



Munich Personal RePEc Archive

Traditional sufficient conditions for Nash implementation may fail on Internet

Wu, Haoyang

6 June 2012

Online at <https://mpra.ub.uni-muenchen.de/39268/>
MPRA Paper No. 39268, posted 06 Jun 2012 13:27 UTC

Traditional sufficient conditions for Nash implementation may fail on Internet

Haoyang Wu*

Abstract

The Maskin's theorem is a fundamental work in the theory of mechanism design. In this paper, we propose that if agents report messages to the designer through channels (*e.g.*, Internet), agents can construct a self-enforcing agreement such that any Pareto-inefficient social choice rule satisfying monotonicity and no-veto will not be Nash implementable when an additional condition is satisfied. The key points are: 1) The agreement is unobservable to the designer, and the designer cannot prevent the agents from constructing such agreement; 2) The agents act non-cooperatively, and the Maskin mechanism remain unchanged from the designer's perspective.

Key words: Mechanism design; Nash implementation; Social choice.

1 Introduction

The theory of mechanism design has been developed and applied to many branches of economics for decades. Nash implementation is a cornerstone of the mechanism design theory. The Maskin's theorem [1] provides an almost complete characterization of social choice rules (SCRs) that are Nash implementable: when the number of agents is at least three, the sufficient conditions for Nash implementation are monotonicity and no-veto, and the necessary condition is monotonicity. Note that an SCR is specified by a designer, a desired outcome from the designer's perspective may not be desirable for the agents (See Table 1 in Section 3.1).

The Maskin mechanism (page 394, [2]) constructed in the proof of Maskin's sufficiency theorem is an abstract mechanism. People seldom consider how the designer actually receives messages from agents. Roughly speaking, there

* Wan-Dou-Miao Research Lab, Suite 1002, 790 WuYi Road, Shanghai, 200051, China.

Email address: yangki76@163.com, Tel: 86-18621753457 (Haoyang Wu).

are two distinct manners: direct and indirect manner. In the former manner, agents report their messages to the designer directly (*e.g.*, speak face to face, hand over, *etc*), thereby the designer can know exactly that a message is reported by an agent himself, not by any other device. In the latter manner, agents report messages to the designer through channels (*e.g.*, Internet, cable *etc*). Thereby, when the designer receives a message from a channel, he cannot know what has happened on the other side of the channel: whether the message is reported by an agent himself, or generated by some device authorized by an agent.

Traditionally, nobody notice the difference between the two manners in the Maskin mechanism. However, in this paper, we will point out that traditional sufficient conditions on Nash implementation may fail if agents report messages to the designer in an indirect manner. The rest of the paper is organized as follows: Section 2 recalls preliminaries of the mechanism design theory given by Serrano [2]; Section 3 is the main part of this paper, where we will propose a self-enforcing agreement to help agents break through the restriction of Maskin's sufficiency theorem. Section 4 draws the conclusion.

2 Preliminaries

Let $N = \{1, \dots, n\}$ be a finite set of *agents* with $n \geq 3$, $A = \{a^1, a^2, \dots\}$ be a finite set of social *outcomes*. The information held by the agents is summarized in the concept of a *state*. The true state is not verifiable by the designer. We denote by t a typical state and by \mathcal{T} the domain of possible states. At state $t \in \mathcal{T}$, each agent $j \in N$ is assumed to have a complete and transitive *preference relation* \succeq_j^t over the set A . We denote by $\succeq^t = (\succeq_1^t, \dots, \succeq_n^t)$ the profile of preferences in state t , and denote by \succ_j^t the strict preference part of \succeq_j^t .

Fix a state t , we refer to the collection $E = \langle N, A, (\succeq_j^t)_{j \in N} \rangle$ as an *environment*. Let ε be the class of possible environments. A *social choice rule* (SCR) F is a mapping $F : \varepsilon \rightarrow 2^A \setminus \{\emptyset\}$. A *mechanism* $\Gamma = ((M_j)_{j \in N}, g)$ describes a message or strategy set M_j for agent j , and an outcome function $g : \prod_{j \in N} M_j \rightarrow A$. M_j is unlimited except that if a mechanism is direct, *i.e.*, $M_j = T_j$.

An SCR F satisfies *no-veto* if, whenever $a \succeq_j^t b$ for all $b \in A$ and for every agent j but perhaps one k , then $a \in F(E)$. An SCR F is *monotonic* if for every pair of environments E and E' , and for every $a \in F(E)$, whenever $a \succeq_j^t b$ implies that $a \succeq_j^{t'}$ b , there holds $a \in F(E')$. We assume that there is *complete information* among the agents, *i.e.*, the true state t is common knowledge among them. Given a mechanism $\Gamma = ((M_j)_{j \in N}, g)$ played in state t , a *Nash equilibrium* of Γ in state t is a strategy profile m^* such that:

$\forall j \in N, g(m^*(t)) \succeq_j^t g(m_j, m_{-j}^*(t)), \forall m_j \in M_j$. Let $\mathcal{N}(\Gamma, t)$ denote the set of Nash equilibria of the game induced by Γ in state t , and $g(\mathcal{N}(\Gamma, t))$ denote the corresponding set of Nash equilibrium outcomes. An SCR F is *Nash implementable* if there exists a mechanism $\Gamma = ((M_j)_{j \in N}, g)$ such that for every $t \in \mathcal{T}$, $g(\mathcal{N}(\Gamma, t)) = F(t)$.

Maskin [1] provided an almost complete characterization of SCRs that were Nash implementable. The main results of Ref. [1] are two theorems: 1) (*Necessity*) If an SCR is Nash implementable, then it is monotonic. 2) (*Sufficiency*) Let $n \geq 3$, if an SCR is monotonic and satisfies no-veto, then it is Nash implementable. In order to facilitate the following investigation, we briefly recall the Maskin mechanism given by Serrano [2] as follows:

Consider a mechanism $\Gamma = ((M_j)_{j \in N}, g)$, where agent j 's message set is $M_j = A \times \mathcal{T} \times \mathbb{Z}_+$, \mathbb{Z}_+ is the set of non-negative integers. A typical message sent by agent j is described as $m_j = (a_j, t_j, z_j)$. The outcome function g is defined in the following three rules: (1) If for every agent $j \in N$, $m_j = (a, t, 0)$ and $a \in F(t)$, then $g(m) = a$. (2) If $(n - 1)$ agents $j \neq k$ send $m_j = (a, t, 0)$ and $a \in F(t)$, but agent k sends $m_k = (a_k, t_k, z_k) \neq (a, t, 0)$, then $g(m) = a$ if $a_k \succ_k^t a$, and $g(m) = a_k$ otherwise. (3) In all other cases, $g(m) = a'$, where a' is the outcome chosen by the agent with the lowest index among those who announce the highest integer.

3 Nash implementation on Internet

This section is the main part of this paper. In the beginning, we will show an example of SCR which satisfies monotonicity and no-veto. It is Nash implementable although all agents dislike it. Then, we will propose a self-enforcing agreement using complex numbers, by which the agents may break through the Maskin's sufficiency theorem and make the SCR not Nash implementable.

3.1 A Pareto-inefficient SCR that satisfies monotonicity and no-veto

Let $N = \{Apple, Lily, Cindy\}$, $\mathcal{T} = \{t^1, t^2\}$, $A = \{a^1, a^2, a^3, a^4\}$. In each state $t \in \mathcal{T}$, the preference relations $(\succeq_j^t)_{j \in N}$ over the outcome set A and the corresponding SCR F are given in Table 1. The SCR F is *Pareto-inefficient* from the agents' perspectives because in state t^2 , all agents unanimously prefer a Pareto-optimal outcome a^1 : for each agent $j \in N$, $a^1 \succ_j^{t^2} a^2 \in F(t^2)$.

Table 1: An SCR satisfying monotonicity and no-veto is Pareto-inefficient from the agents' perspectives.

State t^1			State t^2		
<i>Apple</i>	<i>Lily</i>	<i>Cindy</i>	<i>Apple</i>	<i>Lily</i>	<i>Cindy</i>
a^3	a^2	a^1	a^4	a^3	a^1
a^1	a^1	a^3	a^1	a^1	a^2
a^2	a^4	a^2	a^2	a^2	a^3
a^4	a^3	a^4	a^3	a^4	a^4
$F(t^1) = \{a^1\}$			$F(t^2) = \{a^2\}$		

Suppose the true state is t^2 . At first sight, $(a^1, t^1, 0)$ might be a unanimous m_j for each agent j , because by doing so a^1 would be generated by rule 1 of the Maskin mechanism. However, *Apple* has an incentive to unilaterally deviate from $(a^1, t^1, 0)$ to $(a^4, *, *)$ in order to trigger rule 2 (where $*$ stands for any legal value), since $a^1 \succ_{Apple}^{t^1} a^4$, $a^4 \succ_{Apple}^{t^2} a^1$; *Lily* also has an incentive to unilaterally deviate from $(a^1, t^1, 0)$ to $(a^3, *, *)$, since $a^1 \succ_{Lily}^{t^1} a^3$, $a^3 \succ_{Lily}^{t^2} a^1$.

Note that either *Apple* or *Lily* can certainly obtain her expected outcome only if just one of them deviates from $(a^1, t^1, 0)$ (If this case happened, rule 2 would be triggered). But this condition is unreasonable, because all agents are rational, nobody is willing to give up and let the others benefit. Therefore, both *Apple* and *Lily* will deviate from $(a^1, t^1, 0)$. As a result, rule 3 will be triggered. Since *Apple* and *Lily* both have a chance to win the integer game, the winner is uncertain and the final outcome is also uncertain between a^3 and a^4 .

To sum up, although every agent prefers a^1 to a^2 in state t^2 , a^1 cannot be yielded in Nash equilibrium. Indeed, the Maskin mechanism makes the Pareto-inefficient outcome a^2 be Nash implementable in state t^2 .

Can the agents find a way to break through the Maskin's sufficiency theorem and let the Pareto-efficient outcome a^1 be Nash implementable in state t^2 ? Interestingly, we will show that the answer may be "yes" if agents report messages to the designer through channels (*e.g.*, Internet). In what follows, first we will define some matrices with complex numbers, then we will propose a self-enforcing agreement to help agents break through the Maskin's sufficiency theorem.

3.2 Definitions

Definition 1: Let $\hat{I}, \hat{\sigma}$ be two 2×2 matrices, and \vec{C}, \vec{D} be two basis vectors:

$$\hat{I} \equiv \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \hat{\sigma} \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \vec{C} \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \vec{D} \equiv \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (1)$$

Hence, $\hat{I}\vec{C} = \vec{C}$, $\hat{I}\vec{D} = \vec{D}$; $\hat{\sigma}\vec{C} = \vec{D}$, $\hat{\sigma}\vec{D} = \vec{C}$.

Definition 2: For $n \geq 3$ agents, suppose each agent $j \in N$ possesses a basis vector. $\vec{\psi}_0$ is defined as the tensor product of n basis vectors \vec{C} :

$$\vec{\psi}_0 \equiv \vec{C}^{\otimes n} \equiv \underbrace{\vec{C} \otimes \dots \otimes \vec{C}}_n \equiv \begin{bmatrix} 1 \\ 0 \\ \dots \\ 0 \end{bmatrix}_{2^n \times 1} \quad (2)$$

$\vec{C}^{\otimes n}$ contains n basis vectors \vec{C} and 2^n elements. $\vec{C}^{\otimes n}$ is also denoted as $\overrightarrow{C \dots C C^n}$. Similarly,

$$\overrightarrow{C \dots C D^n} \equiv \underbrace{\vec{C} \otimes \dots \otimes \vec{C}}_{n-1} \otimes \vec{D} = \begin{bmatrix} 0 \\ 1 \\ \dots \\ 0 \end{bmatrix}_{2^n \times 1} \quad (3)$$

Obviously, there are 2^n possible vectors: $\overrightarrow{C \dots C C^n}, \dots, \overrightarrow{D \dots D D^n}$.

Definition 3: $\hat{J} \equiv \frac{1}{\sqrt{2}}(\hat{I}^{\otimes n} + i\hat{\sigma}^{\otimes n})$, i.e.,

$$\hat{J} \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & & & i \\ \dots & & & \dots \\ & 1 & i & \\ & i & 1 & \\ \dots & & & \dots \\ i & & & 1 \end{bmatrix}_{2^n \times 2^n}, \quad \hat{J}^+ \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & & & -i \\ \dots & & & \dots \\ & 1 & -i & \\ & -i & 1 & \\ \dots & & & \dots \\ -i & & & 1 \end{bmatrix}_{2^n \times 2^n} \quad (4)$$

where the symbol i denotes an imaginary number, and \hat{J}^+ is the conjugate transpose of \hat{J} .

Definition 4:

$$\vec{\psi}_1 \equiv \hat{J}\vec{\psi}_0 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ \dots \\ 0 \\ i \end{bmatrix}_{2^n \times 1} \quad (5)$$

Definition 5: For $\theta \in [0, \pi]$, $\phi \in [0, \pi/2]$,

$$\hat{\omega}(\theta, \phi) \equiv \begin{bmatrix} e^{i\phi} \cos(\theta/2) & i \sin(\theta/2) \\ i \sin(\theta/2) & e^{-i\phi} \cos(\theta/2) \end{bmatrix}. \quad (6)$$

$\hat{\Omega} \equiv \{\hat{\omega}(\theta, \phi) : \theta \in [0, \pi], \phi \in [0, \pi/2]\}$. Hence, $\hat{I} = \hat{\omega}(0, 0)$.

Definition 6: For $j = 1, \dots, n$, $\theta_j \in [0, \pi]$, $\phi_j \in [0, \pi/2]$, let $\hat{\omega}_j = \hat{\omega}(\theta_j, \phi_j)$,

$$\vec{\psi}_2 \equiv [\hat{\omega}_1 \otimes \dots \otimes \hat{\omega}_n] \vec{\psi}_1. \quad (7)$$

The dimension of $\hat{\omega}_1 \otimes \dots \otimes \hat{\omega}_n$ is $2^n \times 2^n$. Since only two elements in $\vec{\psi}_1$ are non-zero, it is not necessary to calculate the whole $2^n \times 2^n$ matrix to yield $\vec{\psi}_2$. Indeed, we only need to calculate the leftmost and rightmost column of $\hat{\omega}_1 \otimes \dots \otimes \hat{\omega}_n$ to derive $\vec{\psi}_2$.

Definition 7: $\vec{\psi}_3 \equiv \hat{J}^+ \vec{\psi}_2$.

Suppose $\vec{\psi}_3 = [\eta_1, \dots, \eta_{2^n}]^T$, let $\Delta = [|\eta_1|^2, \dots, |\eta_{2^n}|^2]$. It can be easily checked that \hat{J} , $\hat{\omega}_j$ ($j = 1, \dots, n$) and \hat{J}^+ are all unitary matrices. Hence, $|\vec{\psi}_3|^2 = 1$. Thus, Δ can be viewed as a probability distribution, each element of which represents the probability that we randomly choose a vector from the set of all 2^n possible vectors $\{\overrightarrow{C \dots C C^n}, \dots, \overrightarrow{D \dots D D^n}\}$.

Definition 8: Condition λ contains five parts. The first three parts are defined as follows:

λ_1 : Given an SCR F , there exist two states $\hat{t}, \bar{t} \in \mathcal{T}$, $\hat{t} \neq \bar{t}$ such that $\hat{a} \succeq_{\bar{t}} \bar{a}$ (for each $j \in N$, $\hat{a} \in F(\hat{t})$, $\bar{a} \in F(\bar{t})$) with strict relation for some agent; and the number of agents that encounter a preference change around \hat{a} in going from state \hat{t} to \bar{t} is at least two. Denote by \hat{N} the set of these agents and $l = |\hat{N}|$ the number of these agents.

λ_2 : Consider the state \bar{t} specified in condition λ_1 , if there exists another $\hat{t}' \in \mathcal{T}$,

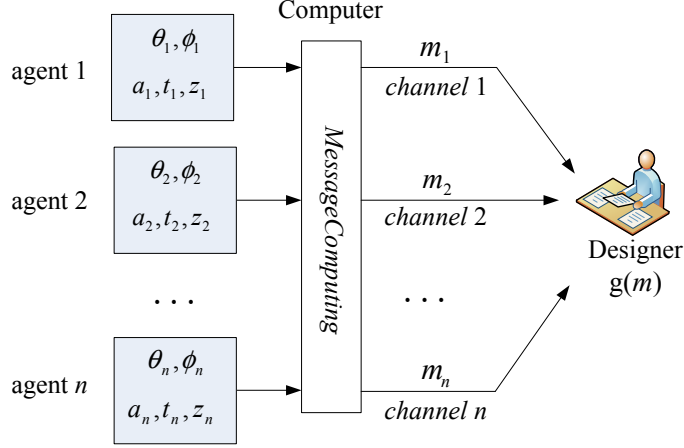


Fig. 1. The set-up of the agreement *ComplexMessage*. The algorithm *MessageComputing* is given in Definition 9.

$\hat{t}' \neq \hat{t}$ that satisfies λ_1 , then $\hat{a} \succeq_j^{\bar{t}} \hat{a}'$ (for each $j \in N$, $\hat{a} \in F(\hat{t})$, $\hat{a}' \in F(\hat{t}')$) with strict relation for some agent.

λ_3 : Consider the outcome \hat{a} specified in condition λ_1 , for any state $t \in \mathcal{T}$, \hat{a} is top ranked for each agent $j \in N \setminus \hat{N}$.

3.3 An agreement using complex numbers

As specified before, in this paper we will discuss how the traditional results on Nash implementation will be changed when agents report messages to the designer through channels. We assume that:

- 1) Each agent has a unique channel connected with the designer. The agents report messages to the designer through these channels.
- 2) After the designer claims the outcome function g , if all agents anticipate the SCR F appeared in rule 1 of g satisfies condition λ (given in Definition 8 and 11), agents can negotiate and construct an agreement *ComplexMessage* as shown in Fig. 1. The algorithm *MessageComputing* is given in Definition 9. In the initial configuration, the computer controls all channels.
- 3) Each agent can freely decide whether to leave his channel to the computer and let the computer send a message to the designer, or to take back his channel and send his message to the designer by himself. If any agent deviates from the initial configuration and chooses the latter option, then this deviation is *observable* to the rest agents and all of them will choose the latter option too, *i.e.*, all agents will take back their channels and send their messages to the designer by themselves. If all agents leave their channels to the computer, the algorithm *MessageComputing* works, *i.e.*, calculates m_1, \dots, m_n and sends them to the designer through channels.

Definition 9: The algorithm *MessageComputing* is defined as follows:

Input: $(\theta_j, \phi_j, a_j, t_j, z_j) \in [0, \pi/2] \times [0, \pi] \times A \times \mathcal{T} \times \mathbb{Z}_+, j = 1, \dots, n.$

Output: $m_j \in A \times \mathcal{T} \times \mathbb{Z}_+, j = 1, \dots, n.$

Step 1: Reading (θ_j, ϕ_j) from each agent $j \in N.$

Step 2: Computing the leftmost and rightmost columns of $\hat{\omega}_1 \otimes \dots \otimes \hat{\omega}_n.$

Step 3: Computing $\vec{\psi}_2 = [\hat{\omega}_1 \otimes \dots \otimes \hat{\omega}_n] \vec{\psi}_1, \vec{\psi}_3 = \hat{J}^+ \vec{\psi}_2,$ and the probability distribution $\Delta.$

Step 4: Randomly choosing a vector from the set of all 2^n possible vectors $\{\vec{C} \dots \vec{C} \vec{C}^n, \dots, \vec{D} \dots \vec{D} \vec{D}^n\}$ according to the probability distribution $\Delta.$

Step 5: For each agent $j \in N,$ let $m_j = (\hat{a}, \hat{t}, 0)$ (or $m_j = (a_j, t_j, z_j)$) if the j -th basis vector of the chosen vector is \vec{C} (or \vec{D}), where \hat{a}, \hat{t} are specified in condition $\lambda_1.$

Step 6: Sending $m = (m_1, \dots, m_n)$ to the designer through channels $1, \dots, n.$

Definition 10: Suppose λ_1 and λ_2 are satisfied and $m = (m_1, \dots, m_n)$ is computed by *MessageComputing*. Suppose the true state is \bar{t} specified in condition $\lambda_1.$ $\$_{C \dots CC}, \$_{C \dots CD}, \$_{D \dots DC}$ and $\$_{D \dots DD}$ are defined as the payoffs to the n -th agent in state \bar{t} when the chosen vector in Step 4 of *MessageComputing* is $\vec{C} \dots \vec{C} \vec{C}^n, \vec{C} \dots \vec{C} \vec{D}^n, \vec{D} \dots \vec{D} \vec{C}^n$ or $\vec{D} \dots \vec{D} \vec{D}^n$ respectively.

Definition 11: Suppose conditions λ_1, λ_2 and λ_3 are satisfied and the true state is $\bar{t},$ consider each message $m_j = (a_j, t_j, z_j),$ where a_j is top-ranked for each agent $j.$ The rest two parts of condition λ are defined as:

$\lambda_4:$ For each agent $j \in N,$ let him be the n -th agent, $\$_{C \dots CC} > \$_{D \dots DD}.$

$\lambda_5:$ For each agent $j \in \hat{N},$ let him be the n -th agent, $\$_{C \dots CC} > \$_{C \dots CD} \cos^2(\pi/l) + \$_{D \dots DC} \sin^2(\pi/l).$

It can be seen from Fig. 1 that after the agreement *ComplexMessage* is constructed, each agent $j \in N$ independently faces two options:

- $S(j, 0):$ leaving his channel to the computer, and submitting $(\theta_j, \phi_j, a_j, t_j, z_j)$ to the algorithm *MessageComputing*.
- $S(j, 1):$ taking back his channel, and submitting (a_j, t_j, z_j) to the designer by himself.

To sum up, if agents report messages to the designer through channels and all agents anticipate the SCR F appeared in rule 1 of g satisfies condition $\lambda,$ they can construct the agreement *ComplexMessage* after the designer claims the outcome function $g.$ The timing steps of the Maskin mechanism are updated as follows:

Time 1: The designer claims the outcome function g to all agents;

Time 2: The agents construct the agreement *ComplexMessage*;

Time 3: Each agent $j \in N$ chooses an option between $S(j, 0)$ and $S(j, 1),$ and m_1, \dots, m_n are sent through channels.

Time 4: The designer receives $m = (m_1, \dots, m_n)$ from n channels;

Time 5: The designer announces the outcome $g(m)$.

3.4 Main result

Proposition 1: For $n \geq 3$, suppose agents send messages to the designer through channels (*e.g.*, Internet). Consider an SCR F that satisfies monotonicity and no-veto. If condition λ is satisfied, then in state \bar{t} the agents can construct a self-enforcing agreement *ComplexMessage* to make the Pareto-inefficient outcome $F(\bar{t})$ not be implemented in Nash equilibrium.

Proof: Since λ_1 and λ_2 are satisfied, then there exist two states $\hat{t}, \bar{t} \in \mathcal{T}$, $\hat{t} \neq \bar{t}$ such that $\hat{a} \succeq_j^{\bar{t}} \bar{a}$ (for each $j \in N$, $\hat{a} \in F(\hat{t})$, $\bar{a} \in F(\bar{t})$) with strict relation for some agent, and \hat{N} contains the agents that encounter a preference change around \hat{a} in going from state \hat{t} to \bar{t} ($l = |\hat{N}| \geq 2$). Suppose the true state is \bar{t} , now let us check what will happen after the agents construct the agreement *ComplexMessage* in Time 2.

From the viewpoints of agents, after constructing *ComplexMessage*, there are two possible cases in Time 3:

1) Suppose every agent $j \in N$ chooses $S(j, 0)$, then the algorithm *MessageComputing* works. Consider the following strategy profile: each agent $j \in N \setminus \hat{N}$ submits $(\theta_j, \phi_j) = (0, 0)$, $(a_j, t_j, z_j) = (\hat{a}, \hat{t}, 0)$; each agent $j \in \hat{N}$ submits $(\theta_j, \phi_j) = (0, \pi/l)$. According to Lemma 1 (see Appendix), this strategy profile is a Nash equilibrium of Γ in state \bar{t} . In Step 4 of *MessageComputing*, the chosen vector is $\overrightarrow{C \cdots CC}$. In Step 5 of *MessageComputing*, $m_j = (\hat{a}, \hat{t}, 0)$ for each $j \in N$. Therefore, in Time 5, $g(m) = \hat{a} \notin F(\bar{t})$. Each agent j 's payoff is $\$_{C \cdots CC}$.

2) Suppose some agent $j \in N$ chooses $S(j, 1)$, *i.e.*, takes back his channel and reports m_j to the designer by himself. Then according to assumption 3 (see Section 3.3), all of the rest agents will observe this deviation, thereby take back their channels and submit messages to the designer by themselves. According the Maskin mechanism, in Time 5, the final outcome will be either $F(\bar{t})$ by rule 1, or uncertain by rule 3 (*i.e.*, each agent j 's payoff is $\$_{D \cdots DD}$).

According to conditions λ_1 and λ_4 , it is not profitable for any agent j to choose $S(j, 1)$, *i.e.*, unilaterally take back his channel and send a message to the designer by himself. As Telser pointed out in page 28, Line 2 [3]: “A party to a self-enforcing agreement calculates whether his gain from violating the agreement is greater or less than the loss of future net benefits that he would incur as a result of detection of his violation and the consequent termination of the agreement by the other party... Hence both parties continue to adhere to an agreement if and only if each gains more from adherence to, than from violations of, its terms.” Therefore, it can be seen that *ComplexMessage* is

a self-enforcing agreement among the agents. Put differently, although the agents collaborate to construct *ComplexMessage* in Time 2, they do not require a third-party to enforce it after then and the game is still non-cooperative.

To sum up, in state \bar{t} , the agents can construct a self-enforcing agreement *ComplexMessage* to make the Pareto-inefficient outcome $F(\bar{t})$ not be implemented in Nash equilibrium. \square

Let us reconsider Table 1. Let $\hat{t} = t^1$, $\hat{a} = a^1$, $\bar{t} = t^2$, $\bar{a} = a^2$. Suppose the true state is t^2 . Since both *Apple* and *Lily* encounter a preference change around a^1 in going from state t^1 to t^2 , condition λ_1 is satisfied. Obviously, λ_2 and λ_3 are also satisfied. Consider the strategy profile as follows:

$$\begin{aligned} (\theta_{Apple}, \phi_{Apple}) &= (0, \pi/2), & (a_{Apple}, t_{Apple}, z_{Apple}) &= (a^4, *, *); \\ (\theta_{Lily}, \phi_{Lily}) &= (0, \pi/2), & (a_{Lily}, t_{Lily}, z_{Lily}) &= (a^3, *, *); \\ (\theta_{Cindy}, \phi_{Cindy}) &= (0, 0), & (a_{Cindy}, t_{Cindy}, z_{Cindy}) &= (a^1, t^1, 0). \end{aligned}$$

Let *Cindy* be the first agent, and for any agent $j \in \{Apple, Lily\}$, let her be the last agent. Consider the payoff to the third agent, suppose $\$_{CCC} = 3$ (the corresponding outcome is a^1), $\$_{CCD} = 5$ (the corresponding outcome is a^4 if $j = Apple$, and a^3 if $j = Lily$), $\$_{DDC} = 0$ (the corresponding outcome is a^3 if $j = Apple$, and a^4 if $j = Lily$), $\$_{DDD} = 1$ (the corresponding outcome is uncertain between a^3 and a^4). Let *Cindy* be the last agent and consider her payoff, suppose $\$_{CCC} = 3$ and $\$_{DDD} = 1$. Hence, λ_4 and λ_5 are satisfied. According to Proposition 1, in state t^2 , the outcome implemented in Nash equilibrium is a^1 , and $F(t^2)$ is not Nash implementable although the SCR F satisfies monotonicity and no-veto.

Remark 1: Some reader may argue that the agreement *ComplexMessage* is a wrapper to the Maskin mechanism that changes the game substantially and henceforth it has no implication on the original Maskin mechanism. However, this viewpoint is not true. Actually, *ComplexMessage* is *unobservable* to the designer because it is hidden behind channels and the designer cannot prevent the agents from constructing such agreement. From the designer's perspective, no matter whether the agents construct the agreement *ComplexMessage* on the other side of channels or not, the Maskin mechanism remain unchanged and the designer acts in the same way: *i.e.*, claims the outcome function g , receives messages $m = (m_1, \dots, m_n)$ from channels and announces the final outcome $g(m)$.

Remark 2: Although the time and space complexity of *MessageComputing* are exponential with the number of agents, *i.e.*, $O(2^n)$, it works well when the number of agents is not very large. For example, the runtime of *MessageComputing* is about 0.5s for 15 agents, and about 12s for 20 agents (MATLAB 7.1, CPU: Intel (R) 2GHz, RAM: 3GB).

Remark 3: The problem of Nash implementation requires complete information among all agents. In the last paragraph of Page 392 [2], Serrano wrote: “*We assume that there is complete information among the agents... This assumption is especially justified when the implementation problem concerns a small number of agents that hold good information about one another*”. Hence, the fact that *MessageComputing* is suitable for small-scale cases (*e.g.*, less than 20 agents) is acceptable for Nash implementation.

4 Conclusion

In this paper, we propose a self-enforcing agreement to help agents avoid a Pareto-inefficient social choice rule if agents report messages to the designer through channels and condition λ is satisfied. Put differently, traditional sufficient conditions for Nash implementation may fail on Internet. With the rapid development of network economics, it will be more and more common that agents communicate with the designer through Internet. In the future, there are many works to do to study the self-enforcing agreement further.

Acknowledgments

The author is very grateful to Ms. Fang Chen, Hanyue Wu (*Apple*), Hanxing Wu (*Lily*) and Hanchen Wu (*Cindy*) for their great support.

References

- [1] E. Maskin, Nash equilibrium and welfare optimality, *Rev. Econom. Stud.* **66** (1999) 23-38.
- [2] R. Serrano, The theory of implementation of social choice rules, *SIAM Review* **46** (2004) 377-414.
- [3] L.G. Telser, A theory of self-enforcing agreements. *Journal of Business* **53** (1980) 27-44.
- [4] A.P. Flitney and L.C.L. Hollenberg, Nash equilibria in quantum games with generalized two-parameter strategies, *Phys. Lett. A* **363** (2007) 381-388.

Appendix

Lemma 1: Suppose condition λ is satisfied and the algorithm *MessageComputing* works. Consider the following strategy profile:

- 1) Each agent $j \in N \setminus \hat{N}$ submits $(\theta_j, \phi_j) = (0, 0)$, $(a_j, t_j, z_j) = (\hat{a}, \hat{t}, 0)$;
- 2) Each agent $j \in \hat{N}$ submits $(\theta_j, \phi_j) = (0, \pi/l)$;

then this strategy profile is a Nash equilibrium of Γ in state \bar{t} , where \bar{t} is specified in condition λ_1 .

Proof: The proof consists of two parts.

Part 1. Let the last l agents be \hat{N} . Consider the following strategy profile: each agent $j = 1, \dots, (n-l)$ submits $(\theta_j, \phi_j) = (0, 0)$, $(a_j, t_j, z_j) = (\hat{a}, \hat{t}, 0)$; each agent $j = (n-l+1), \dots, (n-1)$ submits $(\theta_j, \phi_j) = (0, \pi/l)$, then we will prove the optimal value of (θ_n, ϕ_n) for the n -th agent is $(0, \pi/l)$.

Since condition λ_1 is satisfied, then $l \geq 2$. Let

$$\hat{C}_l \equiv \hat{\omega}(0, \pi/l) = \begin{bmatrix} e^{i\frac{\pi}{l}} & 0 \\ 0 & e^{-i\frac{\pi}{l}} \end{bmatrix}_{2 \times 2}, \quad \text{thus, } \hat{C}_l \otimes \hat{C}_l = \begin{bmatrix} e^{i\frac{2\pi}{l}} & & & \\ & 1 & & \\ & & 1 & \\ & & & e^{-i\frac{2\pi}{l}} \end{bmatrix}_{2^2 \times 2^2},$$

$$\underbrace{\hat{C}_l \otimes \dots \otimes \hat{C}_l}_{l-1} = \begin{bmatrix} e^{i\frac{(l-1)\pi}{l}} & & & \\ & * & & \\ & & \dots & \\ & & & e^{-i\frac{(l-1)\pi}{l}} \end{bmatrix}_{2^{l-1} \times 2^{l-1}}.$$

Here we only explicitly show the up-left and bottom-right entries because only these two entries are useful in the following discussions. The other entries in diagonal are simply represented as symbol $*$. Note that

$$\underbrace{\hat{I} \otimes \dots \otimes \hat{I}}_{n-l} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \dots & \\ & & & 1 \end{bmatrix}_{2^{n-l} \times 2^{n-l}},$$

thus,

$$\underbrace{\hat{I} \otimes \cdots \otimes \hat{I}}_{n-l} \otimes \underbrace{\hat{C}_l \otimes \cdots \otimes \hat{C}_l}_{l-1} = \begin{bmatrix} e^{i\frac{(l-1)\pi}{l}} & & & \\ & * & & \\ & & \cdots & \\ & & & e^{-i\frac{(l-1)\pi}{l}} \end{bmatrix}_{2^{n-1} \times 2^{n-1}} .$$

Suppose the n -th agent chooses arbitrary parameters (θ, ϕ) in his strategy $(\theta, \phi, a_n, t_n, z_n)$, let

$$\hat{\omega}_n(\theta, \phi) = \begin{bmatrix} e^{i\phi} \cos(\theta/2) & i \sin(\theta/2) \\ i \sin(\theta/2) & e^{-i\phi} \cos(\theta/2) \end{bmatrix},$$

then,

$$\underbrace{\hat{I} \otimes \cdots \otimes \hat{I}}_{n-l} \otimes \underbrace{\hat{C}_l \otimes \cdots \otimes \hat{C}_l}_{l-1} \otimes \hat{\omega}_n(\theta, \phi) = \begin{bmatrix} e^{i[\frac{(l-1)\pi}{l} + \phi]} \cos(\theta/2) * & & & & & & & & & & \\ & i e^{i\frac{(l-1)\pi}{l}} \sin(\theta/2) * & & & & & & & & & \\ & & * * & & & & & & & & \\ & & * * & & & & & & & & \\ & & & \cdots & & & & & & & \\ & & & & & * & i e^{-i\frac{(l-1)\pi}{l}} \sin(\theta/2) & & & & \\ & & & & & * & e^{-i[\frac{(l-1)\pi}{l} + \phi]} \cos(\theta/2) & & & & \end{bmatrix}_{2^n \times 2^n} .$$

Recall that

$$\vec{\psi}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ \cdots \\ 0 \\ i \end{bmatrix}_{2^n \times 1},$$

thus,

$$\vec{\psi}_2 = \underbrace{[\hat{I} \otimes \cdots \otimes \hat{I}]_{n-l}} \otimes \underbrace{[\hat{C}_l \otimes \cdots \otimes \hat{C}_l]_{l-1}} \otimes \hat{\omega}_n(\theta, \phi) \vec{\psi}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} e^{i[\frac{(l-1)\pi}{l} + \phi]} \cos(\theta/2) \\ ie^{i\frac{(l-1)\pi}{l}} \sin(\theta/2) \\ 0 \\ \cdots \\ 0 \\ -e^{-i\frac{(l-1)\pi}{l}} \sin(\theta/2) \\ ie^{-i[\frac{(l-1)\pi}{l} + \phi]} \cos(\theta/2) \end{bmatrix}_{2^n \times 1},$$

$$\vec{\psi}_3 = \hat{J}^+ \vec{\psi}_2 = \begin{bmatrix} \cos(\theta/2) \cos(\frac{l-1}{l}\pi + \phi) \\ i \sin(\theta/2) \cos \frac{l-1}{l}\pi \\ 0 \\ \cdots \\ 0 \\ i \sin(\theta/2) \sin \frac{l-1}{l}\pi \\ \cos(\theta/2) \sin(\frac{l-1}{l}\pi + \phi) \end{bmatrix}_{2^n \times 1}$$

The probability distribution Δ is computed from $\vec{\psi}_3$:

$$\begin{aligned} P_{C\dots CC} &= \cos^2(\theta/2) \cos^2(\phi - \frac{\pi}{l}) \\ P_{C\dots CD} &= \sin^2(\theta/2) \cos^2 \frac{\pi}{l} \\ P_{D\dots DC} &= \sin^2(\theta/2) \sin^2 \frac{\pi}{l} \\ P_{D\dots DD} &= \cos^2(\theta/2) \sin^2(\phi - \frac{\pi}{l}) \end{aligned}$$

Obviously,

$$P_{C\dots CC} + P_{C\dots CD} + P_{D\dots DC} + P_{D\dots DD} = 1.$$

Consider the payoff to the n -th agent,

$$\mathcal{S}_n = \mathcal{S}_{C\dots CC} P_{C\dots CC} + \mathcal{S}_{C\dots CD} P_{C\dots CD} + \mathcal{S}_{D\dots DC} P_{D\dots DC} + \mathcal{S}_{D\dots DD} P_{D\dots DD}.$$

Since λ_4 is satisfied, *i.e.*, $\mathcal{S}_{C\dots CC} > \mathcal{S}_{D\dots DD}$, then the n -th agent chooses $\phi = \pi/l$ to minimize $\sin^2(\phi - \frac{\pi}{l})$. As a result, $P_{C\dots CC} = \cos^2(\theta/2)$. Since λ_5 is satisfied, *i.e.*, $\mathcal{S}_{C\dots CC} > \mathcal{S}_{C\dots CD} \cos^2(\pi/l) + \mathcal{S}_{D\dots DC} \sin^2(\pi/l)$, then the n -th agent prefers $\theta = 0$, which leads to $P_{C\dots CC} = 1$ and let \mathcal{S}_n be its maximum $\mathcal{S}_{C\dots CC}$. Therefore, the optimal value of (θ_n, ϕ_n) for the n -th agent is $(0, \pi/l)$.

Part 2. Let the first l agents be \hat{N} . Consider the following strategy profile: each agent $j = 1, \dots, l$ submits $(\theta_j, \phi_j) = (0, \pi/l)$; each agent $j = (l+1), \dots, (n-l+1)$ submits $(\theta_j, \phi_j) = (0, 0)$, $(a_j, t_j, z_j) = (\hat{a}, \hat{t}, 0)$, then we will prove the optimal values of (θ_n, ϕ_n) and (a_n, t_n, z_n) for the n -th agent are $(0, 0)$ and $(\hat{a}, \hat{t}, 0)$.

As shown before,

$$\hat{C}_l = \hat{\omega}(0, \pi/l) = \begin{bmatrix} e^{i\frac{\pi}{l}} & 0 \\ 0 & e^{-i\frac{\pi}{l}} \end{bmatrix}_{2 \times 2}, \quad \hat{C}_l \otimes \hat{C}_l = \begin{bmatrix} e^{i\frac{2\pi}{l}} & & & \\ & 1 & & \\ & & 1 & \\ & & & e^{-i\frac{2\pi}{l}} \end{bmatrix}_{2^2 \times 2^2},$$

$$\underbrace{\hat{C}_l \otimes \dots \otimes \hat{C}_l}_l = \begin{bmatrix} -1 & & & \\ & * & & \\ & & \dots & \\ & & & -1 \end{bmatrix}_{2^l \times 2^l}.$$

Note that

$$\underbrace{\hat{I} \otimes \dots \otimes \hat{I}}_{n-l-1} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \dots & \\ & & & 1 \end{bmatrix}_{2^{n-l-1} \times 2^{n-l-1}},$$

thus,

$$\underbrace{\hat{C}_l \otimes \dots \otimes \hat{C}_l}_l \otimes \underbrace{\hat{I} \otimes \dots \otimes \hat{I}}_{n-l-1} = \begin{bmatrix} -1 & & & \\ & * & & \\ & & \dots & \\ & & & -1 \end{bmatrix}_{2^{n-1} \times 2^{n-1}}.$$

Suppose the n -th agent chooses arbitrary parameters (θ, ϕ) in his strategy $(\theta, \phi, a_n, t_n, z_n)$, let

$$\hat{\omega}_n(\theta, \phi) = \begin{bmatrix} e^{i\phi} \cos(\theta/2) & i \sin(\theta/2) \\ i \sin(\theta/2) & e^{-i\phi} \cos(\theta/2) \end{bmatrix},$$

then,

$$\begin{aligned}
& \underbrace{\hat{C}_l \otimes \dots \otimes \hat{C}_l}_l \otimes \underbrace{\hat{I} \otimes \dots \otimes \hat{I}}_{n-l-1} \otimes \hat{\omega}_n(\theta, \phi) \\
&= \begin{bmatrix} -e^{i\phi} \cos(\theta/2) * \\ -i \sin(\theta/2) * \\ * * \\ * * \\ \dots \\ * -i \sin(\theta/2) \\ * -e^{-i\phi} \cos(\theta/2) \end{bmatrix}_{2^n \times 2^n} . \\
\vec{\psi}_2 = \underbrace{[\hat{C}_l \otimes \dots \otimes \hat{C}_l]}_l \otimes \underbrace{[\hat{I} \otimes \dots \otimes \hat{I}]}_{n-l-1} \otimes \hat{\omega}_n(\theta, \phi) \vec{\psi}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} -e^{i\phi} \cos(\theta/2) \\ -i \sin(\theta/2) \\ 0 \\ \dots \\ 0 \\ \sin(\theta/2) \\ -ie^{-i\phi} \cos(\theta/2) \end{bmatrix}_{2^n \times 1} ,
\end{aligned}$$

$$\vec{\psi}_3 = \hat{J}^+ \vec{\psi}_2 = \begin{bmatrix} -\cos(\theta/2) \cos \phi \\ -i \sin(\theta/2) \\ 0 \\ \dots \\ 0 \\ \cos(\theta/2) \sin \phi \end{bmatrix}_{2^n \times 1} .$$

The probability distribution Δ is computed from $\vec{\psi}_3$:

$$\begin{aligned}
P_{C\dots CC} &= \cos^2(\theta/2)[1 - \sin^2 \phi], & P_{C\dots CD} &= \sin^2(\theta/2) \\
P_{D\dots DC} &= 0, & P_{D\dots DD} &= \cos^2(\theta/2) \sin^2 \phi
\end{aligned}$$

Obviously,

$$P_{C\dots CC} + P_{C\dots CD} + P_{D\dots DC} + P_{D\dots DD} = 1.$$

Consider the payoff to the n -th agent,

$$\$_n = \$_{C\dots CC} \cos^2(\theta/2)[1 - \sin^2 \phi] + \$_{C\dots CD} \sin^2(\theta/2) + \$_{D\dots DD} \cos^2(\theta/2) \sin^2 \phi$$

Since λ_4 is satisfied, *i.e.*, $\$_{C\dots CC} > \$_{D\dots DD}$, then the n -th agent chooses $\phi = 0$. As a result,

$$\$_n = \$_{C\dots CC} \cos^2(\theta/2) + \$_{C\dots CD} \sin^2(\theta/2)$$

Since the n -th agent belongs to $N \setminus \hat{N}$, by condition λ_3 , (a_n, t_n, z_n) can be chosen as $(\hat{a}, \hat{t}, 0)$. According to Step 5 of *MessageComputing*, $\$_{C\dots CC} = \$_{C\dots CD}$. Thus, $\$_n = \$_{C\dots CC}$. In this case, $\hat{\omega}_n(\theta, \phi)$ can be chosen as $\hat{\omega}(0, 0) = \hat{I}$.

By symmetry, in state \bar{t} , consider the following strategy: each agent $j \in N \setminus \hat{N}$ submits $(\theta_j, \phi_j) = (0, 0)$, $(a_j, t_j, z_j) = (\hat{a}, \hat{t}, 0)$; each agent $j \in \hat{N}$ submits $(\theta_j, \phi_j) = (0, \pi/l)$. Then this strategy profile is a Nash equilibrium of Γ in state \bar{t} , and the final outcome implemented in Nash equilibrium is $\hat{a} \notin F(\bar{t})$.

Note: The proof of Lemma 1 cites the derivation of Eq. (25) [4].