S is defined by $S \equiv \{S_i \mid i \in I\}$. If $I' \subseteq I$, then $S_{I'} \equiv \{S_i \mid i \in I'\}$. In this multi-period model, a *matching mechanism* consists of a strategy space \mathcal{S}_i for each $i \in I$ and an injective function $\Gamma: \prod_{i=1}^N \mathcal{S}_i \times F \rightarrow \mathcal{M}$ that selects some matching plan $\mu \in \mathcal{M}$ for each combination of strategy profiles and priority structures $(S, f) \in \prod_{i=1}^N \mathcal{S}_i \times F$. This is an adaptation of the definition used by Abdulkadiroglu and Sönmez (1999) to intertemporal allocation problems. A *direct matching mechanism* is a matching mechanism where $\forall i \in I \ \mathcal{S}_i = \Omega$. As preference relations $P_i \in \Omega$ are ordered sets, element j in S_i can be denoted by $S_i(j)$ and the element in $S_i(j)$ corresponding to period t can be denoted by $S_i^t(j)$ whenever the mechanism is a direct mechanism. All mechanisms discussed in this paper are direct matching mechanisms. A matching mechanism consisting of the strategy space Ω and a function Γ will simply be referred to as the Γ mechanism. If $\Gamma(S, f) = \mu$, then $\Gamma_i(S, f) = \mu(i)$.

3 Mechanism design

3.1 Classes of mechanisms

Matching mechanisms are most commonly studied in a static one-period ($T = 1$) framework. In this framework, $X = A$ and the preferences are defined over A . The mechanism thus selects a single *period 1 matching* μ^1 , rather than a matching plan μ . For intertemporal allocation problems, Kurino (2009) defines two classes of mechanisms. A *spot mechanism* is a mechanism that selects a matching plan by repeating a one-period mechanism in each period. Spot mechanisms are appealing since static one-period mechanisms can easily be converted to spot mechanisms for intertemporal allocation problems. A *futures mechanism* is a mechanism where agents are asked to report their preferences over the entire period during which they participate in the economy. A spot mechanism is less demanding of the agents as they are only asked to report their preferences over objects in A , whereas a futures mechanism requires agents to report preferences over X , a significantly larger set whenever $T \geq 2$. At the same time, a futures mechanism can take more advanced preferences into account. To understand the difference, consider a simple mechanism known as serial dictatorship (*SD*) that will select the same matching plan if implemented as a

spot mechanism (SD^S) as it would if implemented as a futures mechanism (SD^F). In the one-period case, SD assigns the agent of highest priority his most preferred object in A^1 and removes that object from A^1 . The agent of second highest priority is then assigned his most preferred object out of those remaining in A^1 , and so on until all objects have been assigned to some agent or all agents have been assigned some object. SD has been proven to be strategy-proof and Pareto efficient (Svensson, 1994). Although it will not be proven in this paper, the arguments hold for SD^S and SD^F as well. When implemented as a spot mechanism, the process is simply repeated for A^2, A^3, \dots, A^T . When implemented as a futures mechanism, the agent of highest priority is assigned his most preferred consumption path $x \in X$ and all consumption paths involving some $a^t \in x$ are removed from X . The agent of second highest priority is then assigned his most preferred consumption path out of those remaining in X , and so on until all agents have been assigned some consumption path. If $T = 3$ and $(a_1, a_2, a_1) \in X$ is agent $f(1)$'s most preferred consumption path, this simply corresponds to $f(1)$ preferring object a_1 to all other objects in A^1 , a_2 to all other objects in A^2 and a_1 to all other objects in A^3 . If these preferences are reported, SD^S would assign $f(1)$ the consumption path (a_1, a_2, a_1) . The reader can confirm that all other agents in I would be assigned the same consumption paths under both SD^S and SD^F as well. The reason why SD^S and SD^F select the same matching plan is that there are no property rights in the serial dictatorship mechanism. To see this, consider a well known extension of SD known as serial dictatorship with waiting list, (SD_{WL}) which introduces property rights. SD_{WL} assigns objects to agents in the same way as SD , with the difference that agents can only be assigned objects that are not already assigned to some other agent. Whenever an agent is assigned a new object, his previous assignment becomes unassigned. SD_{WL} thus ensures the right of each agent to keep his or her current assignment until some more preferred object is available. The serial dictatorship with waiting list spot mechanism (SD_{WL}^S) is simply SD_{WL} repeated in each period. The serial dictatorship with waiting list futures mechanism (SD_{WL}^F) assigns $f(1)$ his or her most preferred consumption path $x \in X$ such that $x(t)P_{f(1)}\lambda_t(f(1))$ for all $t \in [1, T]$ and such that each $a^t \in x$ is either $\lambda_t(f(1))$ or unassigned. All $a^t \in x$ are assigned to $f(1)$ and all a^t for which $a^t \in \lambda(f(1))$ and $a^t \notin x$ become unassigned. The same step is then repeated in order for $f(2)$ through $f(\#f)$. SD_{WL}^S and SD_{WL}^F do not necessarily select the same matching plans. To see this, consider an allocation problem with endowments $\langle T, I, A, P, \lambda \rangle$, where $T = 3$, $I = \{i_1, i_2\}$, $A_A = \{a_1, a_2\}$, $A_U = \{a_3\}$, $\lambda(i_1) = a_1$, $\lambda(i_2) = a_2$ and P is given by

P_1	P_2
(a_1, a_3, a_1)	(a_2, a_1, a_1)
(a_1, a_1, a_1)	(a_2, a_1, a_2)
\dots	(a_2, a_2, a_2)
	\dots

Furthermore, let $f = (i_1, i_2)$. Under SD_{WL}^F , i_1 is guaranteed (a_1, a_1, a_1) , but is assigned (a_1, a_3, a_1) , which is strictly preferred to (a_1, a_1, a_1) . This results in $a_1 \in A^2$ being unassigned. Next, i_2 is guaranteed (a_2, a_2, a_2) , but can be assigned (a_2, a_1, a_2) , which is strictly preferred to (a_2, a_2, a_2) , as $a_1 \in A^2$ is now unassigned. This matching plan is, however, not necessarily selected by SD_{WL}^S . In period 1, i_1 and i_2 are both assigned their endowments. In period 2, i_1 is guaranteed a_1 , but would prefer to be assigned a_3 . Suppose i_1 reports that $a_3 \in A^2$ is preferred to $a_1 \in A^2$, then $a_1 \in A^2$ would become unassigned. As agent i_2 prefers to be assigned a_1 to a_2 and a_3 in both period 2 and period 3, i_2 would finally be assigned his or her most preferred consumption path (a_2, a_1, a_1) . Agent i_1 , on the other hand, would be assigned either (a_1, a_3, a_2) or (a_1, a_3, a_3) , to both of which agent i_1 prefers (a_1, a_1, a_1) . In this example, i_2 gained from i_1 's inability to predict i_2 's period 3 preferences correctly. Such uncertainties regarding the future preferences of other agents might induce agents to report preferences that make all agents worse off. Consider the same allocation problem as above, with a new preference profile P' , given by

P'_1	P'_2
(a_1, a_3, a_1)	(a_2, a_1, a_2)
(a_1, a_1, a_1)	(a_2, a_2, a_2)
\dots	\dots

SD_{WL}^F would assign (a_1, a_3, a_1) to i_1 , and (a_2, a_1, a_2) to i_2 . SD_{WL}^S would also assign a_1 to i_1 and a_2 to i_2 in period 1. However, as i_1 has no information on i_2 's period 3 preferences, i_1 might choose to hold on to a_1 out of fear of not being able to get it back in period 3. As a result, i_1 is assigned (a_1, a_1, a_1) and i_2 is assigned (a_2, a_2, a_2) . In this scenario, SD_{WL}^F selected a Pareto efficient matching plan, while SD_{WL}^S did not. Since there was no risk of not being assigned a_1 in period 3, i_1 had no incentive not to choose to be assigned a_3 in period 2. This made a_1 available in period 2, making i_2 's most preferred consumption path attainable to i_2 as well. The above example showed a situation in which the SD_{WL}^F selected a matching plan preferred by all agents to the matching plan selected by SD_{WL}^S . Under SD_{WL}^F , both the mechanism designer and the agents have access to more information on

assignments in subsequent periods than they do under SD_{WL}^S . This information may, as demonstrated, be used to find better matchings between agents and objects. Such potential benefits motivate the study of futures mechanisms even when spot mechanisms are easier to implement. It is easy to see that SD^F is not individually rational whenever there are endowments. Furthermore, SD_{WL}^F is only Pareto efficient for allocation problems without endowments, in which case it reduces to SD^F . That SD_{WL}^F is not Pareto efficient is easily confirmed by considering an allocation problem with endowments, where $A_U^t = \emptyset$ for each $t \in [1, T]$ and $\lambda(i)$ is the least preferred consumption path for all agents $i \in I$. The φ mechanism described in section 3.2 is a futures mechanism that satisfies both individual rationality and Pareto efficiency for allocation problems with endowments.

3.2 The φ mechanism

To describe the φ mechanism, some additional concepts are needed. A *trading cycle*, or simply a *cycle*, c^t , is an ordered set (j_1, j_2, \dots, j_k) of agents and objects $j \in I^t \cup A^t$, where $j_1 \rightarrow j_2, j_2 \rightarrow j_3, \dots, j_{k-1} \rightarrow j_k$ and $j_k \rightarrow j_1$ and “ \rightarrow ” is read as “points to”. Further define “ \rightarrow ” such that $i \rightarrow x \iff \forall a \in x \ i \rightarrow a$ and $x \rightarrow i \iff x(1) \rightarrow i$. Which $a^t \in A^t$ or $x \in X$ each $i \in I$ points to is determined by $P_i \in \Omega$ and which $i \in I$ each $a^t \in A^t$ points to is determined by some $f \in F$ in a manner described in section 3.3. Given some $f \in F$, a cycle c^t is defined exclusively by which $a \in A^t$ each $i \in c^t$ points to. A cycle is permitted to involve only one agent and one object. In the algorithm in section 3.3, $|c^t|$ will be used to denote the number of cycles that have formed in period t at the step in consideration. Let a *cycle set* be defined as

$$C = \{c^t \mid i \in c^t \in C \implies \forall t' \in [1, T] \ (\exists c^{t'} \in C: i \in c^{t'})\}. \quad (1)$$

In other words, C is a nested set of cycles, where any agent participating in any cycle in C also participates in some cycle in C in each period $t \in [1, T]$. As each agent $i \in I$ participating in C points to some $a^t \in A^t$ in each period t , each agent participating in C also points to at least one consumption path $x \in X$ in C . For notational ease, denote $i \in c^t \in C$ by $i \in C$ and $a \in c^t \in C$ by $a \in C$. Furthermore, if $i \rightarrow x$ in C , then $x \in C$. Define \mathcal{C} as the set of all C for which

$$\forall t \in [1, T], \forall a^t \in A^t \ (a^t \in x \in C \implies \nexists x' \in C: a^t \in x'), \quad (2)$$

$$\forall t \in [1, T], \forall i \in I (i \in c^t \in C \implies \nexists (c'^t \neq c^t) \in C: i \in c'^t) \quad (3)$$

and

$$C \in \mathcal{C} \implies \nexists C' \in \mathcal{C}: C' \subset C. \quad (4)$$

The first condition ensures that the constraint that all $a^t \in A^t$ be indivisible is satisfied and that no $i \in I$ may point to more than one $a \in A$ in any period. The second condition ensures that each agent participating in C will participate in exactly one $c^t \in C$ in each period and point to exactly one consumption path $x \in X$. The third condition ensures that a cycle set cannot be divided into smaller cycle sets. A set

$$C^\mu \equiv \{ \{C \in \mathcal{C}\} \mid (\forall i \in I \exists j: (i \in C_j \in C^\mu \wedge i \notin C_{-j} \in C^\mu)) \wedge (a^t \in x \in C_j \in C^\mu \implies a^t \notin x' \in C_{-j} \in C^\mu) \} \quad (5)$$

defines a matching plan $\mu \in \mathcal{M}$, under which each agent is assigned the consumption path he or she is pointing to in C^μ . The first condition states that an agent may only participate in one $C \in C^\mu$ and the second condition states that an object may only participate in one cycle set in each period. Not all matching plans can be defined by such a set. The φ mechanism always selects a matching plan μ by selecting some C^μ .

Let I_C denote the ordered set of all agents participating in C . I_C is ordered analogously with g^1 in section 3.3. $I_C(1)$ is the agent participating in C that is of highest priority under f and $I_C(k \geq 2)$ is the agent whose endowment agent $I_C(k-1)$ points to in period 1. Denote the set of all cycle sets $C \in \mathcal{C}$ in which each participating agent $i \in I$ points to an acceptable consumption path by $\mathcal{C}_{IR} \subseteq \mathcal{C}$. In other words, \mathcal{C}_{IR} is the set of all $C \in \mathcal{C}$ that satisfy the individual rationality constraint. A cycle set $C \in \mathcal{C}$ is preferred by an agent to a cycle set $C' \in \mathcal{C}$ if and only if the agent prefers the consumption path he or she points to in C to the one he or she points to in C' . The agent is indifferent between them if he or she points to the same consumption path in both. The φ mechanism selects a matching plan by finding a set of cycle sets

$$\{C \in C^\mu \mid \forall C' \in C^\mu C \in \mathcal{C}_{IR}\}. \quad (6)$$

By the definition of C^μ , each agent $i \in I$ participates in exactly one cycle set in (6). There may be several sets matching the description of (6). To illustrate how the φ mechanism

selects a particular set, define K_i as the set containing all $C \in \mathcal{C}_{IR}$ involving agent i and K_J as the set containing all $C \in \mathcal{C}_{IR}$ involving any agent or object $j \in J \subseteq I \cup \{a^t \in A^t \mid t \in [1, T]\}$. Whenever $\forall C \in K_i, \forall i, i' \in C$ if $i' < i$, K_i is ordered such that $C \in K_i$ is of higher order than $C' \in K_i$ if $CP_{I_C(1)}C'$, or $CR_{I_C(1)}C' \wedge CP_{I_C(2)}C'$, or $CR_{I_C(1)}C' \wedge CR_{I_C(2)}C' \wedge CP_{I_C(3)}C'$ and so on. Generally, $C \in K_i$ is of higher order than $C' \in K_i$ if and only if

$$CP_{I_C(1)}C' \vee (\exists j: (CP_{I_C(j)}C' \wedge \forall j' < j CR_{I_C(j')}C')). \quad (7)$$

In other words, $C \in K_i$ is of higher order than $C' \in K_i$ if C is strictly preferred to C' by some agent $i \in I_C$ and all agents of higher order than i in I_C are indifferent between C and C' . To see that R is equivalent to indifference in statement (7), consider some j for which (7) holds and for which there exists some $j' < j$ such that $CP_{I_C(j')}C'$, then statement (7) holds for j' as well. As all agents in I_C who are indifferent between C and C' necessarily participate in C' as well, the definition is equivalent if I_C is replaced by $I_{C'}$. $K_i \setminus K_J$ is the ordered subset of K_i containing all cycles involving i but not involving any agent or object $j \in J \subseteq I \cup \{a^t \in A^t \mid t \in [1, T]\}$. Lastly, define the set of all blocked $i \in I$ and $a^t \in A^t$ as

$$B_j \equiv \{b \in I \cup \{a^t \in A^t \mid t \in [1, T]\} \mid b \in B_{j-1} \cup \{K_j \setminus K_{B_{j-1}}\}(1)\}, \quad (8)$$

where $B_0 = \emptyset$. B_1 is thus the set of agents $i \in I$ and objects $a^t \in A^t$ participating in $K_1(1)$.

Definition 1. Given some $(P, f) \in \prod_{i=1}^N \Omega \times F$, the φ mechanism is the mechanism that selects $\mu \in \mathcal{M}_{IR}$ such that

$$\begin{aligned} C^\mu = \{ & K_j \in \{\{K_j \setminus K_{B_{j-1}}\}(1) \mid \\ & (i_1 \equiv f(1)) \wedge (i_{j \geq 2} \equiv (i \in I: \forall (i' \neq i) \notin B_{j-1} \text{ if } i') \wedge (\forall (i' \neq i) \in B_{j-1} \text{ if } i))\} \\ & \mid \forall i \in I \exists j: i \in K_j\}. \end{aligned} \quad (9)$$

An algorithm that finds this matching plan is described in Section 3.3. To get an intuitive understanding of Definition 1, it can be thought of as a process in which $f(1)$ first chooses the set of all $C \in \mathcal{C}_{IR}$ that are weakly preferred to all other $C' \in \mathcal{C}_{IR}$. In all such C , $f(1)$ points to his most preferred consumption path out of all consumption paths he points to in any $C \in \mathcal{C}_{IR}$. As a result, for any such cycle sets C and C' , $I_C(2) = I_{C'}(2)$.

Agent $I_C(2)$, where C is an arbitrary element in agent $f(1)$'s choice, then chooses the subset of agent $f(1)$'s choice containing all cycle sets that are weakly preferred by $I_C(2)$ to all other cycle sets in $f(1)$'s choice. Next, agent $I_C(3)$, where C is an arbitrary element in agent $I_C(2)$'s choice, chooses the subset of agent $I_C(2)$'s choice that consists of all cycle sets that are weakly preferred by $I_C(3)$ to all other cycle sets in agent $I_C(2)$'s choice. This process continues until only $K_{f(1)}(1)$ remains, at which point the process starts over with the highest agent not participating in $K_{f(1)}(1)$, i_2 , in place of $f(1)$. At this point, i_2 is restricted to all $C \in \mathcal{C}_{IR}$ that do not involve any agent or object $j \in I \cup \{a^t \in A^t \mid t \in [1, T]\}$ participating in $K_{f(1)}(1)$. When, again, only one cycle set remains, the process starts over with agent i_3 . i_3 is defined as the agent not participating in any of the two selected cycle sets above that is of highest priority under f , and is restricted to all $C \in \mathcal{C}_{IR}$ that do not involve any of the agents or objects $j \in I \cup \{a^t \in A^t \mid t \in [1, T]\}$ participating in any of the two selected cycle sets. The whole process terminates whenever $\forall i \in I \exists j: i \in C_j$. At this point, the selected cycle sets form C^μ , and the φ mechanism selects the matching plan μ . For allocation problems without endowments, the φ mechanism reduces to the serial dictatorship futures mechanism described in Section 3.1.

3.3 The algorithm

In order to describe the algorithm that locates the matching plan selected by the φ mechanism, some further notation is needed. g^t is an ordered subset of I^t , where $g^t(j)$ denotes the j th agent in g^t . At the start of the process, $g^1 = (f(1))$. The set g^t can be partitioned into subsets g_j^t , where g^t is ordered by subset index first, such that $(g_1^t(1), g_1^t(2), \dots, g_1^t(\#g_1^t), g_2^t(1), g_2^t(2), \dots)$. Let lowercase $s_{g(j)}$ denote the ordered subset of $S_{g^t(j)}$ consisting of acceptable consumption paths that are not blocked by any agent $g^t(j')$ for whom $j' < j$ and that are not members of the set S_i^B . At the start of the algorithm, $\forall i \in I S_i^B = \emptyset$. Let ν be a temporary matching, such that $\nu_t(i)$ is an object $a^t \in A^t$ temporarily assigned to agent i at some step of the algorithm. An agent points to some $x \in X$. Through x , the agent points to each object in it, in its respective period. If $T = 2$, $s_i(1) = (a, a')$ and $i \rightarrow s_i(1)$, then this is equivalent to $((i^1 \in I^1) \rightarrow (a^1 \in A^1)) \wedge ((i^2 \in I^2) \rightarrow (a'^2 \in A^2))$. Each $(a^t \in A_A^t) \rightarrow \nu_t^{-1}(a^t)$, each $(a^t \in A_U^t) \rightarrow g_{|a^t|+1}^t(1)$ and $a_0 \rightarrow I$, i.e. a_0 points at all agents in all periods. Before the algorithm starts, μ is an empty matching, where no agent is assigned any consumption

path. The algorithm starts at step 0.

Step 0

Clear g^t for all $t \geq 1$. Generate $g_1^1 = (f(1))$ and set $\nu_1 = \lambda_1(f)$. Go to step 1.

Step 1

If

$$s_{g_{|c^1|+1}^1(1)}^1(1) = \lambda_1(g_{|c^1|+1}^1(1)), \quad (10)$$

then go to step 1.1.2.a.

If

$$s_{g_{|c^1|+1}^1(1)}^1(1) \in A_U^1, \quad (11)$$

then go to step 1.1.2.b.

Otherwise go to step 1.1.

Step 1.1

$g^1(\#g^1) \rightarrow s_{g^1(\#g^1)}(1) \rightarrow \nu_1^{-1}(s_{g^1(\#g^1)}^1(1))$. Place $\nu_1^{-1}(s_{g^1(\#g^1)}^1(1))$ at the bottom of g^1 .

If

$$s_{g^1(\#g^1)}(1) = \lambda(g^1(\#g^1)), \quad (12)$$

then go to step 1.1.1.

If

$$s_{g^1(\#g^1)}(1) \neq \lambda(g^1(\#g^1)) \wedge s_{g^1(\#g^1)}^1(1) \notin \nu_1(g_{|c^1|+1}^1(1)) \cup A_U^1, \quad (13)$$

then go to step 1.1.

If

$$s_{g^1(\#g^1)}(1) \neq \lambda(g^1(\#g^1)) \wedge s_{g^1(\#g^1)}^1(1) = \nu_1(g_{|c^1|+1}^1(1)), \quad (14)$$

then go to step 1.1.2.a.

If

$$s_{g^1(\#g^1)}(1) \neq \lambda(g^1(\#g^1)) \wedge s_{g^1(\#g^1)}^1(1) \in A_U^1, \quad (15)$$

then go to step 1.1.2.b.

Step 1.1.1

Sort $s_{g^1(\#g^{1-1})}(1)$ into $S_{g^1(\#g^{1-1})}^B$, clear $S_{g^1(\#g^1)}^B$ and remove $g^1(\#g^1)$ from g^1 . Go to step 1.

Step 1.1.2.a

A cycle $c_{|c^1|+1}^1$ is formed. Modify ν_1 such that all agents in g^1 are assigned the objects they point to and generate $g_1^2 = (g^1(1))$. Modify ν_2 such that for all $i \in f$, $\nu_2(i) = \nu_1(i)$ if $\nu_1(i) \in A^2$ and $\nu_2(i) = a_0$ otherwise. Go to step 2.

Step 1.1.2.b

A cycle $c_{|c^1|+1}^1$ is formed. Modify ν_1 such that all agents in g^1 are assigned the objects they point to and such that $\nu_1(g_{|c^1|}^1(1)) \in A_U^1$ if $\nu_1(g_{|c^1|}^1(1)) \neq a_0$. Generate $g_1^2 = (g^1(1))$ and modify ν_2 such that for all $i \in f$, $\nu_2(i) = \nu_1(i)$ if $\nu_1(i) \in A^2$ and $\nu_2(i) = a_0$ otherwise. Go to step 2.

Step $t \geq 2$

If

$$s_{g_{|c^t|+1}^t}(1) = \nu_t(g_{|c^t|+1}^t(1)), \quad (16)$$

then go to step $t.1.4.a$.

If

$$s_{g_{|c^t|+1}^t}(1) \in A_U^t, \quad (17)$$

then go to step $t.1.4.b$.

Otherwise go to step $t.1$.

Step $t.1$

$g^t(\#g^t) \rightarrow s_{g^t(\#g^t)}(1) \rightarrow \nu_t^{-1}(s_{g^t(\#g^t)}^t(1))$ and place $\nu_t^{-1}(s_{g^t(\#g^t)}^t(1))$ at the bottom of g^t .

If

$$g^t(\#g^t) \notin g^1 \wedge s_{g^t(\#g^t)}(1) = \lambda(g^t(\#g^t)), \quad (18)$$

then go to step $t.1.1$.

If

$$g^t(\#g^t) \notin g^1 \wedge s_{g^t(\#g^t)}(1) \neq \lambda(g^t(\#g^t)), \quad (19)$$

then go to step $t.1.2$.

If

$$g^t(\#g^t) \in g^1 \wedge s_{g^t(\#g^t)}(1) \neq \lambda(g^t(\#g^t)) \wedge s_{g^t(\#g^t)}^t(1) \notin \nu_t(g_{|c^t|+1}^t(1)) \cup A_U^t, \quad (20)$$

then go to step $t.1$.

If

$$g^t(\#g^t) \in g^1 \wedge s_{g^t(\#g^t)}(1) \neq \lambda(g^t(\#g^t)) \wedge s_{g^t(\#g^t)}^t(1) = \nu_t(g_{|c^t|+1}^t(1)), \quad (21)$$

then go to step $t.1.4.a$.

If

$$g^t(\#g^t) \in g^1 \wedge s_{g^t(\#g^t)}(1) \neq \lambda(g^t(\#g^t)) \wedge s_{g^t(\#g^t)}^t(1) \in A_U^t, \quad (22)$$

then go to step $t.1.4.b$.

Step $t.1.1$

Sort $s_{g^t(\#g^t-1)}(1)$ into $S_{g^t(\#g^t-1)}^B$ and clear $S_{g^t(\#g^t)}^B$. Go to step 0.

Step $t.1.2$

Generate $g_{|c^t|+1}^1 = (g^t(\#g^t))$. Go to step 1.

Step $t.1.4.a$

A cycle $c_{|c^t|+1}^t$ is formed. Modify ν_t such that all agents in g^t are assigned the objects they point to. Go to step $t.1.5$.

Step $t.1.4.b$

A cycle $c_{|c^t|+1}^t$ is formed. Modify ν_t such that all agents in g^t are assigned the objects they point to and such that $\nu_t(g_{|c^t|}^t(1)) \in A_U^t$. Go to step $t.1.5$.

Step $t.1.5$

If

$$\#g^t < \#g^1, \quad (23)$$

then go to step $t.1.5.a$.

If

$$\#g^t = \#g^1, \quad (24)$$

then go to step $t.1.5.b$.

Step $t.1.5.a$

Generate $g_{|c^t|+1}^t = (\{g^1 \setminus g^t\}(1))$. Go to step t .

Step $t.1.5.b$

If

$$t < T, \quad (25)$$

then modify ν_{t+1} such that for all $i \in f$, $\nu_{t+1}(i) = \nu_t(i)$ if $\nu_t(i) \in A^{t+1}$ and $\nu_{t+1}(i) = a_0$ otherwise. Go to step $t + 1$.

If

$$t = T, \quad (26)$$

then go to step T.2.

Step T.2

Modify μ such that all agents in g^1 are assigned the consumption paths $x \in X$ they are pointing to. Remove all $\{a^t \in A^t \mid \exists i \in g^1: i \rightarrow a^t\}$ from A^t and sort all strategies involving at least one such a^t into S_i^B for all $i \in f$. Remove all $i \in g^1$ from f . Go to step 0.

The process ends when all agents in I have been assigned some $x \in X$ under μ . I.e., the process ends when $\mu \in \mathcal{M}$.

3.4 Example

Consider an allocation problem $\langle T, I, A, P, \lambda \rangle$ where $T = 2$, $I = \{i_1, i_2, \dots, i_6\}$, $A_A^1 = A_A^2 = \{a_1, a_2, \dots, a_5\}$, $A_U^1 = A_U^2 = \{a_6\}$, $\lambda(i_j) = a_j$ for $j = 1, 2, \dots, 5$, $\lambda(i_6) = a_0$ and P is given by

P_1	P_2	P_3	P_4	P_5	P_6
(a_4, a_2)	(a_4, a_4)	(a_4, a_4)	(a_1, a_6)	(a_5, a_3)	\dots
\dots	(a_1, a_1)	(a_5, a_5)	\dots	(a_3, a_5)	
	(a_2, a_1)	(a_5, a_4)		(a_5, a_5)	
	\dots	\dots		\dots	

Furthermore, let $f = (i_1, i_2, i_3, i_4, i_5, i_6)$.

Step 0

$g_1^1 = i_1$ is generated and ν_1 is modified such that $\nu_1 = \lambda_1$. Go to step 1.

Step 1

$|c^1| = 0$ and $s_{g_1^1(c^1+1)}(1) = s_1(1) = (a_4, a_2)$. $s_{g_1^1(c^1+1)}(1) = a_4 \neq \lambda_1(i_1) = a_1$ and $a_4 \notin A_U^1$. Go to step 1.1.

Step 1.1

$(g^1(1) = a_1) \rightarrow (s_1^1(1) = a_4) \rightarrow (\nu_1^{-1}(a_4) = i_4)$. i_4 is placed at the bottom of g^1 . Now $s_{g^1(\#g^1)}(1) = s_4(1) = (a_1, a_3)$. $((a_1, a_3) \neq \lambda(i_4)) \wedge (s_4^1(1) = a_1) = (\nu_1(g_1^1(1)) = \nu_1(i_1) = a_1)$. Go to step 1.1.2.a.

Step 1.1.2.a

A cycle c_1^1 is formed. ν_1 is modified such that $\nu_1(i_1) = a_4$ and $\nu_1(i_4) = a_1$. ν_2 is set such that $\nu_2 = \nu_1$ and $g_1^2 = (i_1)$ is generated. Go to step 2.

Step 2

$s_{g_{|c_1^2|+1}^2}(1) = s_1(1) = (a_4, a_2)$. $(s_1^2(1) = a_2) \neq (\nu_2(g_1^2(1)) = a_4)$ and $a_2 \notin A_U^t$. Go to step 2.1.

Step 2.1

$i_1 \rightarrow (a_4, a_2) \rightarrow (\nu_2^{-1}(a_2) = i_2)$ and i_2 is placed at the bottom of g^t . Note that a_4^1 and a_1^1 are pointed to by i_1 and i_4 respectively. Both (a_4, a_4) and (a_1, a_1) are thus blocked strategies for agent i_2 , giving $s_2(1) = (a_2, a_1)$. As $i_2 \notin g^1$ and $s_2(1) = (a_2, a_1) \neq \lambda_{i_2}$, go to step 2.1.2.

Step 2.1.2

$g_2^1 = (i_2)$ is generated. Go to step 1.

Step 1

$s_2^1(1) = a_2 = \lambda_1(i_2)$. Go to step 1.1.2.a.

Step 1.1.2.a

A cycle c_2^1 is formed, involving only i_2 and a_2 . ν_1 is modified such that $\nu_1(i_2) = a_2$. Then ν_2 is set such that $\nu_2 = \nu_1$ and $g_1^2 = (i_1)$ is generated, replacing the existing g_1^2 . Go to step 2, which is repeated exactly as above.

Step 2.1

$i_1 \rightarrow (a_4, a_2) \rightarrow i_2$ and i_2 is placed at the bottom of g^2 , just like before. However, now $i_2 \in g^1$ holds, $s_2(1) \neq \lambda(i_2)$ and $s_2^2(1) = a_1 \notin (\nu_2(i_1) = a_4) \cup A_U^2$. Go to step 2.1.

Step 2.1

$i_2 \rightarrow (a_2, a_1) \rightarrow i_4$ and i_4 is placed at the bottom of g^2 . As $i_4 \in g^1$, $s_4(1) \neq \lambda(i_4)$ and $s_4^2(1) = a_6 \in A_U^2$, go to step 2.1.4.b.

Step 2.1.4.b

A cycle c_1^2 is formed. ν_t is modified such that $\nu_2(i_1) = a_2$, $\nu_2(i_2) = a_1$ and $\nu_2(i_4) = a_6$ and such that $a_4 \in A_U^2$. Go to step 2.1.5.

Step 2.1.5

$\#g^2 = \#g^1 = 3$. Go to step 2.1.5.b.

Step 2.1.5.b

$T = 2$. Go to step T.2.

Step T.2

$\mu(i_1) = (a_4, a_2)$, $\mu(i_2) = (a_2, a_1)$ and $\mu(i_4) = (a_1, a_6)$. $\{a_1, a_2, a_4\}$ are removed from A^1 , $\{a_1, a_2, a_6\}$ are removed from A^2 and $\{i_1, i_2, i_4\}$ are removed from f . Go to step 0.

The reader can confirm that a cycle set $C \in \mathcal{C}_{IR}$ has formed at step T.2. At this point, $A_A^1 = A_A^2 = \{a_3, a_5\}$, $A_U^1 = \{a_6\}$, $A_U^2 = \{a_4\}$ and $f = (i_3, i_5, i_6)$.

Step 0

Both g^1 and g^2 are cleared. $g_1^1 = (i_3)$ is generated and ν_1 is modified such that $\nu_1 = \lambda_1(f)$. This implies that $\nu_1(i_3) = a_3$, $\nu_1(i_4) = a_4$ and $\nu_1(i_6) = a_0$. Go to step 1.

Step 1

As $(a_4, a_4) \in S_3^B$, $s_3(1) = (a_5, a_5)$. $s_3^1(1) = a_5 \neq \lambda_1(i_3) = a_3$ and $s_3^1(1) \notin A_U^1$. Go to step 1.1.

Step 1.1

$i_3 \rightarrow (a_5, a_5) \rightarrow i_5$ and i_5 is placed at the bottom of g^1 . As $i_3 \rightarrow (a_5, a_5)$, both (a_5, a_3) and (a_3, a_5) are blocked and $s_5(1) = (a_5, a_5) = \lambda(i_5)$. Go to step 1.1.1.

Step 1.1.1

(a_5, a_5) is sorted into S_3^B and i_5 is removed from g^1 . Go to step 1.

Step 1

As $(a_5, a_5) \in S_3^B$, $s_3(1) = (a_5, a_4)$. $a_5 \neq \lambda_1(i_3) = a_3$ and $a_5 \notin A_U^1$. Go to step 1.1.

Step 1.1

$i_3 \rightarrow (a_5, a_4) \rightarrow i_5$ and i_5 is placed at the bottom of g^1 . $s_5(1) = (a_3, a_5)$, as (a_5, a_3) is blocked by i_3 . $(a_3, a_5) \neq \lambda(i_5)$ and $a_3 = \nu_1(i_3)$. Go to step 1.1.2.a.

Step 1.1.2.a

A cycle c_1^1 is formed. ν_1 is modified such that $\nu_1(i_3) = a_5$ and $\nu_1(i_5) = a_3$ and $g_1^2 = (i_3)$ is generated. ν_2 is modified such $\nu_2(i_5) = a_3$ and $\nu_2(i_3) = a_5$. Go to step 2.

Step 2

$s_3(1) = (a_5, a_4)$ and $a_4 \in A_U^t$. Go to step 2.1.4.b.

Step 2.1.4.b

A cycle c_1^2 is formed. ν_2 is modified such that $\nu_2(i_3) = a_4$ and such that $a_5 \in A_U^2$. Go to step 2.1.5.

Step 2.1.5

$\#g^2 = 1 < \#g^1 = 2$. Go to step 2.1.5.a.

Step 2.1.5.a

$g_2^2 = (\{g^1 \setminus g^2\}(1)) = (i_5)$ is generated. Go to step 2.

Step 2

$s_5(1) = (a_3, a_5)$ and $a_5 \in A_U^2$. Go to step 2.1.4.b

Step 2.1.4.b

A cycle c_2^2 is formed. ν_2 is modified such that $\nu_2(i_5) = a_5$ and such that $a_3 \in A_U^2$. Go to step 2.1.5.

Step 2.1.5

$\#g^2 = \#g^1 = 2$. Go to step 2.1.5.b.

Step 2.1.5.b

$T = 2$. Go to step T.2.

Step T.2

μ is modified such that $\mu(i_3) = (a_5, a_4)$ and $\mu(i_5) = (a_3, a_5)$. The remaining $a \in A_A^t$ are removed such that $A_A^t = \emptyset$ for $t = 1, 2$ and $\{i_3, i_5\}$ are removed from f . Go to step 0.

At this point, $f = (i_6)$, $A_U^1 = a_6$ and $A_U^2 = a_3$. It is easy to see that i_6 will be assigned (a_6, a_3) , yielding the final matching plan μ , where $\mu(i_1) = (a_4, a_2)$, $\mu(i_2) = (a_2, a_1)$, $\mu(i_3) = (a_5, a_4)$, $\mu(i_4) = (a_1, a_6)$, $\mu(i_5) = (a_3, a_5)$ and $\mu(i_6) = (a_6, a_3)$. Note that the algorithm locates three cycle sets $C \in C^\mu$.

4 Properties

As mentioned in the introduction, mechanisms are often evaluated by examining which properties they satisfy. This section will investigate whether the φ mechanism satisfies individual rationality, Pareto efficiency, non-bossiness and strategy-proofness. Proposition 1 states that the φ mechanism is individually rational, Proposition 2 states that it is Pareto efficient and Proposition 3 states that it is non-bossy. Furthermore, Proposition 4 states that the φ mechanism is not strategy proof and Proposition 5 presents the conditions under which it is manipulable.

To examine the properties of the φ mechanism, some additional concepts are helpful. Recall the definition of B_j and amend it by adding the underlined $\cup i$, such that

$$B'_j \equiv \{b \in I \cup \{a^t \in A^t \mid t \in [1, T]\} \mid b \in B'_{j-1} \cup \{K_j \setminus K_{B'_{j-1} \cup i}\}(1)\}, \quad (27)$$

where $B'_0 = \emptyset$ and consequently $K_{B'_0} = \emptyset$. Let the agent i 's *consumption set* be defined by

the ordered set

$$\begin{aligned}
X_i \equiv \{x \in C \in & \\
& \{ \{ \{K_j \setminus K_{B'_{j-1}}\}(1), \{K_j \setminus K_{B'_{j-1}}\}(2), \dots, \{K_j \setminus K_{B'_{j-1}}\}(k_j) \mid \\
& \quad (\{K_j \setminus K_{B'_{j-1}}\}(k_j + 1) = \{K_j \setminus K_{B'_{j-1} \cup i}\}(1)) \wedge \\
(i_1 \equiv f(1)) \wedge (i_{j \geq 2} \equiv (i_j \in I) \notin B'_{j-1}: \forall (i' \in I) \notin B'_{j-1} \ i_j f i') \} \mid i_j f i \} \cup & \\
& \{ \{K_i \setminus K_{B'_j}\}(1), \{K_i \setminus K_{B'_j}\}(2), \dots, \{K_i \setminus K_{B'_j}\}(k) \mid \\
& \quad (\forall (i' \in I) \notin B'_j \ i f i') \wedge (i \rightarrow \lambda(i) \text{ in } \{K_i \setminus K_{B'_j}\}(k)) \} \\
& \mid i \rightarrow x \text{ in } C \}. \quad (28)
\end{aligned}$$

Any consumption path $(x \neq \lambda(i)) \in X_i$ can be rejected by reporting preferences under which x is not acceptable. X_i is ordered such that $x \in \{K_j \setminus K_{B'_{j-1}}\}(k)$ is of higher order than $x' \in \{K_{j'} \setminus K_{B'_{j'-1}}\}(k')$ if and only if $j < j'$ or $(j = j') \wedge (k < k')$. To phrase it differently, the consumption path $x \in X_i$ is of higher order than $x' \in X_i$ under X_i if there exists some $C \in \{C_j \mid (i \in C_j) \wedge (i \rightarrow x)\}$ that the algorithm processes before all $C \in \{C_j \mid (i \in C_j) \wedge (i \rightarrow x')\}$. Hence, $X_i(j)$ is the j th consumption path that i has the opportunity of rejecting. It is always true that $\varphi_i(P, f) = X_i(1)$. X_i will be deconstructed and explained in the proof of Lemma 1, which states that X_i defines the full set of consumption paths attainable by i using some strategy $S_i \in \Omega$.

Lemma 1. $\forall i \in I, \forall (P, f) \in \prod_{i=1}^N \Omega \times F \ X_i = \{\varphi_i(\{S_i, P_{I \setminus i}\}, f) \mid S_i \in \Omega\}$.

Proof. For this proof, the reader will have to refer to section 3.3 to confirm the order in which cycle sets are processed by the algorithm. Consider some priority structure $f \in F$ and some agent $i \in I$ who reports some preference relation $P'_i \in \Omega$ such that $\forall x \in X \ \lambda(i) R'_i x$ and $P'_i \neq P_i$, where P_i is agent i 's true preference relation and R'_i denotes weak preference under P'_i . The algorithm will start by processing the cycle set $K_{f(1)}(1)$. If $i \in K_{f(1)}(1)$, then $K_{f(1)}(1)$ will be blocked by i and the algorithm will continue by processing $K_{f(1)}(2)$, $K_{f(1)}(3)$ and so on until $i \notin K_{f(1)}(k+1)$, as defined in the definition of X_i . While it does not invalidate this proof, it should be noted that the algorithm technically does not process cycle sets one by one. Rather, it processes subsets of $K_{f(1)}$ such that if $i = g^1(j)$ at some step of the algorithm and agent $g^1(j-1)$ is assigned the same consumption path under $K_{f(1)}(1), K_{f(1)}(2) \dots K_{f(1)}(k)$, then all cycle sets in $\{K_{f(1)}(1), \dots, K_{f(1)}(k)\}$ are

skipped simultaneously if blocked by i . The algorithm will thus process all elements in

$$\{K_1(1), K_1(2), \dots, K_1(k) \mid K_1(k+1) = \{K_1 \setminus K_i\}(1) \wedge i_1 \equiv f(1) \neq i\}. \quad (29)$$

If $K_1(1) = \{K_1 \setminus K_i\}(1)$, then the above set is empty. At this point, the algorithm will assign all agents in $\{K_1 \setminus K_i\}(1)$ the consumption paths they point to. All such agents are removed from the process along with the objects $a^t \in A^t$ they point to, thereby blocking all cycle sets involving some agent or object in

$$B'_1 \equiv \{b \in I \cup \{a^t \in A^t \mid t \in [1, T]\} \mid b \in \{K_1 \setminus K_i\}(1)\}. \quad (30)$$

As all agents in $\{K_1 \setminus K_i\}(1)$ have been removed from the process, the algorithm continues by processing $\{K_2 \setminus K_{B'_1}\}(1)$. As this cycle is also blocked by agent i , the algorithm will continue by processing $\{K_2 \setminus K_{B'_1}\}(2)$ and so on until the first cycle set in $K_2 \setminus K_{B'_1}$ that does not involve agent i , $\{K_2 \setminus K_{B'_1 \cup i}\}(1)$, is reached. At this point, all agents in $\{K_2 \setminus K_{B'_1 \cup i}\}(1)$ are assigned the consumption paths they point to and are removed from the process along with the objects $a^t \in A^t$ they point to, thereby blocking all cycle sets involving some agent or object in

$$B'_2 \equiv \{b \in I \cup \{a^t \in A^t \mid t \in [1, T]\} \mid b \in B'_1 \cup \{K_2 \setminus K_{B'_1 \cup i}\}(1)\}. \quad (31)$$

Generally, this process continues for the sets K_3, K_4 and so on. The algorithm has thus been shown to process all elements in

$$\begin{aligned} & \{ \{K_j \setminus K_{B'_{j-1}}\}(1), \{K_j \setminus K_{B'_{j-1}}\}(2), \dots, \{K_j \setminus K_{B'_{j-1}}\}(k_j) \mid \\ & \quad (\{K_j \setminus K_{B'_{j-1}}\}(k_j + 1) = \{K_j \setminus K_{B'_{j-1} \cup i}\}(1)) \wedge \\ & \quad (i_1 \equiv f(1)) \wedge (i_{j \geq 2} \equiv (i_j \in I) \notin B'_{j-1}: \forall (i' \in I) \notin B'_{j-1} \ i_j f i') \} \mid i_j f i \}. \quad (32) \end{aligned}$$

When the algorithm arrives at a point when the condition $i_j f i$ no longer holds, then $\forall i' \notin B'_{j-1} \ i f i'$. In other words, agent i is the agent with highest priority under f who is still in the process. The algorithm then proceeds by processing $\{K_i \setminus K_{B_{j-1}}\}(1)$, $\{K_i \setminus K_{B_{j-1}}\}(2)$ and so on, until it reaches some cycle set in $K_i \setminus K_{B_{j-1}}$ in which $i \rightarrow \lambda(i)$. Note that the cycle set involving only agent i and his or her endowment is always a member of $K_i \setminus K_{B_{j-1}}$ as $i \notin B_{j-1}$ and if $\lambda^t(i) \in B_{j-1}$ for any $t \in [1, T]$, agent i would also be a member of B_{j-1} .

The endowment $\lambda(i)$ can never be rejected by agent i , and i is then finally assigned $\lambda(i)$. This covers all elements in

$$\{ \{K_i \setminus K_{B'_j}\}(1), \{K_i \setminus K_{B'_j}\}(2), \dots, \{K_i \setminus K_{B'_j}\}(k) \mid \forall (i' \in I) \notin B'_j \ i f i' \wedge (i \rightarrow \lambda(i) \text{ in } \{K_i \setminus K_{B'_j}\}(k)) \}. \quad (33)$$

It has thus been shown that if some agent $i \in I$ blocks all cycles he or she is able to, the algorithm will process all elements in X_i . X_i is then defined as the set of consumption paths agent i points to in each cycle set in the union of the sets (32) and (33). Consider an arbitrary element in X_i , $X_i(j) \neq \lambda(i)$, and a strategy $P''_i \in \Omega$, where

$$\forall x' \in X \setminus X_i(j) \ X_i(j) P''_i \lambda(i) R''_i x'. \quad (34)$$

Under this strategy, agent i will be assigned $X_i(j)$. Hence,

$$\forall j \leq \#X_i \ \exists S_i \in \Omega: \varphi_i(\{S_i, P_{I \setminus i}\}, f) = X_i(j). \quad (35)$$

Next, consider an arbitrary subset $X'_i \subseteq \{X_i \setminus \lambda(i)\}$ and a strategy $P''_i \in \Omega$, where

$$\forall x' \in X'_i, \forall x \in X \setminus X'_i \ x' P''_i \lambda(i) R''_i x. \quad (36)$$

Then,

$$X'_i \neq \emptyset \implies \varphi_i(\{P''_i, P_{I \setminus i}\}, f) = X'_i(1) \quad (37)$$

and

$$X'_i = \emptyset \implies \varphi_i(\{P''_i, P_{I \setminus i}\}, f) = \lambda(i). \quad (38)$$

This covers the entire strategy space Ω . Consequently,

$$\forall S_i \in \Omega \ \varphi_i(\{S_i, P_{I \setminus i}\}, f) \in X_i. \quad (39)$$

Statements (35) and (39) imply Lemma 1. □

No $x \notin X_i$ is attainable using any strategy $S_i \in \Omega$, and all $x \in X_i$ are attainable using some strategy $S_i \in \Omega$. Note that whenever $i = f(1)$, (32) is an empty set. While this implies that $X_{f(1)}$ can be defined by a shorter expression it does not imply a smaller

consumption set for agent $f(1)$. As there are no cycle sets that are blocked by some agent of higher priority, agent $f(1)$'s consumption set is given by all consumption paths he or she points to under any $C \in \mathcal{C}_{IR}$.

4.1 Positive results

Definition 2. A matching plan $\mu \in \mathcal{M}$ is individually rational if and only if $\mu(i)$ is acceptable to i , for all $i \in I$.

Define $\mathcal{M}_{IR} \subseteq \mathcal{M}$ as the set of all individually rational matching plans.

Definition 3. A direct mechanism Γ is individually rational if and only if

$$\forall (S, f) \in \prod_{i=1}^N \Omega \times F \quad \Gamma(S, f) \in \mathcal{M}_{IR}. \quad (40)$$

In other words, a mechanism is individually rational if it always selects individually rational matching plans. Individual rationality is considered a desirable property as it ensures that no agent is ever made worse off by the mechanism. If the mechanism is individually rational, there is no incentive for any agent not to participate in the economy.

Proposition 1. The φ mechanism is individually rational.

Proof. Trivial. Agents are only assigned consumption paths that they point to. By construction, agents only point to acceptable consumption paths. \square

Definition 4. A matching plan μ is Pareto efficient if and only if

$$\nexists \mu' \in \mathcal{M} : \mu' R_i \mu \quad \forall i \in I \text{ and } \mu' P_i \mu \text{ for some } i \in I. \quad (41)$$

That is, a matching plan is Pareto efficient if no agent can be made better off without making some agent worse off.

Definition 5. A direct mechanism Γ is Pareto efficient if the matching plan $\Gamma(P, f)$ is Pareto efficient for all $(P, f) \in \prod_{i=1}^N \Omega \times F$.

Pareto efficiency is clearly considered a desirable property by mechanism designers interested in the utility of the agents.

Proposition 2. *The mechanism φ is Pareto efficient.*

Proof. Consider some arbitrary $(P, f) \in \prod_{i=1}^N \Omega \times F$ and a new ordered set h , containing all agents in I . h is partitioned into $h_1, h_2, \dots, h_{\#C^\mu}$, where $\#C^\mu$ is the number of cycle sets in C^μ . h_1 is equivalent to g^1 at step $T.2$, when the first $C \in C^\mu$ forms in the algorithm. h_j is equivalent to g^1 at step $T.2$, when the j 'th $C \in C^\mu$ forms in the algorithm. The elements in h are ordered by subindex first, such that $h = \{h_1(1), h_1(2), \dots, h_1(\#h_1), h_2(1), h_2(2), \dots, h_{\#C^\mu}(\#h_{\#C^\mu})\}$. Under $\varphi(P, f)$, agent $h(1) \in I$ is assigned his or her most preferred consumption path $x \in X$ from $\{x \mid x \in \mu \in \mathcal{M}_{IR}\}$. Denote this x by x_1 . Then for some $\mu' \in \mathcal{M}$,

$$\mu' P_{h(1)} \varphi(P, f) \implies \mu' \notin \mathcal{M}_{IR} \iff \exists i \in I: \lambda P_i \mu'. \quad (42)$$

By transitivity and Proposition 1,

$$\exists i \in I: \lambda P_i \mu' \implies \exists i \in I: \varphi(P, f) P_i \mu'. \quad (43)$$

Hence,

$$\mu' P_{h(1)} \varphi(P, f) \implies \exists i \in I: \varphi(P, f) P_i \mu' \quad (44)$$

and

$$\nexists \mu' P_{h(1)} \varphi(P, f): \mu' R_i \varphi(P, f) \quad \forall i \in I. \quad (45)$$

Agent $h(1)$ can only be made better off by making some other agent worse off.

Under $\varphi(P, f)$, agent $h(2)$ is assigned his or her most preferred consumption path from

$$\{x \mid x \in \mu \in \mathcal{M}_{IR} \wedge \mu(h(1)) = x_1\}. \quad (46)$$

Denote this x by x_2 . Then for some $\mu' \in \mathcal{M}$,

$$\mu' P_{h(2)} \varphi(f, P) \implies (\mu' \notin \mathcal{M}_{IR} \vee \mu'(h(1)) \neq x_1). \quad (47)$$

By statements (42) and (43),

$$\mu' \notin \mathcal{M}_{IR} \implies \exists i \in I: \varphi(f, P) P_i \mu'. \quad (48)$$

Since all preference relations over X are strict,

$$\mu'(h(1)) \neq x_1 \implies (\mu' P_{h(1)} \varphi(P, f) \vee \varphi(P, f) P_{h(1)} \mu'). \quad (49)$$

By statement (44),

$$\mu' P_{h(1)} \varphi(P, f) \implies \exists i \in I : \varphi(f, P) P_i \mu'. \quad (50)$$

As $h(1) \in I$,

$$\varphi(P, f) P_{h(1)} \mu' \implies \exists i \in I : \varphi(f, P) P_i \mu'. \quad (51)$$

By statements (47) through (51),

$$\mu' P_{h(2)} \varphi(f, P) \implies \exists i \in I : \varphi(f, P) P_i \mu' \quad (52)$$

and

$$\nexists \mu' P_{h(2)} \varphi(f, P) : \mu' R_i \varphi(f, P) \quad \forall i \in I. \quad (53)$$

Agent $h(2)$ can only be made better off by making some other agent worse off.

Consider some arbitrary $j > 2$. Agent $h(j)$ is assigned his or her most preferred consumption path from

$$\{x \mid (x \in \mu \in \mathcal{M}_{IR}) \wedge (\forall i \in [1, j-1] \mu(h(i)) = x_i)\}. \quad (54)$$

Denote this consumption path by x_j . Note that this recursively defines x_i for $i > 2$ as well.

Then for some $\mu' \in \mathcal{M}$,

$$\mu' P_{h(j)} \varphi(P, f) \implies (\mu' \notin \mathcal{M}_{IR} \vee \exists i < j : \mu'(h(i)) \neq x_i). \quad (55)$$

As all preference relations on X are strict,

$$\exists i < j : \mu'(h(i)) \neq x_i \implies (\mu' P_{h(i)} \varphi(P, f) \vee \varphi(P, f) P_{h(i)} \mu'). \quad (56)$$

By the general version of statement (51):

$$\varphi(P, f) P_{h(i)} \mu' \implies \exists i' \in I : \varphi(P, f) P_{i'} \mu' \quad (57)$$

and by statement (48),

$$(\mu' \notin \mathcal{M}_{IR} \vee \varphi(P, f)P_{h(i)}\mu') \implies \exists i' \in I : \varphi(P, f)P_{i'}\mu'. \quad (58)$$

The only outcome not yet addressed is $\mu'P_{h(i)}\varphi(P, f)$, for which statement (55) can be reapplied. By statements (55), (56) and (58),

$$\mu'P_{h(j)}\varphi(P, f) \implies (\exists i \in I : \varphi(P, f)P_i\mu') \vee (\mu'P_{h(j')} \varphi(P, f) \text{ for some } j' < j). \quad (59)$$

Statement (59) can be reapplied until $(\exists i \in I : \varphi(P, f)P_i\mu') \vee (j' = 1)$ is true. If $j' = 1$, then statement (44) is applicable. Thus, by statements (44) and (59),

$$\forall j \in I, \forall f \in F: \mu'P_{h(j)}\varphi(P, f) \implies \exists i' \in I : \varphi(P, f)P_{i'}\mu'. \quad (60)$$

No agent can be made better off without making some other agent worse off. Hence, for any $(P, f) \in \prod_{i=1}^N \Omega \times F$,

$$\nexists \mu' \in \mathcal{M}: \mu'R_i\varphi(P, f) \forall i \in I \text{ and } \mu'P_i\varphi(P, f) \text{ for some } i \in I. \quad (61)$$

□

Definition 6. *A mechanism Γ is non-bossy if and only if*

$$\forall S_i \in \Omega, \forall (P, f) \in \prod_{i=1}^N \Omega \times F \quad \Gamma_i(\{S_i, P_{I \setminus i}\}, f) = \Gamma_i(P, f) \implies \Gamma(\{S_i, P_{I \setminus i}\}, f) = \Gamma(P, f). \quad (62)$$

In other words, a mechanism is non-bossy if no agent can affect any other agent's assignment without affecting his or her own assignment as well. The concept of non-bossiness is due to Satterthwaite and Sonnenschein (1981).

Proposition 3. *The φ mechanism is non-bossy.*

Proof. Consider some $i \in I$, $(P, f) \in \prod_{i=1}^N \Omega \times F$ and some C^μ , defining $\mu = \varphi(P, f) \in \mathcal{M}_{IR}$. By definition, $\varphi_{i'}(P, f) \in C_j$ for some $C_j \in C^\mu$ for each $i' \in I$. Under C^μ , $\forall i' \in I \quad i' \rightarrow \mu(i')$. As f is given, C is fully determined by which $x \in X$ each $i' \in C$ points to. If $i \in C \in C^\mu$, there are three alternatives. Either $\#\{i' \in I \mid i' \in C\} = 1$, ($\#\{i' \in I \mid i' \in C\} \geq$

2) $\wedge (\forall(i' \neq i) \in C \text{ } ifi')$ or $(\#\{i \in I \mid i \in C\} \geq 2) \wedge (\exists i' \in C: i'fi)$. Recall that K_i is ordered such that

$$\{x \in \{K_i \setminus K_{B_j}\}(k) \mid i \rightarrow x\} P_i \{x' \in \{K_i \setminus K_{B_j}\}(k') \mid i \rightarrow x'\} \implies k < k', \quad (63)$$

where both sets only have one element each.

Case 1.

If $(\#\{i' \in I \mid i' \in C\} = 1) \vee (\#\{i' \in I \mid i' \in C\} \geq 2 \wedge \forall i' \in C \text{ } ifi')$ under C^μ , then $C = \{K_i \setminus K_{B_j}\}(1) \in C^\mu$, where $\forall i' \notin B_j \text{ } ifi'$ and $\forall i' \in B_j \text{ } i'fi$. Thus,

$$\begin{aligned} \varphi_i(P, f) &= \{x \in \{K_i \setminus K_{B_j}\}(1) \mid ((i' \neq i) \notin B_j \iff ifi') \wedge (i \rightarrow x)\} \\ &\implies \forall S_i \in \{S_i \in \Omega \mid \varphi_i(\{S_i, P_{I \setminus i}\}, f) = \varphi_i(P, f)\} \\ \varphi_i(\{S_i, P_{I \setminus i}\}, f) &= \{x \in \{K_i \setminus K_{B_j}\}(1) \mid ((i' \neq i) \notin B_j \iff ifi') \wedge (i \rightarrow x)\} \\ &\implies \forall S_i \in \{S_i \in \Omega \mid \varphi_i(\{S_i, P_{I \setminus i}\}, f) = \varphi_i(P, f)\} \text{ } i \rightarrow \mu(i) \in \{K_i \setminus K_{B_j}\}(1) \in C^\mu \\ &\implies \forall S_i \in \{S_i \in \Omega \mid \varphi_i(\{S_i, P_{I \setminus i}\}, f) = \varphi_i(P, f)\} \text{ } i \in C \in C^\mu \\ &\implies \forall S_i \in \{S_i \in \Omega \mid \varphi_i(\{S_i, P_{I \setminus i}\}, f) = \varphi_i(P, f)\}, \forall i' \in I \text{ } i' \in C_j, \text{ for some } C_j \in C^\mu. \end{aligned} \quad (64)$$

The second implication follows from the fact that all agents are assigned the consumption paths they point to in C^μ . The third implication follows from the fact that for a given $f \in F$, each C is defined by which $x \in X$ each $i \in C$ points to. The fourth implication follows from the fact that if no agent changes which consumption path he or she points to, the same cycle sets in C^μ will form. By the definition of C^μ and the fact that if all agents participate in some cycle set in C^μ under $S \in \prod_{i=1}^N \Omega$, then $\varphi(S, f) = \mu$,

$$\forall S_i \in \Omega \text{ } ((\varphi_i(\{S_i, P_{I \setminus i}\}, f) = \varphi_i(P, f)) \implies (\varphi(\{S_i, P_{I \setminus i}\}, f) = \mu = \varphi(P, f))). \quad (65)$$

This proves non-bossiness for case 1.

Case 2.

If $(\#\{i' \in I \mid i' \in C\} \geq 2) \wedge (\exists i' \in C: i'fi)$ under C^μ , then i participates in the cycle set

$$\{C = \{K_j \setminus K_{B_{j-1}}\}(1) \in C^\mu \mid (j \geq 1) \wedge (i \in C)\} \quad (66)$$

whenever it exists. If it does not exist, then $\#\{i \in I \mid i \in C\} \geq 2 \wedge \exists i' \in C: i'fi$ is a false statement and case 1 applies. Note that

$$\begin{aligned} \{C = \{K_j \setminus K_{B_{j-1}}\}(1) \in C^\mu \mid (j \geq 1) \wedge (i \in C)\} \neq \emptyset &\implies \\ \{C = \{K_{j'} \setminus K_{B_{j'-1}}\}(1) \in C^\mu \mid (j' > j) \wedge (i \in C)\} &= \emptyset, \end{aligned} \quad (67)$$

as $i \in B_{j'-1}$. Whenever a cycle set $C \in C^\mu$ as defined by statement (66) forms under $\varphi(P, f)$, it also forms under $\varphi(\{P'_i, P_{I \setminus i}\}, f)$ if and only if $\mu(i)P'_i\lambda(i)$.

$$\forall S_i \in \{P'_i \in \Omega \mid \mu(i)P'_i\lambda(i)\} \quad i \in C \in C^\mu \implies \varphi(\{S_i, P_{I \setminus i}\}, f) = \mu = \varphi(P, f). \quad (68)$$

By Proposition 1,

$$\begin{aligned} \forall S_i \in \{P'_i \in \Omega \mid \lambda(i)P'_i\mu(i)\} \quad \varphi_i(\{S_i, P_{I \setminus i}\}, f) \neq \mu(i) \\ \implies \varphi(\{S_i, P_{I \setminus i}\}, f) \neq \mu = \varphi(P, f). \end{aligned} \quad (69)$$

This follows from the fact that no cycle sets that are reported as acceptable to an agent are rejected. Note that the case where $\mu(i)R'_i\lambda(i) \wedge \lambda(i)R_i\mu(i)$ is case 1. Statements (68) and (69) thus cover all $S_i \in \Omega$ in case 2. Therefore,

$$\forall S_i \in \{S_i \in \Omega \mid \varphi_i(\{S_i, P_{I \setminus i}\}, f) = \varphi_i(P, f)\} \quad i \in C \in C^\mu, \quad (70)$$

where C is defined by (66). From this, it follows that

$$\begin{aligned} \forall S_i \in \{S_i \in \Omega \mid \varphi_i(\{S_i, P_{I \setminus i}\}, f) = \varphi_i(P, f)\}, \forall i' \in I \quad i' \in C_j, \text{ for some } C_j \in C^\mu \\ \implies \forall S_i \in \Omega \quad (\varphi_i(\{S_i, P_{I \setminus i}\}, f) = \varphi_i(P, f) \implies \varphi(\{S_i, P_{I \setminus i}\}, f) = \mu = \varphi(P, f)). \end{aligned} \quad (71)$$

This proves non-bossiness for case 2. Statements (65) and (71) yield the desired result. \square

The intuition is simple. If agent i participates in some $C \in C^\mu$, where $\varphi(P, f) = \mu$, then i can only prevent the mechanism from selecting μ by preventing C from forming. If agent i is not the agent with highest priority in C , C will form whenever the consumption path i points to in C is preferred to $\lambda(i)$. Agent i can thus only block it by reporting that the consumption path he would be assigned under C is not acceptable, in which case i would be assigned some other consumption path. If agent i is the agent with highest priority in C , C only forms when it is preferred by i to all $C' \in \mathcal{C}_{IR}$ that involve no agents $\neg i \in I$ or objects $a^t \in A^t$ participating in any cycle set that has already formed. C can thus only be blocked by i by reporting that some $C' \in \mathcal{C}_{IR}$ that involves no agent or object participating in any cycle that has already formed is preferred to C . Recall that $(C \in C^\mu)P_i(C' \in C^{\mu'}) \iff \mu(i)P_i\mu'(i)$, implying that i can only block C by being assigned some $x \neq \mu(i)$. Thus, there is no way for i to affect any other agent's assignment and still retain his or her own assignment.

4.2 Negative results

Definition 7. A mechanism Γ is manipulable by a coalition $I' \subset I$ at some $(P, f) \in \prod_{i=1}^N \Omega \times F$ if and only if

$$\exists S \in \prod_{i=1}^N \Omega: \Gamma(f, \{S_{I'}, P_{I \setminus I'}\})P_i \Gamma(f, P) \quad \forall i \in I'. \quad (72)$$

In other words, a mechanism is manipulable whenever it is possible for some coalition of agents to benefit from misrepresenting their preferences. Such a coalition may consist of a single agent. This definition is an adaptation of the definition used by Andersson and Svensson (2008).

Definition 8. A mechanism is (coalitionally) strategy-proof if and only if it is not manipulable by any (coalition $I' \subset I$) agent $i \in I$ at any $(P, f) \in \prod_{i=1}^N \Omega \times F$.

Proposition 4. The mechanism φ is not strategy-proof.

Proof. Consider an allocation problem $\langle T, I, A, P, \lambda \rangle$, where $T = 2$, $I = \{i_1, i_2\}$, $A = \{a_1, a_2\}$, $\lambda(1) = (a_1, a_1)$, $\lambda(2) = (a_2, a_2)$ and P is given by

P_1	P_2
(a_2, a_2)	(a_2, a_1)
(a_2, a_1)	(a_1, a_1)
(a_1, a_2)	(a_1, a_2)
(a_1, a_1)	(a_2, a_2)

Furthermore, let $f = (i_1, i_2)$. This yields $\varphi_1(P, f) = (a_2, a_2)$ and $\varphi_2(P, f) = (a_1, a_1)$. Next, consider strategy S_2 ,

S_2
(a_2, a_1)
(a_2, a_2)
(a_1, a_1)
(a_1, a_2)

and note that $\varphi_1(\{P_1, S_2\}, f) = (a_1, a_2)$ and $\varphi_2(\{P_1, S_2\}, f) = (a_2, a_1)$. Consequently, $\varphi_2(\{P_1, S_2\}, f)P_2\varphi(P, f)$ and the mechanism is not strategy-proof. \square

Corollary 1. *The mechanism is not coalitionally strategy-proof.*

Proof. An agent is a coalition. \square

Pareto optimality is conditional on the assumption that all agents report their true preferences. If there are incentives for agents not to report their true preferences, the mechanism does not necessarily select a Pareto efficient matching plan. This makes manipulability a problematic issue. It is therefore relevant to examine precisely why and when the φ mechanism is vulnerable to manipulation. To illustrate the problem, consider an allocation problem $\langle T, I, A, P, \lambda \rangle$, where $T = 2$, $I = \{i_1, i_2\}$, $A = \{a_1, a_2\}$, $\lambda(i_1) = a_1$, $\lambda(i_2) = a_2$ and P is given by

P_1	P_2
(a_2, a_2)	(a_2, a_1)
(a_2, a_1)	(a_1, a_2)
(a_1, a_2)	(a_1, a_1)
(a_1, a_1)	(a_2, a_2)

Furthermore, let $f = (i_1, i_2)$. In this problem, $\varphi_1(P, f) = (a_2, a_2)$ and $\varphi_2(P, f) = (a_1, a_1)$. A cycle set C is formed such that agent i_1 is assigned his or her most preferred consumption path, while agent i_2 is assigned his or her third most preferred consumption path. Agent

i_2 can improve his or her assignment by blocking C and C can be blocked by reporting that (a_1, a_1) is not acceptable, e.g. by reporting

$$\begin{array}{c} S_2 \\ \hline (a_2, a_1) \\ (a_1, a_2) \\ (a_2, a_2) \\ (a_1, a_1) \end{array}$$

A cycle set C' would form such that $\varphi_1(\{P_1, S_2\}, f) = (a_2, a_1)$ and $\varphi_2(\{P_1, S_2\}, f) = (a_1, a_2)$. Just like above, C' can be blocked by agent i_2 by reporting some S'_2 under which neither (a_1, a_1) nor (a_1, a_2) is acceptable. This would produce a new cycle set, where $\varphi_1(\{P_1, S'_2\}, f) = (a_1, a_2)$ and $\varphi_2(\{P_1, S'_2\}, f) = (a_2, a_1)$. Agent i_1 is then assigned his or her third most preferred consumption path, while agent i_2 is assigned his or her most preferred consumption path. In this manner, an agent can always choose for the other agent among that agent's acceptable consumption paths when $N = 2$.

Whenever $N > 2$, the assignment of the first agent cannot be selected as freely by the second agent. If the second agent blocks a number of cycle sets, a cycle set might form, in which the first agent participates but the second agent does not. For example, consider an allocation problem $\langle T, I, A, P, \lambda \rangle$, where $T = 2$, $I = \{i_1, i_2, i_3\}$, $A = \{a_1, a_2, a_3\}$, $\lambda(i) = a_i$ for $i = 1, 2, 3$ and P is given by

$$\begin{array}{ccc} P_1 & P_2 & P_3 \\ \hline (a_3, a_3) & (a_2, a_1) & (a_2, a_2) \\ (a_1, a_2) & (a_1, a_1) & (a_1, a_1) \\ \dots & \dots & \dots \end{array}$$

Furthermore, let $f = (i_1, i_2)$. In this problem, $\varphi_1(P, f) = (a_3, a_3)$, $\varphi_2(P, f) = (a_1, a_1)$ and $\varphi_3(P, f) = (a_2, a_2)$. There exists a cycle set $C \in \mathcal{C}_{IR}$ in which i_2 would be assigned (a_2, a_1) and i_1 would be assigned (a_1, a_2) . Agent i_2 would clearly prefer this consumption path. However, suppose i_2 blocks the cycle set formed under the strategy profile P by reporting

$$\begin{array}{c} S_2 \\ \hline (a_2, a_1) \\ (a_2, a_2) \\ \dots \end{array}$$

Under $S = \{P_{I \setminus i_2}, S_2\}$, a new cycle set is formed involving only agents i_1 and i_3 , such that $\varphi_1(S, f) = (a_3, a_3)$, $\varphi_2(S, f) = (a_2, a_2)$ and $\varphi_3(S, f) = (a_1, a_1)$. The consumption path (a_2, a_1) is thus unattainable by i_2 . That is, $(a_2, a_1) \notin X_2$. This is true irrespective of the existence of an individually rational cycle set in which i_2 is assigned (a_2, a_1) . Proposition 5 defines precisely when the φ mechanism is not manipulable for the general case.

Proposition 5. *The mechanism φ is not manipulable by any agent $i \in I$ at $(P, f) \in \prod_{i=1}^N \Omega \times F$ if and only if*

$$\forall i \in I, \forall j \leq \#X_i \quad X_i(1)R_iX_i(j). \quad (73)$$

Proof.

$$\begin{aligned} \exists i \in I: \neg(X_i(1)R_iX_i(j)) \text{ for some } j \leq \#X_i &\implies \exists i \in I: X_i(j)P_iX_i(1) \text{ for some } j \leq \#X_i \\ &\implies \exists S_i \in \Omega: \varphi(\{S_i, P_{I \setminus i}\}, f)P_i\varphi(P, f). \end{aligned} \quad (74)$$

The first implication follows from the definition of weak preference. The second implication follows from Lemma 1 and from the fact that $\varphi_i(P, f) = X_i(1)$. Furthermore,

$$\begin{aligned} \exists i \in I: \varphi(\{S_i, P_{I \setminus i}\}, f)P_i\varphi(P, f) \text{ for some } S_i \in \Omega &\implies \exists j \leq \#X_i: X_i(j)P_iX_i(1) \\ &\implies \exists j \leq \#X_i: \neg(X_i(1)R_iX_i(j)). \end{aligned} \quad (75)$$

The first implication follows from Lemma 1 and from the fact that $\varphi(P, f) = X_i(1)$. The second implication follows from the definition of weak preference. By statements (74) and (75),

$$\forall i \in I \quad \nexists S_i \in \Omega: \varphi(\{S_i, P_{I \setminus i}\}, f)P_i\varphi(P, f) \iff \forall i \in I, \forall j \leq \#X_i \quad X_i(1)R_iX_i(j). \quad (76)$$

□

5 Concluding remarks

The purpose of this paper was to present a futures mechanism that solves intertemporal allocation problems, while satisfying individual rationality, Pareto efficiency and non-

bossiness. The φ futures mechanism was presented, and it was proven to satisfy individual rationality, Pareto efficiency and non-bossiness. Furthermore, it was proven that the φ mechanism does not satisfy strategy-proofness and the conditions under which the mechanism is manipulable were presented and proven. The model could be extended by introducing dynamics in the form of entry and exit of agents, as in Bloch and Cantala (2011) and Kurino (2009, 2013), or in the form of production and destruction of objects. It could, however, not be extended by introducing dynamics in preferences, as in Abdulkadiroglu and Loertscher (2007), since this would make futures mechanisms inapplicable. The φ mechanism is presented in a general form and custom tailoring for various real world applications, such as house and office space allocation problems, is possible. Furthermore, the mechanism could be made more general by allowing objects to be assigned to more than one agent per period. This might not make it applicable for school choice problems, but it could make it applicable for e.g. course bidding problems, as studied in a one-period setting by Sönmez and Ünver (2010). Another extension worth considering is to allow the priority structure to differ for different objects or groups of objects, as in Ergin (2002). Finally, it would be possible to devote further studies to providing a complete characterization of the φ mechanism.

6 References

- Abdulkadiroglu, A., & Loertscher, S. (2007). Dynamic House Allocations. Working paper.
- Abdulkadiroglu, A., & Sönmez, T. (1999). House Allocation with Existing Tenants. *Journal of Economic Theory*, 88, 233-260.
- Andersson, T., & Svensson, L-G. (2008). Weakly fair allocations and strategy-proofness. *Review of Economic Design*, 11, 321-338
- Bloch, F., & Cantala, D. (2011). Markovian Assignment Rules. *Social Choice and Welfare*.
- Dur, U. (2011). Dynamic School Choice Problem. Working paper.
- Ergin, H.I. (2002). Efficient Resource Allocation on the Basis of Priorities. *Econometrica*, 70, 2489-2497
- Gale, D., & L.S. Shapley. (1962). College admissions and the stability of marriage. *American Mathematical Monthly*, 69, 9-15.
- Kurino, M. House Allocation with Overlapping Agents: A Dynamic Mechanism Design Approach. (2009). Jena Economic Research Papers, 75.
- Kurino, M. House Allocation with Overlapping Generations. (2013). Working paper.
- Satterthwaite, M., & Sonnenschein, H. (1981). Strategy-proof allocation mechanisms at differential points. *The Review of Economic Studies*, 48, 587-597.
- Shapley, L., & Scarf, H. (1974). On cores and indivisibility. *Journal of Mathematical Economics*, 1, 23-37.
- Sönmez, T., & Ünver, M.U. (2010). Course bidding at business schools. *Interna-*

tional Economic Review, 51, 99-123.

Svensson, L-G. (1994). Queue allocation of indivisible goods. *Social Choice and Welfare*, 11.

Ünver, M.U. (2009). Dynamic Kidney Exchange. *The Review of Economic Studies*, 77, 372-414.

Zou, J., Gujar, S., & Parkes, D. (2010). Tolerable Manipulability in Dynamic Assignment without Money. *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*.