Discussion Paper No. 653

COMPUTATIONAL COMPLEXITY IN THE DESIGN OF VOTING RULES

Koji Takamiya
and
Akira Tanaka

March 2006
Revised July 2006

The Institute of Social and Economic Research
Osaka University
6-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan

Computational complexity in the design of voting rules[*]

Koji TAKAMIYA[†]
Institute of Social and Economic Research,
Osaka University

Akira TANAKA[‡]
Graduate School of
Information Science and Technology,
Hokkaido University

June 29 2006

Abstract

This paper discusses an aspect of computational complexity in social choice theory. We consider the problem of designing voting rules, which is formulated in terms of simple games. We prove that it is an NP-complete problem to decide whether a given simple game is stable, or not.

JEL Classification— C71, D71.

Keywords— computational complexity, NP-completeness, simple game, core, stability, Nakamura number.

0 Introduction

0.1 Motivation: Computational aspects of social choice

This paper studies an aspect of computational complexity in social choice theory. Social choice is innately associated with computational problems. Social choice theory hypothesizes the "social planner." The role of the social planner is to make social decisions by centralized systems, or to set up mechanisms which make social decisions through decentralized processes. In centralized processes, the social planner has to collect and process all the information necessary to make social decisions. This should undoubtedly involve huge computation. Also in decentralized processes, the social planner has to do a lot of computation: The planner designs mechanisms in which individuals act pursuing their own interests. This requires careful examination of the results that these mechanisms will lead to, which should involve computation of, among others, noncooperative equilibrium, and social welfare.

[*]We thank Takuma Wakayama for useful comments. All errors are our own responsibility.

[†]Corresponding author. Institute of Social and Economic Research, Osaka University. 6-1 Mihogaoka Ibaraki Osaka 567-0047 JAPAN. E-mail: takamiya@iser.osaka-u.ac.jp.

[‡]Division of Computer Science, Graduate School of Information Science and Technology, Hokkaido University. Kita14 Nishi9 Sapporo 060-0814 JAPAN.

We endorse the view that computational consideration is much relevant to social choice. This view is traditional (which can date back to Arrow (1951)). It seems, however, that computational problems in social choice have been attracted only limited attention by economists although some computer scientists have studied those problems for years. We suppose that computational consideration in social choice has growing importance in recent years accompanying the expansion in the use of computers in public decisions. In particular, recent real-life applications of the theory of "mechanism design" are worth noting. One of the prominent examples of such applications is the design of matching markets such as hospital-intern matching, school choice and kidney exchange. (Roth and Sotomayor (1990), Abdulkadiroğlu and Sönmez (2003), and Roth, Sönmez and Ünver (2004).)

In considering the computational aspects of social choice, we suppose that the examination based on the theory of **computational complexity** is significant. The theory of computational complexity is a subject in theoretical computer science that provides rigorous treatment of "hardness" of computational problems. If a social choice process involves some computation that is very hard, then it may be unrealistic to use such a process in practice. Therefore, it would be very important to evaluate the hardness of computations associated with social choice. We suppose that the theory of computational complexity provides relevant tools for this evaluation.

One of the most important concepts in computational complexity is that of **NP-completeness**. Consider a problem which is to be answered in "yes" or "no." Then the problem is classified as P if there is an algorithm which solves the problem in polynomial time. This means that the time taken to solve the problem increases polynomially as the size of the problem becomes larger. Whether or not a problem is P gives an important criterion for the tractability of the problem. The problem being not P suggests that the problem is not tractable by direct means of computation.

On the other hand, there are some problems such that if one has a "certificate" that shows that the answer to the problem is "yes," then it can be verified in polynomial time that the answer is indeed "yes." (whether or not the problem is to be solved in polynomial time.) In short, the problem is verifiable in polynomial time. Such problems are classified as NP. Clearly, "verifying" is a weaker requirement than "solving." That is, any P problem is NP, too.

A problem that is **NP-complete** is an NP problem such that any NP problem is "reduced" to that problem. Here, that one problem X is "reduced" to some other problem Y means that the problem X can be restated in the form of the problem Y by some transformation which can be computed in polynomial time. That is to say, the NP-complete problems are the class of NP problems such that by solving any of those problems, one can solve any NP problem with some additional computation that is carried out in polynomial time. Therefore, if at least one NP-complete problem is solved in polynomial time, then every NP problem can be solved in polynominal time, too. In this sense, the NP-complete problems are the set of the most difficult problems in all the NP problems.

At the present time, it is not known whether or not the NP-complete problems can be solved in polynomial time. To paraphrase, it is an open question whether or not any NP problem is in fact P.[1] It is strongly believed, however, that this is not the case, that is, no NP-complete problem is believed to be solved in polynomial time. With this belief, a practical implication of knowing a problem is NP-complete is that the problem is intrinsically difficult, that is, not only being difficult with the present state of knowledge and technology but perhaps difficult in an essential sense.

0.2 Computational Complexity in voting rule design

This paper considers the problem of designing voting rules. In many real-life organizations, people have to design some rules to make collective decisions. Consider a group of people who want to set up such a decision rule. Then there would be two requirements that the group would want to impose on the rule. Firstly, the group would want the rule to properly reflect the distribution of "decision power" specified in advance. For example, in a stockholders' meeting, it would be required that the larger the share one has, the bigger the "decision power" the one possesses. And this may be embodied as the number of votes distributed proportional to the share.

Secondly, the group would want the rule to work for any conceivable cases, that is, any combination of preferences of the members of the group. For example, the simple majority rule can always output winners for any votes. On the contrary, the Condorcet winner rule does not always output winners. It would not be desirable for the group to have decision rules like the latter.

In this paper, we formulate this problem in terms of **simple games**. A simple game represents a "power distribution" in the group in a very simple way: Each subgroup is categorized "powerful" or "not powerful."[2] Formally, a simple game is just a list of the subgroups that are labeled "powerful." To meet the first requirement that we have imposed in the above, we require the voting rule to be a selection or subcorrespondence of the **core** of the simple game.[3] The core is the set of alternatives which cannot be blocked by any "powerful" subgroups. That is to say, the core is those alternatives which do not meet any effective oppositions given the power structure described by the simple game.

To fulfil the second requirement, the core has to be nonempty for every preferences that the members of the group may have. If the core of a simple game has this property, then the simple game is called **stable**. Thus the question of designing a voting rule is to ask whether one has a stable simple game that meets the prescribed power distribution.

In searching for stable simple games, it is essential to check whether any given

[1] This is considered one of the most important open problems in mathematics. See Cook (2000).

[2] One may object that this is too simple a representation of power distribution. By using "effectivity functions," which is a generalization of simple games, one can obtain more elaborate expression of power distribution. However, the purpose of this paper is to show the difficulties in having rules consistent with given power structures. For this purpose, it suffices to consider only special cases . See Remark 2 in Section 3.

[3] Shinotsuka and Takamiya (2003) provides an axiomatic support to this requirement. They showed that the core is the only rule that, in a sense, "properly reflects" the power structure represented by the simple game.

simple game is stable, or not. The question we ask in this paper is how computationally complex this checking is. We prove that it is an NP-complete problem to decide whether a given simple game is stable or unstable. In fact, we prove that the problem is NP-complete not only in the general case but in the special case in which the set of "powerful" subgroups consists only of "large" subgroups. Here, that a subgroup is "large" means that the number of individuals in the subgroup is no less than the total population minus three (if the total population is no less than three.) This appears a very strong restriction so that it seems to significantly reduce the difficulty of the problem. At the same time, this restriction is natural because in many real-life situations, power is assigned only to large subgroups. For the proof, we will utilize the clear-cut result proved by Nakamura (1979) that gives a necessary and sufficient condition for stability.

The significance of our results is that they cast a practical limit on the availability of stable voting rules. Our results suggest that there would be some cases in which the designer of the voting rule cannot check the performance of the rule in a practical sense.

1 Preliminaries

1.1 NP-completeness

The following is only a brief and informal description of NP-completeness. For detailed exposition, Garey and Johnson (1979) is a standard reference. A **decision problem** (**problem**, henceforth) is a problem which is to be answered in "yes" or "no." A problem consists of the descriptions of
(i) **instance**, which is the free parameters of the problem, and
(ii) **question**, which is to be answer in "yes" or "no."

Consider a problem. Then consider algorithms which solve this problem, that is, output "yes" or "no" to each instance as the input.[4] (Here we exclude such algorithms that never stop for certain inputs because this paper deals only with those algorithms which terminate in finite time.) Then we say that the problem belongs to the class $\mathcal{P}$ if there is an algorithm that solves the problem in polynomial time, that is, the maximum time (the number of steps) taken to solve the problem is bounded by some fixed polynomial in the size of the instance (the length of the input).

We say that the problem belongs to the class $\mathcal{NP}$ if there is an algorithm that **verifies** the answer to the problem in polynomial time, that is, if there is an algorithm which runs in polynomial time, and is such that for any given instance, the answer to the problem is "yes" if, and only if, there is a certain additional input, called a **certificate**, with which this algorithm outputs "yes."

It is straightforward to have $\mathcal{P} \subset \mathcal{NP}$. At the present time, however, it is unknown whether $\mathcal{P} = \mathcal{NP}$, or not. This is one of the fundamental problems in computer science and mathematics. (See Cook (2000).) It is widely believed that $\mathcal{P} \neq \mathcal{NP}$.

[4] In this paper, by the term "algorithm," we mean "deterministic algorithm." We are not going into the formal definitions of these concepts here.

Let Π_1 and Π_2 be problems. We say that Π_1 is **polynomial-time reducible to** Π_2 if
(i) there exists a function f that transforms any instance I of Π_1 to the instance $f(I)$ of Π_2 such that the answer to Π_1 for the instance I is "yes" if, and only if, the answer to Π_2 for the instance $f(I)$ is "yes," and
(ii) the function f is embodied by a polynomial-time algorithm.
Intuitively, Π_2 is harder or at least as hard as Π_1. Denote this relation by $\Pi_1 \propto \Pi_2$. Note that $\propto$ is transitive.

A problem Π_1 is called **NP-hard** if for any $\Pi_2 \in \mathcal{NP}$, $\Pi_2 \propto \Pi_1$. Further, Π_1 is called **NP-complete** if Π_1 is NP-hard, and $\Pi_1 \in \mathcal{NP}$. Intuitively, a problem being NP-complete means that the problem is one of the "hardest" among all NP problems. A problem being NP-hard means that it is at least as hard as the NP-complete problems.

There are numerous problems which are known to be NP-complete. One method which is often used in proving a certain problem Π_1 is NP-complete is "reduction." This is done by the following steps:
(i) Specify a known NP-complete problem Π_2.
(ii) Show $\Pi_2 \propto \Pi_1$ by giving a concrete polynomial-time transformation.
(iii) Show $\Pi_1 \in \mathcal{NP}$.
We will utilize this method in the sequel.

Given a problem Π, the **complementary problem** $\bar{\Pi}$ of Π is the problem which has the same instance but has the opposite question to Π. That is, if Π asks if $\exists x : p(x)$, then $\bar{\Pi}$ asks if $\forall x, \neg p(x)$. Let $co\mathcal{NP}$ denote the set of problems Π for which $\bar{\Pi} \in \mathcal{NP}$. It is an open problem whether $co\mathcal{NP} = \mathcal{NP}$. It is strongly believed that $co\mathcal{NP} \neq \mathcal{NP}$. In fact, this is related to the $\mathcal{P} = \mathcal{NP}$ problem: $co\mathcal{NP} \neq \mathcal{NP}$ implies $\mathcal{P} \neq \mathcal{NP}$. Further, if there exists an NP-complete problem Π such that $\bar{\Pi} \in \mathcal{NP}$, then $co\mathcal{NP} = \mathcal{NP}$. (Theorem 7.2 in Garey and Johnson (1979) p. 156.) Thus it is strongly suggested that if a problem is NP-complete, then its complementary problem does not belong to $\mathcal{NP}$. Finally, we note that if a problem Π is NP-complete, then $\bar{\Pi}$ is NP-hard.

1.2 Simple games

A **simple game** is a list $\mathcal{G} = (N, \mathcal{W}, \Omega)$. Here N is a nonempty finite set of individuals. A **coalition** is a nonempty subset of N. $\mathcal{W}$ is a nonempty class of coalitions. Any element of $\mathcal{W}$ is called a **winning coalition**. Any coalition which is not winning is called a **losing coalition**. Ω is a nonempty finite set of **alternatives**.

A **preference relation** on Ω is a weak ordering (i.e. complete and transitive binary relation) over Ω. For each $i \in N$, denote by $\mathcal{R}^i$ the set of preference relations. Denote a generic element of $\mathcal{R}^i$ by $\succeq^i$. As usual, denote the asymmetrical part of $\succeq^i$ by $\succ^i$, and the symmetrical part by $\sim^i$. Let $\mathcal{R} = \prod_{i \in N} \mathcal{R}^i$. Then any element of $\mathcal{R}$ is called a **preference profile**. Denote a generic preference profile by $\succeq$. A **simple game with preferences** is a simple game coupled with a preference profile.

Let a simple game with preferences $(\mathcal{G}, \succeq)$ be given. Let $x, y \in \Omega$. And let S

be a coalition. Then let us denote $x \succ^S y$ if

$$\forall i \in S, \; x \succ^i y.$$

Then the **core** of $(\mathcal{G}, \succeq)$ is the set

$$\{x \in \Omega \mid \nexists y \in \Omega : \exists S \in \mathcal{W} : y \succ^S x\}.$$

A simple game $\mathcal{G}$ is called **stable** if the core of $(\mathcal{G}, \succeq)$ is nonempty for any $\succeq \in \mathcal{R}$. One of the fundamental questions in the theory of simple games is whether a given simple game is stable, or not. Nakamura (1979) gave the answer to this question by providing a necessary and sufficient condition for stability in terms of "Nakamura number."

Call a simple game $\mathcal{G}$ **weak** if $\bigcap \mathcal{W} \neq \emptyset$. Let $\mathcal{G}$ be a simple game which is not weak. Then the **Nakamura number** of $\mathcal{G}$ is

$$\mathcal{V}(\mathcal{G}) = \min\{\#\sigma \mid \emptyset \neq \sigma \subset \mathcal{W}, \; \bigcap \sigma = \emptyset\}.$$

Theorem 1 (Nakamura, 1979) *Let a simple game $\mathcal{G}$ be given. $\mathcal{G}$ is stable if, and only if, (i) $\mathcal{G}$ is weak, or (ii) $\#\Omega < \mathcal{V}(\mathcal{G})$.*

In the sequel, we will utilize this theorem for proving our main results.

2 Results

Let us define the problem that we will examine.

NAME: UNSTABILITY.
INSTANCE: A finite set N; A collection $\mathcal{W} = \{W_1, \cdots, W_m\}$ of subsets of N; A finite set Ω.
QUESTION: Is the simple game $\mathcal{G} = (N, \mathcal{W}, \Omega)$ not stable?

Let us refer to the complementary problem of UNSTABILITY as STABILITY. One may think that STABILITY, rather than UNSTABILITY, is the natural question to ask. However, as we will see, STABILITY may not belong to $\mathcal{NP}$ whereas UNSTABILITY does (see just below Theorem 2), but both problems are NP-hard. Although these two problems are distinct problems from the view point of computation, their practical meaning is the same.

Although UNSTABILITY is our primary problem, we consider an "easier" version of UNSTABILITY so that we will obtain stronger results, In this "easier" version, winning coalitions are restricted to "large" coalitions. Here, that a coalition is "large" means that the number of individuals in the coalition is no less than the total population minus three if the total population is no less than three. This restriction is natural because in many real-life situations, power is assigned only to large subgroups. Also, this restriction appears so strong that it seems to largely reduce the difficulty of the problem. As we will prove, however, this "easier" version is in fact as difficult as the original problem. We call this problem

3-UNSTABILITY, and its complementary problem 3-STABILITY. In the following, we prove that even 3-UNSTABILITY is NP-complete.

NAME: 3-UNSTABILITY.
INSTANCE: A finite set N; A collection $\mathcal{W} = \{W_1, \cdots, W_m\}$ of subsets of N with for all $i = 1, \cdots, m$, $\#W_i \geq \#N - 3$ if $\#N \geq 3$; A finite set Ω.
QUESTION: Is the simple game $\mathcal{G} = (N, \mathcal{W}, \Omega)$ not stable?

Lemma 1 *3-UNSTABILITY is NP-hard.*

Proof of Lemma 1 The proof is done by a simple reduction. We use the following known NP-complete problem ([SP5] in Garey and Johnson (1979)).

NAME: 3-MINIMUM COVER.
INSTANCE: A finite set S; A collection of $\mathcal{C} = \{C_1, \cdots, C_m\}$ of subsets of S with for all $i = 1, \cdots, m$, $\#C_i \leq 3$; a positive integer $k \leq m$.
QUESTION: Does $\mathcal{C}$ contain a cover for S with at most k subsets? In other words, does there exist a collection $\mathcal{D}$ such that $\mathcal{D} \subset \mathcal{C}$, $\#\mathcal{D} \leq k$, and $\bigcup \mathcal{D} = S$?

Let an instance $(S, \mathcal{C}, k)$ of 3-MINIMUM COVER be given. Let us construct a simple game $\mathcal{G} = (N, \mathcal{W}, \Omega)$ as follows: Let $N := S$, $\mathcal{W} := \{N \setminus C \mid C \in \mathcal{C}\}$ and $\Omega := \{1, 2, \cdots, k\}$. Then clearly, $\mathcal{W}$ consists of "large" coalitions only, i.e. $\forall W \in \mathcal{W},\ \#W \geq \#N - 3$. And we have the following relations:

$$\bigcup \mathcal{C} = S \Leftrightarrow \bigcap \mathcal{W} = \emptyset, \tag{1}$$

$$\bigcup \mathcal{C} = S \Rightarrow \min\{\#\mathcal{D} \mid \mathcal{D} \subset \mathcal{C},\ \bigcup \mathcal{D} = S\} = \mathcal{V}(\mathcal{W}) \tag{2}$$

Suppose that the answer to 3-MINIMUM COVER is "yes." Then we have $\bigcup \mathcal{C} = S$ and $\min\{\#\mathcal{D} \mid \mathcal{D} \subset \mathcal{C},\ \bigcup \mathcal{D} = S\} \leq k$. This implies that the simple game is not weak by (1), and that $\mathcal{V}(\mathcal{W}) \leq \#\Omega$ by (2). Then by Theorem 1, the answer to 3-UNSTABILITY for the instance which has just been constructed is "yes."

Suppose that the answer to 3-UNSTABILITY is "yes." Then by Theorem 1, it follows that the simple game is not weak and that $\mathcal{V}(\mathcal{W}) \leq \#\Omega$. By (1) and (2), this implies $\bigcup \mathcal{C} = S$ and $\min\{\#\mathcal{D} \mid \mathcal{D} \subset \mathcal{C},\ \bigcup \mathcal{D} = S\} \leq k$, that is, the answer to 3-MINIMUM COVER is "yes." □

Lemma 2 *UNSTABILITY belongs to $\mathcal{NP}$.*

Proof of Lemma 2 Let an instance of UNSTABILITY be given. Then by Theorem 1, the answer to the problem is "yes" for this instance if, and only if, there exists a class of coalitions σ such that

$$\emptyset \neq \sigma \subset \mathcal{W},\ \bigcap \sigma = \emptyset \text{ and } \#\sigma \geq \#\Omega. \tag{3}$$

Then the class σ constitutes the certificate. And the time taken to check whether σ satisfies (3) grows only in linear order. Thus it is checkable in polynomial time. □

From these two lemmas, we have the following conclusion.

Theorem 2 *UNSTABILITY and* 3*-UNSTABILITY are both NP-complete.*

As mentioned in Sec. 1.1 (in the last paragraph), if there exists some NP-complete problem whose complementary problem belongs to $\mathcal{NP}$, then $co\mathcal{NP} = \mathcal{NP}$. (Theorem 7.2 in Garey and Johnson (1979) p. 156). Thus the hypothesis $co\mathcal{NP} \neq \mathcal{NP}$, which is supported widely, implies that (3-)STABILITY is considered not to belong to $\mathcal{NP}$. However, we still have the following.

Corollary 1 *STABILITY and* 3*-STABILITY are both NP-hard.*

The following result, which immediately follows from Theorem 2, may be of some interest. Let us consider the problem of unstability stated in the below, in which winning coalitions are restricted to those coalitions which exceeds in size some fixed proportion of the total population. Let a number p with $0 \leq p < 1$ be given.

NAME: QUOTAp-UNSTABILITY.
INSTANCE: A finite set N; A collection $\mathcal{W} = \{W_1, \cdots, W_m\}$ of subsets of N with for all $i = 1, \cdots, m$, $\#W_i \geq p\#N$; A finite set Ω.
QUESTION: Is the simple game $\mathcal{G} = (N, \mathcal{W}, \Omega)$ not stable?

Corollary 2 *For any* p *with* $0 \leq p < 1$*, QUOTAp-UNSTABILITY is NP-complete.*

Proof of Corollary 2 Clealy, there is some $\bar{N}$ such that $N > \bar{N}$ implies $p\#N \leq \#N - 3$. Thus for sufficiently large N, the problem includes 3-UNSTABILITY as special cases. Thus for the inputs which are sufficiently large, the maximum running time is at least as large as that of 3-UNSTABILITY. Then Theorem 2 implies that the problem is NP-complete. $\square$

3 Remarks

Remark 1 Consider "2-UNSTABILITY," a natural variant of 3-UNSTABILITY, that is, the problem of unstability in which winning coalitions are restricted to such coalitions that are no smaller than the total population minus two (instead of three) if the total population is no less than two. Then this problem belongs to $\mathcal{P}$. This is because there is a polynomial time algorithm for solving "2-MINIMUM COVER," a natural variant of 3-MINIMUM COVER, that is, the problem of checking if there is a cover of S in the family of subsets $\mathcal{C}$ of S with $\#C \leq 2$ for all $C \in \mathcal{C}$. (See [SP5] in Garey and Johnson (1979)).

Remark 2 Mizutani, Hirade and Nishino (1993), and Boros and Gurvich (2000) showed that checking the unstability of effectivity functions is NP-complete. Simple games can be regarded as a special case of effectivity functions (Moulin and Peleg, 1982). Thus our results imply that the unstability of effectivity functions is NP-hard. Therefore, provided that the unstability of effectivity functions belongs to $\mathcal{NP}$, which can be proved by Keiding's condition for stability (Keiding, 1985)), their NP-completeness result follows from our results. Further, it is worth noting that our proof of NP-hardness is significantly simpler than theirs.

Remark 3 Bartholdi, Narasimhan and Tovey (1991) is closely related to our paper. Whereas our interest is about simple games, a general model, they study spatial voting games, a more specific model. Their results have some overlap with ours: They showed that it is NP-complete to check the unstability of q-weighted majority voting games. This class of voting games is a subclass of simple games. Thus clearly, their result implies that UNSTABILITY is NP-hard, which is a part of our conclusions. To clarify what is new in our paper, we note that the NP-hardness and completeness of 3-UNSTABILITY, the strongest part of our results, is independent of the result by Bartholdi *et al.* The class of simple games with "large" winning coalitions, which is assumed in 3-UNSTABILITY, does not include nor is not included by the set of q-weighted majority voting games, which Bartholdi *et al* deals with. We are not aware of similar results to ours reported elsewhere. Furthermore, our proof of NP-hardness is much simpler than theirs.

References

[1] Abdulkadiroğlu A and Sönmez T (2003): School choice: A mechanism design approach. *Amer Econ Review* 93: 729–47.

[2] Arrow KJ (1951): *Social Choice and Individual Values.* Wiley. New York.

[3] Bartholdi III JJ, Narasimhan LS and Tovey CA (1991): Recognizing majority-rule equilibrium in spatial voting game. *Soc Choice Welfare* 8: 183–97.

[4] Boros E and Gurvich V (2000): Stable effectivity functions and perfect graphs. *Math Soc Sciences* 39: 175–94.

[5] Cook S (2000): The P versus NP problem. Clay Mathematics Institute. available at (http://www.claymath.org/millennium/P_vs_NP/Official_Problem_Description.pdf.)

[6] Garey MR and Johnson DS (1979): *Computers and Intractability: A Guide to the Theory of NP-completeness.* WH Freeman. New York.

[7] Keiding H (1985): Necessary and sufficient condition for stability of effectivity functions. *Int J Game Theory* 14: 93–101.

[8] Mizutani M, Hiraide Y and Nishino H (1993): Computational complexity to verify the unstability of effectivity function. *Int J Game Theory* 22: 225–39.

[9] Moulin H and Peleg B (1982): Cores of effectivity functions and implementation theory. *J Math Econ* 10: 115–45.

[10] Nakamura K (1979): The vetoer in a simple game with ordinal preferences. *Int J Game Theory* 8: 55–61

[11] Roth A, Sönmez T and Ünver U (2004): Kidney exchange. *Quarterly J Econ* 119: 457–88.

[12] Roth A and Sotomayor M (1990): *Two-Sided Matching: A Study in Game Theoretic Modeling and Analysis.* Cambridge University Press. London/New York.

[13] Shinotsuka T and Takamiya K (2003): The weak core of simple games with ordinal preferences: Implementation in Nash equilibrium. *Games Econ Behavior* 44: 379–89.