Available online at www.sciencedirect.com
SCIENCE DIRECT®

Games and Economic Behavior 49 (2004) 424–434

GAMES and Economic Behavior

www.elsevier.com/locate/geb

Note

Nash implementation and uncertain renegotiation

Pablo Amorós

Departamento de Teoría e Historia Económica, Universidad de Málaga, Campus El Ejido, E-29013 Málaga, Spain
CentrA, Spain

Received 26 February 2003
Available online 21 April 2004

Abstract

We study Nash implementation when the outcomes of the mechanism can be renegotiated among the agents but the planner does not know the renegotiation function that they will use. We characterize the social objectives that can be implemented in Nash equilibrium when the same mechanism must work for every admissible renegotiation function, and show the importance of allowing the planner to sometimes take away resources from the agents.

JEL classification: C70; D78

Keywords: Implementation theory; Nash equilibrium; Renegotiation function

1. Introduction

Implementation theory concerns the problem of designing mechanisms (or game forms) whose equilibrium outcomes are desirable according to the objectives of a planner. Most papers related with implementation assume implicitly that the outcomes of the mechanism will be enforced.[1] Nevertheless, there exist numerous situations in which the agents are not bound to the mechanism.[2] In particular, Maskin and Moore (1999) argued that if the outcome of the mechanism is not Pareto-efficient from the agents' perspective, they may decide to renegotiate it. Then, they considered implementation where any Pareto-inefficient outcome suggested by the mechanism is replaced by a Pareto-superior outcome

E-mail address: pag@uma.es (P. Amorós).

[1] There are some studies which have addressed the ability of the planner to enforce out-of-equilibrium outcomes that are known to be undesirable (see, for example, Baliga et al., 1997 and Chakravorti et al., 2003).

[2] See Hurwicz (1994) for a discussion of enforceability in mechanism design.

0899-8256/$ – see front matter
doi:10.1016/j.geb.2004.03.001

according to an exogenous renegotiation function. In the same spirit, Jackson and Palfrey (2001) analyzed implementation where a general state-contingent function converts the outcomes of the mechanism. This function allows them to deal with different problems of enforcement other than renegotiation (for instance, the state-contingent function could also model individual rationality constraints).[3] In this paper, however, we focus on its interpretation as a renegotiation function.

Given any such renegotiation function, Maskin and Moore (1999) and Jackson and Palfrey (2001) obtain characterizations of Nash implementation that have intuitive relationships to the standard results (see, e.g., Maskin, 1999). It must be stressed, however, that these characterizations depend on the exogenous specification of the renegotiation function.[4] This poses a new problem since, in many settings, the planner may not know the specific renegotiation function when the mechanism is designed (although the planner knows that any Pareto-inefficient outcome will be renegotiated, he may not know the particular bargaining strengths of each agent when he designs the mechanism).

This paper aims to study the Nash implementation problem in this sort of situations. For that, we assume that there exists a set of admissible renegotiation functions, G, so that, although the planner knows that the true renegotiation function must be in that set, the precise function is unknown to him. Specifically, we make two reasonable assumptions about admissible renegotiation functions:

(1) renegotiated outcomes are always Pareto-efficient, and
(2) no agent ends up worse off after renegotiating.

In the spirit of the Nash equilibrium concept, we assume that the agents know the true renegotiation function in G (i.e., they know each other's bargaining strengths when playing the mechanism), but it is unknown to the planner. In this framework, we propose a new form of implementation where the same mechanism must work for every admissible renegotiation function in G (which we call Nash implementation in G). The characterizations of Nash implementation when the renegotiation function is fixed are extended to this setting. Not surprisingly, the fact that the planner ignores the renegotiation function acts as a constraint on what can be implemented.

Next, we examine some applications within the classical exchange environment. In this setting, allowing the planner to sometimes take away resources from the agents (i.e., free-disposal) seems to be crucial for Nash implementability in G. We show that the constrained Walrasian correspondence, the core correspondence, and the Pareto-efficient and envy-free correspondence are Nash implementable in G if and only if free-disposal is allowed. More specifically, if free-disposal is not allowed, none of these correspondences is Nash implementable for some given admissible renegotiation functions (even if the planner knew that functions), and therefore Nash implementation in G is not possible

[3] Ma et al. (1988) were the first to point out that individual rationality constraints must be imposed both in and out of equilibrium. Jackson and Palfrey (2001) proposed to model these constraints by means of a function that reverts any non-individually rational outcome of the mechanism to the status quo.

[4] In the face of individual rationality constraints, Jackson and Palfrey (2001) endogenized the state-contingent function in the context of a dynamic model where agents can force the mechanism to be replayed. In a previous paper, Jackson and Palfrey (1998) examined this form of implementation in a bargaining model.

either. If free-disposal is allowed, on the other hand, these correspondences are Nash implementable in G. The intuition behind this result is the following. Renegotiation acts as a constraint on what outcomes (profiles of commodity bundles) can arise not only in equilibrium, but also off the equilibrium path. When free-disposal is not allowed, the off-equilibrium path constraints associated with some admissible renegotiation functions prevent the mechanisms from enforcing equilibrium behavior and breaking undesirable equilibria at the same time. Under free-disposal, however, the planner can give any feasible commodity bundle to an agent and prevent him from renegotiating by throwing away the rest of resources (so that the other agents receive nothing). Although free-disposal cannot occur on the equilibrium path, these "non-renegotiable" outcomes can be selected by the mechanisms off the equilibrium path to enforce equilibrium behavior and to assure that undesired profiles of strategies are not equilibria for any admissible renegotiation function (making Nash implementation in G possible).

The rest of the paper is organized as follows. Section 2 presents the general setting. Section 3 establishes necessary and sufficient conditions for Nash implementation in G. Section 4 deals with the free-disposal condition for Nash implementation in G in economic environments. Finally, Section 5 makes some concluding remarks.

2. Definitions

Consider an environment with a known set A of feasible *alternatives or outcomes*, a set $I = \{1, 2, \dots, n\}$ of *agents*, and a set S of possible *states*.

Agents' *preferences* over alternatives depend on the state: for each state $s \in S$, each agent $i \in I$ has a preference ordering $R_i(s)$ on the set A. Let $P_i(s)$ denote the strict part of $R_i(s)$.

Let 2^A denote the set of all subsets of A. A *social choice rule* (SCR) is a correspondence $F : S \to 2^A$, which associates each state s with a subset of the set alternatives.

An SCR is supposed to represent the objectives of a social planner. The implementation problem arises when the planner cannot identify directly the outcomes recommended by the SCR. To obtain these alternatives in a decentralized way, the planner must design a *mechanism*. A mechanism is a pair $\Gamma = (M, h)$, where $M = \times_{i=1}^{n} M_i$, M_i is the set of possible *messages* for agent i, and $h : M \to A$ is the *outcome function*.

Suppose that the mechanism is $\Gamma = (M, h)$ and the agents report $m \in M$ at state $s \in S$. If $h(m)$ is inefficient from the agents' perspective at state s, they might decide to renegotiate the outcome to something which Pareto-dominates it. Following Maskin and Moore (1999), we suppose that the renegotiation process can be expressed as a function $g : A \times S \to A$, which we call *renegotiation function*.

Taking into account that the final outcome may not come directly from the mechanism but instead from the renegotiation function, we have the following natural extensions of the standard notions of Nash equilibrium and Nash implementation:

Definition 1. The message profile $m \in M$ is a *g-Nash equilibrium* of mechanism $\Gamma = (M, h)$ at state $s \in S$ when $g(h(m), s) R_i(s) g(h(\widehat{m}_i, m_{-i}), s)$ for all $i \in I$ and all $\widehat{m}_i \in M_i$. Let $N_g(\Gamma, s)$ denote the set of g-Nash equilibria of Γ at s.

Definition 2. The mechanism $\Gamma = (M, h)$ *g-Nash implements* the SCR F when, for all $s \in S$:

(1) For all $a \in F(s)$ there exists $m \in N_g(\Gamma, s)$ such that $g(h(m), s) = a$.
(2) If $m \in M$ is such that $g(h(m), s) \notin F(s)$, then $m \notin N_g(\Gamma, s)$.

If such a mechanism exists then F is g-Nash implementable.

Jackson and Palfrey (2001) showed that if an SCR is g-Nash implementable, then it satisfies the following condition:

Definition 3. An SCR F is *g-monotonic* when, for all $s \in S$ and all $a \in F(s)$, there exists $z \in A$ such that:

(1) $g(z, s) = a$.
(2) For all $s' \in S$ such that $g(z, s') \notin F(s')$, there exist $y \in A$ and $i \in I$ such that $g(z, s) R_i(s) g(y, s)$ and $g(y, s') P_i(s') g(z, s')$.

Furthermore, Jackson and Palfrey showed that if there are at least three agents, g-monotonicity is not only a necessary condition for g-Nash implementability, but it is sufficient when combined with g-no veto power:

Definition 4. An SCR F satisfies *g-no veto power* (*g-NVP*) when, for all $s \in S$, all $z \in A$, and all $i \in I$, the following is true: if $g(z, s) R_j(s) g(y, s)$ for all $y \in A$ and all $j \neq i$, then $g(z, s) \in F(s)$.

It must be stressed that in the g-Nash implementation approach the renegotiation function is taken as fixed (i.e., g is part of the data of the problem). Then, the choice of the mechanism depends on g. It may be, however, that the true renegotiation function that the agents will use is unknown to the planner when the mechanism is designed. This poses a new problem since the fact that a mechanism g-Nash implements F does not necessarily imply that the same mechanism $\tilde{g}$-Nash implements F for some other $\tilde{g} \neq g$.

We study Nash implementation in this sort of situations. For that purpose, we assume that there is a set G of *admissible renegotiation functions* such that, although the planner knows that the true renegotiation function must be in that set, the precise function is unknown to him.

We always assume that any function in G satisfies the following two properties:

Pareto-efficiency. For all $a \in A$ and all $s \in S$, there is no $b \in A$ such that $b R_i(s) g(a, s)$ for all $i \in I$, with strict preference for some $i \in I$.

Individual rationality. For all $a \in A$, all $s \in S$, and all $i \in I$, $g(a, s) R_i(s) a$.

In the spirit of the Nash equilibrium concept, we assume that the agents know the true renegotiation function $g \in G$. In this case, when designing the mechanism, the planner knows that the agents will play any mechanism according to the g-Nash equilibrium concept for some $g \in G$, but he is unaware of the precise renegotiation function. Therefore, if the planner wants to be sure of implementing the SCR F, the same mechanism should g-Nash implement F for all $g \in G$. This is what we call Nash implementation in G.

Definition 5. An SCR F is *Nash implementable in G* if and only if there exists a single mechanism which g-Nash implements F for all $g \in G$.

3. Necessary and sufficient conditions for Nash implementation in *G*

In this section, we study necessary and sufficient conditions for Nash implementation in G. The key condition is a generalization of the g-monotonicity condition. It takes into account the fact that, for any given alternative and state, the final outcome may depend on the true renegotiation function.

Definition 6. An SCR F is *monotonic in G* when, for all $g \in G$, $s \in S$, and $a \in F(s)$, there exists $z \in A$ such that:

(1) $g(z, s) = a$.
(2) For all $\tilde{g} \in G$ and $s' \in S$ such that $\tilde{g}(z, s') \notin F(s')$, there exists $y \in A$ and $i \in I$ such that $g(z, s)R_i(s)g(y, s)$ and $\tilde{g}(y, s')P_i(s')\tilde{g}(z, s')$.

The intuition behind this condition is that, if $a \in F(s)$, then Nash implementability in G implies the existence of a single mechanism such that, for each $g \in G$, there exists a g-Nash equilibrium at s yielding a as final outcome. Moreover, if for some other $\tilde{g} \in G$ the final outcome associated with this equilibrium is not F-optimal in other state s', then it cannot be a $\tilde{g}$-Nash equilibrium at s'.

Theorem 1. *If the SCR F is Nash implementable in G, then F is monotonic in G.*

Proof. Let $\Gamma = (M, h)$ be a mechanism that Nash implements in G the SCR F. Let $g \in G$, $s \in S$, and $a \in F(s)$. Then, there exists $m \in N_g(\Gamma, s)$ with $g(h(m), s) = a$. Let $\tilde{g} \in G$ and $s' \in S$ be such that $\tilde{g}(h(m), s') \notin F(s')$. Then $m \notin N_{\tilde{g}}(\Gamma, s')$ and so, there exist $i \in I$ and $\widehat{m}_i \in M_i$ such that, $\tilde{g}(h(\widehat{m}_i, m_{-i}), s')P_i(s')\tilde{g}(h(m), s')$. Since $m \in N_g(\Gamma, s)$, $g(h(m), s)R_i(s)g(h(\widehat{m}_i, m_{-i}), s)$. Let $z = h(m)$ and $y = h(\widehat{m}_i, m_{-i})$ to satisfy the definition of monotonicity in G. □

Although monotonicity in G alone is not a sufficient condition for Nash implementation in G, it is sufficient when combined with g-NVP for all $g \in G$. The proof of this result follows the logic of the proofs of Nash implementability: we propose an extension of the canonical mechanism for Nash implementation where, besides announcing a state, an outcome, and an integer, each agent also has to announce a renegotiation function.[5]

[5] The purpose of this mechanism is not to be used as a method for actual decision making, but to find theoretical limits on what can be implemented.

Theorem 2. *If $n \geqslant 3$ and the SCR F is monotonic in G and satisfies g-NVP for all $g \in G$, then F is Nash implementable in G.*

Proof. For all $g \in G$, all $s \in S$ and all $a \in F(s)$, let $A_g^{s,a}$ be the set of alternatives satisfying points (1) and (2) of the definition of monotonicity in G.

Let $\Gamma = (M, h)$ be the following mechanism. For all $i \in I$, the message space is $M_i = G \times S \times A \times \{0, 1, 2, \ldots\}$. The outcome function $h : M \to A$ is defined as follows:

Rule 1. If there are $g \in G$, $s \in S$, $a \in F(s)$, and $z \in A_g^{s,a}$ such that $m_i = (g, s, z, 0)$ for all $i \in I$, then $h(m) = z$.

Rule 2. Suppose there exist $g \in G$, $s \in S$, $a \in F(s)$, $z \in A_g^{s,a}$, and $j \in I$ such that $m_i = (g, s, z, 0)$ for all $i \neq j$, but $m_j = (\tilde{g}, s', y, k) \neq (g, s, z, 0)$. Then

$$h(m) = \begin{cases} z, & \text{if } g(y, s) P_j(s) g(z, s), \\ y, & \text{if } g(z, s) R_j(s) g(y, s). \end{cases}$$

Rule 3. In all other cases, let $h(m)$ be the alternative announced by the agent who announced the highest integer (possible ties are broken by choosing the agent with the lowest index).

Step 1. For all $g \in G$, all $s \in S$ and all $a \in F(s)$, there is $m \in N_g(\Gamma, s)$ such that $g(h(m), s) = a$.

Let $g \in G$, $s \in S$ and $a \in F(s)$. Since F is monotonic in G, there exists $z \in A_g^{s,a}$. Let $m \in M$ be such that $m_i = (g, s, z, 0)$ for all $i \in I$. Then Rule 1 applies to m and $h(m) = z$. Since $z \in A_g^{s,a}$, $g(h(m), s) = a$. Moreover, $m \in N_g(\Gamma, m)$. To see this consider any unilateral deviation by some agent i to $\widehat{m}_i = (\tilde{g}, s', y, k)$. Then Rule 2 comes into effect, and therefore $g(h(m), s) R_i(s) g(h(\widehat{m}_i, m_{-i}), s)$.

Step 2. For all $g \in G$, all $s \in S$ and all $m \in N_g(\Gamma, s)$, $g(h(m), s) \in F(s)$.

Let $g \in G$, $s \in S$ and $m \in N_g(\Gamma, s)$. Suppose first that Rule 1 applies to m. Then, there are $\tilde{g} \in G$, $s' \in S$, $a' \in F(s')$ and $z' \in A_{\tilde{g}}^{s',a'}$ such that $m_i = (\tilde{g}, s', z', 0)$ for all $i \in I$. Therefore $h(m) = z'$ and $\tilde{g}(z', s') = a'$. Suppose by contradiction that $g(z', s) \notin F(s)$. By monotonicity in G, there exist $y \in A$ and $i \in I$ such that $\tilde{g}(z', s') R_i(s') \tilde{g}(y, s')$ and $g(y, s) P_i(s) g(z', s)$. Consider a unilateral deviation by agent i to $\widehat{m}_i = (g, s, y, 1)$. By Rule 2 we have $h(\widehat{m}_i, m_{-i}) = y$, which contradicts that $m \in N_g(\Gamma, s)$.

Suppose now that either Rule 2 or Rule 3 applies to m. Then there is $j \in I$ such that, by making a unilateral deviation, any agent $i \neq j$ can make the mechanism to select any alternative $y \in A$ via Rule 3. Therefore, since $m \in N_g(\Gamma, s)$, for all $y \in A$ and all $i \neq j$, $g(h(m), s) R_i(s) g(y, s)$. Then, since F satisfies g-NVP, $g(h(m), s) \in F(s)$. □

4. Applications

In this section we study the Nash implementability in G of some well-known social choice rules and show the importance of allowing the planner to sometimes take away resources from the agents.

Consider an exchange environment where each agent $i \in I$ owns a bundle $\omega_i \in \mathbb{R}^l_+$ of l commodities that is fixed and known. For each state $s \in S$, each agent $i \in I$ has a preference ordering $R_i(s)$ over $\mathbb{R}^l_+$ that is continuous, strictly convex, and strictly monotone. Let $A = \{a \in \mathbb{R}^{l\times n}_+ \colon \sum a_i \leqslant \sum \omega_i\}$ be the set of feasible allocations, and let $A^* = \{a \in \mathbb{R}^{l\times n}_+ \colon \sum a_i = \sum \omega_i\}$ be the set of feasible allocations in which no resource is ever thrown away (where a_i denotes the ith entry of a). Denote the vector of market prices for commodities by $p \in \mathbb{R}^l$. The allocation $a \in A$ and the price vector $p \in \mathbb{R}^l$ constitute a *constrained Walrasian equilibrium* at $s \in S$ if, for each agent $i \in I$, a_i maximizes $R_i(s)$ over the set $\{b_i \in \mathbb{R}^l_+ \colon b_i \leqslant \sum \omega_i$ and $pb_i \leqslant p\omega_i\}$. The *constrained Walrasian correspondence*, $W : S \to 2^A$, selects for each state s the set of feasible allocations that can be supported as a constrained Walrasian equilibrium for some $p \in \mathbb{R}^l$.

In this context, we say that *free-disposal* is allowed when the alternatives selected by the implementing mechanisms can be such that $\sum h_i(m) < \sum \omega_i$ (so that not all the resources are distributed among the agents). We say that a renegotiation function is *feasible* when, for all $a \in A$ and $s \in S$, $\sum g_i(a,s) \leqslant \sum a_i$. Let G^* be the set of all feasible renegotiation functions satisfying Pareto-efficiency and individual rationality.

Our next proposition shows that if free-disposal is not allowed (i.e., if no resource is ever thrown away), then there is some $g \in G^*$ such that W violates the necessary condition for g-Nash implementation.

Proposition 1. *If free-disposal is not allowed, then the constrained Walrasian correspondence violates g-monotonicity for some $g \in G^*$.*

Proof. Note that if free-disposal is not allowed, then any implementing mechanism $\Gamma = (M, h)$ must be such that $h(m) \in A^*$ for all $m \in M$. Consider the two-person, two-good example represented in Fig. 1. There are two states, s and s'. In state s, the agents have indifference curves I_1^s and I_2^s, respectively. In state s', the indifference curves of agent 1 are represented by the dotted curves $I_1^{s'}$, while the preferences of agent 2 do not change (i.e., $I_2^{s'} = I_2^s$). In this example, allocation a can be supported as a constrained Walrasian equilibrium at state s for the price vector p (i.e., $a \in W(s)$). Let $g \in G^*$ be a renegotiation function such that any possible gains from renegotiation are given to agent 1.[6] Let $z \in A^*$ be such that $g(z,s) = a$. Note that: (1) z must belong to the indifference curve $I_2^s(a)$ of agent 2, and (2) $g(z,s') = g(a,s')$. Moreover, $g(a,s')$ cannot be supported as a constrained Walrasian equilibrium at s', and then $g(z,s') \notin W(s')$. It is easy to see, however, that any $y \in A^*$ such that $g(y,s')P_1(s')g(z,s')$ must belong to an indifference curve of agent 2 that passes below $I_2^s(a)$, and therefore $g(y,s)P_1(s)g(z,s)$. Similarly, any $y \in A^*$ such that $g(y,s')P_2(s')g(z,s')$ must belong to an indifference curve of agent 2 that passes above $I_2^s(a)$, and therefore $g(y,s)P_2(s)g(z,s)$. Hence, F violates g-monotonicity. □

Roughly speaking, Proposition 1 says that if free-disposal is not allowed, then there are some admissible renegotiation functions for which Nash implementation of W is not

[6] We have chosen this renegotiation function to ease the exposition. One can find similar examples with symmetric renegotiation functions.

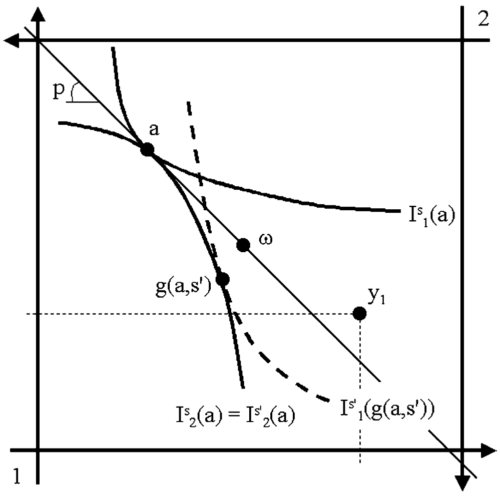

Fig. 1. Illustration of Proposition 1.

possible even if the planner knew the specific renegotiation function. Naturally, this implies that Nash implementation in G^* is not possible either.[7]

The former impossibility result can be avoided if we allow free disposal. Let us reconsider the example analyzed in Proposition 1. Let $y = (y_1, 0) \in A$ be an allocation where agent 1 gets y_1 (as represented in Fig. 1) and agent 2 gets nothing (the rest of resources, $\omega - y_1$, are thrown away). Note that $aR_1(s)y$, $yP_1(s')g(a,s')$, and $g(y,s) = g(y,s') = y$ (since agent 2 has nothing, no renegotiation is possible). Then $g(a,s) = a$, $g(a,s)R_1(s)g(y,s)$, and $g(y,s')P_1(s')g(a,s')$, so that g-monotonicity holds. In fact, if free-disposal is allowed, then W is Nash implementable in G^*.[8]

Proposition 2. *If $n \geqslant 3$ and free-disposal is allowed, then the constrained Walrasian correspondence is Nash implementable in G^*.*

Proof. We will show that, under free-disposal, W verifies the sufficient conditions for Nash implementation in G^* stated in Theorem 2. First note that, as soon as there are at least three agents, g-NVP is trivially satisfied for all $g \in G^*$, since its hypothesis is never met (given that preferences are strictly monotone). Next we show that W satisfies monotonicity in G^*. Let $g \in G^*$, $s \in S$, and $a \in W(s)$. By the first Fundamental Theorem

[7] If an SCR is Nash implementable in G, then it is g-Nash implementable for all $g \in G$. The converse implication is not true (the fact that for each $g \in G$ there exists a mechanism of g-Nash implementing F does not guarantee that there exists a single mechanism of g-Nash implementing F for all $g \in G$).

[8] One could view free disposal as a problematic assumption, since it allows inefficient outcomes to stand. Thus, in a contractual context where there is no planner and the mechanism is a sort of constitution, the agents could rescind their mechanism to exploit any ex-post gains. We must stress, however, that in many situations the planner is a real person who can throw away resources (but cannot avoid agents' renegotiation when they have enough commodities).

of Welfare Economics, a is Pareto-efficient at s, and then $g(a,s)=a$. Let $\tilde{g}\in G^*$ and $s'\in S$ be such that $\tilde{g}(a,s')\notin W(s')$.

Claim 1. There exist $b\in A$ and $i\in I$ such that $aR_i(s)b$ and $bP_i(s')\tilde{g}(a,s')$.

We prove this claim in three steps.

Step 1.1. If Claim 1 is false then $\tilde{g}(a,s')\neq a$.

Suppose that $\tilde{g}(a,s')=a$. Let $p\in\mathbb{R}^l$ be a price vector that supports a as a constrained Walrasian equilibrium at s. Since $a\notin W(s')$, there are $i\in I$ and $b_i\neq a_i$ such that $b_i\leqslant\sum\omega_i$, $pb_i\leqslant p\omega_i$, and $b_iP_i(s')a_i$. Moreover, since (a,p) constitute a constrained Walrasian equilibrium at s, we have $a_iR_i(s)b_i$. Let $b\in A$ be a feasible allocation where agent i gets b_i. Then $aR_i(s)b$ and $bP_i(s')\tilde{g}(a,s')$. Therefore Claim 1 holds.

Step 1.2. If Claim 1 is false then $\tilde{g}(a,s')R_i(s)a$ for all $i\in I$.

Suppose that $a_iP_i(s)\tilde{g}_i(a,s')$ for some $i\in I$. Then, since preferences are strictly monotone, $\tilde{g}_i(a,s')\neq\sum\omega_i$. By continuity of preferences, there is $\epsilon>0$ such that, for all $b_i\in\mathbb{R}^l_+$ with $\|b_i-\tilde{g}_i(a,s')\|<\epsilon$, then $a_iP_i(s)b_i$. Moreover, since preferences are strictly monotone, there is $b_i\leqslant\sum\omega_i$ such that $\|b_i-\tilde{g}_i(a,s')\|<\epsilon$ and $b_iP_i(s')\tilde{g}_i(a,s')$. Therefore, Claim 1 holds.

Step 1.3. If Claim 1 is false then a is not Pareto-efficient at s.

Suppose that Claim 1 is false. Then, by Steps 1.1 and 1.2, and since preferences are strictly convex, $[\lambda\tilde{g}(a,s')+(1-\lambda)a]P_i(s)a$ for all $i\in I$ and all $\lambda\in(0,1)$, which contradicts the fact that a is Pareto-efficient at s.

Claim 2. There exist $y\in A$ and $i\in I$ such that $g(a,s)R_i(s)g(y,s)$ and $\tilde{g}(y,s')P_i(s')\times\tilde{g}(a,s')$.

Let $b\in A$ and $i\in I$ be as defined in Claim 1, and let $y=(0,\ldots,b_i,\ldots,0)$. Since only one agent has a positive amount of at least one good, no renegotiation is possible, and then $g(y,s)=\tilde{g}(y,s')=y$. Then, since $g(a,s)=a$, by Claim 1 $g(a,s)R_i(s)g(y,s)$ and $\tilde{g}(y,s')P_i(s')\tilde{g}(a,s')$. □

The results stated in Propositions 1 and 2 also hold for some other well-known social choice rules within the classical exchange environment, like the core correspondence or the Pareto-efficient and envy-free correspondence. The core correspondence, $C:S\to 2^A$, selects for each state s the subset of feasible allocations $a\in A$ such that there is no $I'\subseteq I$ and $b\in A$ satisfying (1) $\sum_{i\in I'}b_i\leqslant\sum_{i\in I'}\omega_i$, and (2) $bR_i(s)a$ for all $i\in I'$, with strict preference for some $i\in I'$. The Pareto-efficient and envy-free correspondence, $E:S\to 2^A$, selects for each state s the subset of feasible allocations $a\in A$ such that (1) there is no $b\in A$ such that $b_iR_i(s)a_i$ for all $i\in I$, with strict preference for some $i\in I$ (i.e., a is Pareto-efficient at s), and (2) $a_iR_i(s)a_j$ for all $i,j\in I$ (i.e., a is envy-free at s). The same example analyzed in Proposition 1 can be used to show that, if free-disposal is not allowed, neither C nor E are Nash implementable in G^* (note that in this example $a\in C(s)$ and, since $\omega P_1(s')g(a,s')$, then $g(a,s') \notin C(s')$; similarly, assuming that $\omega_1 = \omega_2$, we have $a \in E(s)$ and, since $g_2(a,s')P_1(s')g_1(a,s')$, then $g(a,s') \notin E(s')$). Furthermore, a proof analogous to the one of Proposition 2 can be used to show that if $n \geqslant 3$ and free-disposal is allowed, then both C and E satisfy the sufficient conditions for Nash implementation in G^*.

When analyzing implementation and renegotiation in other frameworks in which agents do not necessarily have commodities one can find conditions related to free-disposal. The idea is to sometimes prevent an agent from renegotiating the part of the outcome that goes to him by giving nothing to the rest (of course, this is only possible when the outcomes consist of a different element for each agent). For example, in a principal-agent environment where the outcomes are profiles of contracts (one for each agent), Amorós and Moreno (2001) propose a mechanism implementing the first-best choice rule[9] for any renegotiation function in a wide class that sometimes have to offer a contract to just one agent (so that he cannot renegotiate his contract with other agents).

5. Concluding remarks

In this paper, we have characterized Nash implementation when the outcomes of the mechanism can be renegotiated but the planner does not know the nature of the renegotiation process. We call this Nash implementation in G. We have shown that some well-known social choice rules in exchange environments are Nash implementable in G if and only if the planner can sometimes take away resources from the agents.

We see some scope for further development and extension of the model studied in this paper. One line of research could involve to study the case in which the true renegotiation function is unknown not only to the planner, but also to the agents (this situation can arise when the agents do not know each other's bargaining strengths when they play the mechanism). Another line of research could involve to extend our analysis to the case in which the no enforcement of the mechanism is due to individual rationality constraints.

Acknowledgments

I thank C. Beviá, A. Cabrales, L. Corchón, B. Dutta, B. Moreno, D. Perez-Castrillo, S. Puy, and an anonymous referee for their comments. I am also grateful to the seminar participants at Universitat Pompeu Fabra, SCW meeting in Pasadena, and Workshop on Social Choice and Welfare Economics in Málaga. Financial assistance from Fundación Ramón Areces and Ministerio de Ciencia y Tecnología under project SEC2002-01926 is gratefully acknowledged.

[9] In this setting, the first-best choice rule is a function that selects for each state s the profile of contracts that would maximize the profits of the principal if he knew the state.

References

Amorós, P., Moreno, B., 2001. Implementation of optimal contracts under adverse selection. Rev. Econ. Design 6, 41–61.

Baliga, S., Corchón, L., Sjöström, T., 1997. The theory of implementation when the planner is a player. J. Econ. Theory 77, 15–33.

Chakravorti, B., Corchón, L., Wilkie, S., 2003. Credible implementation. Games Econ. Behav. In press.

Hurwicz, L., 1994. Adjustment processes, mechanisms, and institutions. Rev. Econ. Design 1, 1–14.

Jackson, M.O., Palfrey, T., 1998. Efficiency and voluntary implementation in markets with repeated pairwise bargaining. Econometrica 66, 1353–1388.

Jackson, M.O., Palfrey, T., 2001. Voluntary implementation. J. Econ. Theory 98, 1–25.

Ma, C., Moore, J., Turnbull, S., 1988. Stopping agents from cheating. J. Econ. Theory 46, 355–372.

Maskin, E., 1999. Nash equilibrium and welfare optimality. Rev. Econ. Stud. 66, 23–38.

Maskin, E., Moore, J., 1999. Implementation and renegotiation. Rev. Econ. Stud. 66, 39–56.