Two-agent Nash implementation: A new result

Wu, Haoyang

5 April 2011

Two-agent Nash implementation:
A new result

Haoyang Wu [*]

Abstract

[Moore and Repullo, *Econometrica* **58** (1990) 1083-1099] and [Dutta and Sen, *Rev. Econom. Stud.* **58** (1991) 121-128] are two fundamental papers on two-agent Nash implementation. Both of them are based on Maskin's classic paper [Maskin, *Rev. Econom. Stud.* **66** (1999) 23-38]. A recent work [Wu, http://arxiv.org/abs/1002.4294, *Inter. J. Quantum Information*, 2010 (accepted)] shows that when an additional condition is satisfied, the Maskin's theorem will no longer hold by using a quantum mechanism. Furthermore, this result holds in the macro world by using an algorithmic mechanism. In this paper, we will investigate two-agent Nash implementation by virtue of the algorithmic mechanism. The main result is: The sufficient and necessary conditions for Nash implementation with two agents shall be amended, not only in the quantum world, but also in the macro world.

Key words: Quantum game theory; Mechanism design; Nash implementation.

1 Introduction

Game theory and mechanism design play important roles in economics. Game theory aims to investigate rational decision making in conflict situations, whereas mechanism design just concerns the *reverse* question: given some desirable outcomes, can we design a game that produces them? Ref. [1] is seminal work in the field of mechanism design. It provides an almost complete characterization of social choice rules that are Nash implementable when the number of agents is at least three. In 1990, Moore and Repullo [2] gave a necessary and sufficient condition for Nash implementation with two agents and many agents. Dutta and Sen [3] independently gave an equivalent result for two-agent Nash implementation. In 2009, Busetto and Codognato [4]

---

[*] Department of Computer Science, Xi'an, 710049, China.
  Email addresses: hywch@mail.xjtu.edu.cn, Tel: 86-18621753457 (Haoyang Wu).

gave an amended necessary and sufficient condition for two-agent Nash implementation. These papers together construct a framework for two-agent Nash implementation.

In 2010, Wu [5] claimed that the Maskin's theorem are amended by virtue of a quantum mechanism, i.e., a social choice rule that is monotonic and satisfies no-veto will not be Nash implementable if it satisfies an additional condition. Although current experimental technologies restrict the quantum mechanism to be commercially available, Wu [6] propose an algorithmic mechanism that amends the sufficient and necessary conditions for Nash implementation with three or more agents in the macro world. Inspired by these results, it is natural to ask what will happen if the algorithmic mechanism can be generalized to two-agent Nash implementation. This paper just concerns this question.

The rest of this paper is organized as follows: Section 2 recalls preliminaries of two-agent Nash implementation given by Moore and Repullo [2]. Section 3 and 4 are the main parts of this paper, in which we will propose two-agent quantum and algorithmic mechanisms respectively. Section 5 draws the conclusions. In Appendix, we explain that the social choice rule given in Section 3 satisfies condition $\mu2$ defined by Moore and Repullo.

2 Preliminaries

Consider an environment with a finite set $I = \{1, 2\}$ of agents, and a (possibly infinite) set A of feasible outcomes. The profile of the agents' preferences over outcomes is indexed by $\theta \in \Theta$, where Θ is the set of preference profiles. Under θ, agent $j \in I$ has preference ordering $R_j(\theta)$ on the set A. Let $P_j(\theta)$ denote the strict preference relation corresponding to $R_j(\theta)$.

For any $j \in I$, $\theta \in \Theta$ and $a \in A$, let $L_j(a, \theta)$ be the lower contour set of agent j at a under θ, i.e., $L_j(a, \theta) = \{\hat{a} \in A : aR_j(\theta)\hat{a}\}$. For any $j \in I$, $\theta \in \Theta$ and $C \subseteq A$, let $M_j(C, \theta)$ be the set of maximal elements in C for agent j under θ, i.e., $M_j(C, \theta) = \{\hat{c} \in C : \hat{c}R_j(\theta)c, \text{ for all } c \in C\}$.

A *social choice rule* (SCR) is a correspondence $f : \Theta \to A$ that specifies a nonempty set $f(\theta) \subseteq A$ for each preference profile $\theta \in \Theta$. A *mechanism* is a function $g : S \to A$ that specifies an outcome $g(s) \in A$ for each vector of strategies $s = (s_1, s_2) \in S = S_1 \times S_2$, where S_j denotes agent j's strategy set.

A mechanism g together with a preference profile $\theta \in \Theta$ defines a game in normal form. Let $NE(g, \theta) \subseteq S$ denote the set of pure strategy Nash equilibria of the game (g, θ). A mechanism g is said to *Nash implement* an SCR f if for all $\theta \in \Theta$, $\{g(s) : s \in NE(g, \theta)\} = f(\theta)$.

Condition μ: There is a set $B \subseteq A$, and for each $j \in I$, $\theta \in \Theta$, and $a \in f(\theta)$, there is a set $C_j(a, \theta) \subseteq B$, with $a \in M_j(C_j(a, \theta), \theta)$ such that for all $\theta^* \in \Theta$, (i), (ii) and (iii) are satisfied:
(i) if $a \in M_1(C_1(a, \theta), \theta^*) \cap M_2(C_2(a, \theta), \theta^*)$, then $a \in f(\theta^*)$;
(ii) if $c \in M_j(C_j(a, \theta), \theta^*) \cap M_k(B, \theta^*)$, for $j, k \in I$, $j \neq k$, then $c \in f(\theta^*)$;
(iii) if $d \in M_1(B, \theta^*) \cap M_2(B, \theta^*)$, then $d \in f(\theta^*)$;

Condition $\mu 2$: Condition μ holds. In addition, for each 4-tuple $(a, \theta, b, \phi) \in A \times \Theta \times A \times \Theta$, with $a \in f(\theta)$ and $b \in f(\phi)$, there exists $e = e(a, \theta, b, \phi)$ contained in $C_1(a, \theta) \cap C_2(b, \phi)$ such that for all $\theta^* \in \Theta$, (iv) is satisfied:
(iv) if $e \in M_1(C_1(a, \theta), \theta^*) \cap M_2(C_2(b, \phi), \theta^*)$, then $e \in f(\theta^*)$.

Theorem 1 (Moore and Repullo, 1990): Suppose that there are two agents. Then a social choice rule f is Nash implementable if and only if it satisfies condition $\mu 2$.

To facilitate the following discussion, here we cite the Moore-Repullo's mechanism as follows: For each agent $j \in I$, Let $S_j = \{(\theta_j, a_j, b_j, n_j) \in \Theta \times A \times B \times N : a_j \in f(\theta_j)\}$, where N denotes the set of non-negative integers, and define the mechanism $g : S \to A$ such that for any $s \in S$:
(1) if $(a_1, \theta_1) = (a_2, \theta_2) = (a, \theta)$, then $g(s) = a$;
(2) if $(a_1, \theta_1) \neq (a_2, \theta_2)$ and $n_1 = n_2 = 0$, then $g(s) = e(a_2, \theta_2, a_1, \theta_1)$;
(3) if $(a_1, \theta_1) \neq (a_2, \theta_2)$ and $n_1 > n_2 = 0$, then $g(s) = b_1$ if $b_1 \in C_1(a_2, \theta_2)$, and $g(s) = e(a_2, \theta_2, a_1, \theta_1)$ otherwise;
(4) if $(a_1, \theta_1) \neq (a_2, \theta_2)$ and $n_2 > n_1 = 0$, then $g(s) = b_2$ if $b_2 \in C_2(a_1, \theta_1)$, and $g(s) = e(a_2, \theta_2, a_1, \theta_1)$ otherwise;
(5) if $(a_1, \theta_1) \neq (a_2, \theta_2)$ and $n_1 \geq n_2 > 0$, then $g(s) = b_1$;
(6) if $(a_1, \theta_1) \neq (a_2, \theta_2)$ and $n_2 > n_1 > 0$, then $g(s) = b_2$.

3   A two-agent quantum mechanism

In this section, first we will show an example of a Pareto-inefficient two-agent SCR f that satisfies condition $\mu 2$, i.e., it is Nash implementable according to Moore-Repullo's mechanism. Then, we will propose a two-agent version of quantum mechanism, which amends the sufficient and necessary conditions for Nash implementation for two agents. Hence, f will not be Nash implementable in the quantum domain.

3.1 A Pareto-inefficient two-agent SCR

Consider an SCR f given in Table 1. $I = \{1, 2\}$, $A = \{a^1, a^2, a^3, a^4\}$, $\Theta = \{\theta^1, \theta^2\}$. In each preference profile, the preference relations over the outcome set A and the corresponding SCR f are given in Table 1. f is Pareto-inefficient from the viewpoint of two agents because in the preference profile $\theta = \theta^2$, both agents prefer a Pareto-efficient outcome $a^1 \in f(\theta^1)$: for each agent $j \in I$, $a^1 P_j(\theta^2) a^2$. However, since f satisfies condition $\mu 2$ (see the Appendix), it is Nash implementable according to Moore-Repullo's theorem.

Table 1. A Pareto-inefficient two-agent SCR f that satisfies condition $\mu 2$.

	θ^1		θ^2	
agent 1	agent 2	agent 1	agent 2
a^3	a^2	a^4	a^3
a^1	a^1	a^1	a^1
a^2	a^4	a^2	a^2
a^4	a^3	a^3	a^4
$f(\theta^1) = \{a^1\}$		$f(\theta^2) = \{a^2\}$	

3.2 A two-agent quantum mechanism

Following Ref. [5], here we will propose a two-agent quantum mechanism to help agents combat "bad" social choice functions. According to Eq (4) in Ref. [8], two-parameter quantum strategies are drawn from the set:

$$\hat{\omega}(\theta, \phi) \equiv \begin{bmatrix} e^{i\phi} \cos(\theta/2) & i \sin(\theta/2) \\ i \sin(\theta/2) & e^{-i\phi} \cos(\theta/2) \end{bmatrix}, \tag{1}$$

$\hat{\Omega} \equiv \{\hat{\omega}(\theta, \phi) : \theta \in [0, \pi], \phi \in [0, \pi/2]\}$, $\hat{J} \equiv \cos(\gamma/2)\hat{I}^{\otimes n} + i \sin(\gamma/2)\hat{\sigma}_x^{\otimes n}$, where γ is an entanglement measure, and $\hat{I} \equiv \hat{\omega}(0, 0)$, $\hat{D} \equiv \hat{\omega}(\pi, \pi/2)$, $\hat{C} \equiv \hat{\omega}(0, \pi/2)$.

Without loss of generality, we assume:
1) Each agent $j \in I$ has a quantum coin j (qubit) and a classical card j. The basis vectors $|C\rangle \equiv (1, 0)^T$, $|D\rangle \equiv (0, 1)^T$ of a quantum coin denote head up and tail up respectively.
2) Each agent $j \in I$ independently performs a local unitary operation on his/her own quantum coin. The set of agent j's operation is $\hat{\Omega}_j = \hat{\Omega}$. A
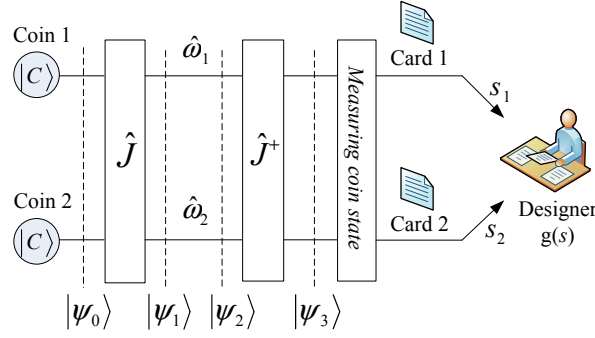
4

Fig. 1. The setup of a two-agent quantum mechanism. Each agent has a quantum coin and a card. Each agent independently performs a local unitary operation on his/her own quantum coin.

strategic operation chosen by agent j is denoted as $\hat{\omega}_j \in \hat{\Omega}_j$. If $\hat{\omega}_j = \hat{I}$, then $\hat{\omega}_j(|C\rangle) = |C\rangle$, $\hat{\omega}_j(|D\rangle) = |D\rangle$; If $\hat{\omega}_j = \hat{D}$, then $\hat{\omega}_j(|C\rangle) = |D\rangle$, $\hat{\omega}_j(|D\rangle) = |C\rangle$. $\hat{I}$ denotes "*Not flip*", $\hat{D}$ denotes "*Flip*".

3) The two sides of a card are denoted as Side 0 and Side 1. The information written on the Side 0 (or Side 1) of card j is denoted as $card(j,0)$ (or $card(j,1)$). A typical card of agent j is described as $c_j = (card(j,0), card(j,1)) \in S_j \times S_j$, where S_j is defined in Moore-Repullo's mechanism. The set of c_j is denoted as $C_j \equiv S_j \times S_j$.

4) There is a device that can measure the state of two quantum coins and send strategies to the designer.

Note that if $\hat{\Omega}_j$ is restricted to be $\{\hat{I}, \hat{D}\}$, then $\hat{\Omega}_j$ is equivalent to $\{$*Not flip*, *Flip*$\}$.

Definition 1: A *two-agent quantum mechanism* is defined as $\hat{G} : \hat{S} \to A$, where $\hat{S} = \hat{S}_1 \times \hat{S}_2$, $\hat{S}_j = \hat{\Omega}_j \times C_j$ $(j \in I)$. $\hat{G}$ can also be written as $\hat{G} : (\hat{\Omega}_1 \otimes \hat{\Omega}_2) \times (C_1 \times C_2) \to A$, where $\otimes$ represents tensor product.

We shall use $\hat{S}_{-j}$ to express $\hat{\Omega}_k \times C_k$ $(k \neq j)$, and thus, a strategy profile is $\hat{s} = (\hat{s}_1, \hat{s}_2)$, where $\hat{s}_1 = (\hat{\omega}_1, c_1) \in \hat{S}_1$, $\hat{s}_2 = (\hat{\omega}_2, c_2) \in \hat{S}_2$. A *Nash equilibrium* of $\hat{G}$ played in a preference profile θ is a strategy profile $\hat{s}^* = (\hat{s}_1^*, \hat{s}_2^*)$ such that for any agent $j \in I$, $\hat{s}_j \in \hat{S}_j$, $\hat{G}(\hat{s}_1^*, \hat{s}_2^*)R_j(\theta)\hat{G}(\hat{s}_j, \hat{s}_{-j}^*)$. For each $\theta \in \Theta$, the pair $(\hat{G}, \theta)$ defines a game in normal form. Let $NE(\hat{G}, \theta) \subseteq \hat{S}$ denote the set of pure strategy Nash equilibria of the game $(\hat{G}, \theta)$. Fig. 1 illustrates the setup of the two-agent quantum mechanism $\hat{G}$. Its working steps are shown as follows:

Step 1: The state of each quantum coin is set as $|C\rangle$. The initial state of the two quantum coins is $|\psi_0\rangle = |CC\rangle$.

Step 2: Given a preference profile θ, if the two following conditions are satisfied, goto Step 4:

1) There exists $\theta' \in \Theta$, $\theta' \neq \theta$ such that $a'R_j(\theta)a$ (where $a' \in f(\theta')$, $a \in f(\theta)$) for each agent $j \in I$, and $a'P_k(\theta)a$ for at least one agent $k \in I$;

2) If there exists $\theta'' \in \Theta$, $\theta'' \neq \theta'$ that satisfies the former condition, then $a'R_j(\theta)a''$ (where $a' \in f(\theta')$, $a'' \in f(\theta'')$) for each agent $j \in I$, and $a'P_k(\theta)a''$ for at least one agent $k \in I$.

Step 3: Each agent j sets $c_j = ((\theta_j, a_j, b_j, n_j), (\theta_j, a_j, b_j, n_j)) \in S_j \times S_j$ and $\hat{\omega}_j = \hat{I}$. Goto Step 7.

Step 4: Each agent j sets $c_j = ((\theta', a', *, 0), (\theta_j, a_j, b_j, n_j))$. Let the two quantum coins be entangled by $\hat{J}$. $|\psi_1\rangle = \hat{J}|CC\rangle$.

Step 5: Each agent j independently performs a local unitary operation $\hat{\omega}_j$ on his/her own quantum coin. $|\psi_2\rangle = [\hat{\omega}_1 \otimes \hat{\omega}_2]\hat{J}|CC\rangle$.

Step 6: Let the two quantum coins be disentangled by $\hat{J}^+$. $|\psi_3\rangle = \hat{J}^+[\hat{\omega}_1 \otimes \hat{\omega}_2]\hat{J}|CC\rangle$.

Step 7: The device measures the state of the two quantum coins and sends $card(j, 0)$ (or $card(j, 1)$) as the strategy s_j to the designer if the state of quantum coin j is $|C\rangle$ (or $|D\rangle$).

Step 8: The designer receives the overall strategy $s = (s_1, s_2)$ and let the final outcome be $g(s)$ using rules (1)-(6) of the Moore-Repullo's mechanism. END.

Given two agents, consider the payoff to the second agent, we denote by $\$_{CC}$ the expected payoff when the two agents both choose $\hat{I}$ (the corresponding collapsed state is $|CC\rangle$), and denote by $\$_{CD}$ the expected payoff when the first agent choose $\hat{I}$ and the second agent chooses $\hat{D}$ (the corresponding collapsed state is $|CD\rangle$). $\$_{DD}$ and $\$_{DC}$ are defined similarly. For the case of two-agent Nash implementation, the condition λ in Ref. [5] is reformulated as the following condition λ':

1) λ'_1: Given an SCR f, a preference profile $\theta \in \Theta$ and $a \in f(\theta)$, there exists $\theta' \in \Theta$, $\theta' \neq \theta$ such that $a'R_j(\theta)a$ (where $a' \in f(\theta')$, $a \in f(\theta)$) for each agent $j \in I$, and $a'P_k(\theta)a$ for at least one agent $k \in I$. In going from θ' to θ both agents encounter a preference change around a'.

2) λ'_2: If there exists $\theta'' \in \Theta$, $\theta'' \neq \theta'$ that satisfies λ'_1, then $a'R_j(\theta)a''$ (where $a' \in f(\theta')$, $a'' \in f(\theta'')$) for each agent $j \in I$, and $a'P_k(\theta)a''$ for at least one agent $k \in I$.

3) λ'_3: For each agent $j \in I$, let him/her be the second agent and consider his/her payoff, $\$_{CC} > \$_{DD}$.

4) λ'_4: For each agent $j \in I$, let him/her be the second agent and consider his/her payoff, $\$_{CC} > \$_{CD}\cos^2\gamma + \$_{DC}\sin^2\gamma$.

Proposition 1: For two agents, given a preference profile $\theta \in \Theta$ and a "bad" SCR f (from the viewpoint of agents) that satisfies condition $\mu2$, agents who satisfies condition λ' can combat the "bad" SCR f by virtue of a two-agent quantum mechanism $\hat{G} : \hat{S} \to A$, i.e., there exists a Nash equilibrium $\hat{s}^* \in NE(\hat{G}, \theta)$ such that $\hat{G}(\hat{s}^*) \notin f(\theta)$.

The proof is straightforward according to Proposition 2 in Ref. [5]. Let us reconsider the SCR f given in Section 3.1. Obviously, when the true preference profile is θ^2, the two conditions in Step 2 of $\hat{G}$ are satisfied. Hence, $\hat{G}$

will enter Step 4. In Step 4, two agents set $c_1 = ((\theta^1, a^1, *, 0), (\theta^2, a^2, *, 0))$, $c_2 = ((\theta^1, a^1, *, 0), (\theta^2, a^2, *, 0))$. For any agent $j \in I$, let him/her be the second agent. Consider the payoff of the second agent, suppose $\$_{CC} = 3$ (the corresponding outcome is a^1), $\$_{CD} = 5$ (the corresponding outcome is $e(a^1, \theta^1, a^2, \theta^2) = a^4$ if $j = 1$, and $e(a^2, \theta^2, a^1, \theta^1) = a^3$ if $j = 2$), $\$_{DC} = 0$ (the corresponding outcome is $e(a^2, \theta^2, a^1, \theta^1) = a^3$ if $j = 1$, and $e(a^1, \theta^1, a^2, \theta^2) = a^4$ if $j = 2$), $\$_{DD} = 1$ (the corresponding outcome is a^2). Hence, condition λ_3' is satisfied, and condition λ_4' becomes: $3 \geq 5 \cos^2 \gamma$. If $\sin^2 \gamma \geq 0.4$, condition λ_4' is satisfied.

Therefore, in the preference profile $\theta = \theta^2$, there exists a novel Nash equilibrium $\hat{s}^* = (\hat{s}_1^*, \hat{s}_2^*)$, where $\hat{s}_1^* = \hat{s}_2^* = (\hat{C}, ((\theta^1, a^1, *, 0), (\theta^2, a^2, *, 0)))$, such that in Step 8 the strategy received by the designer is $s = (s_1, s_2)$, where $s_1 = s_2 = (\theta^1, a^1, *, 0)$. Consequently, $\hat{G}(\hat{s}^*) = g(s) = a^1 \notin f(\theta^2) = \{a^2\}$, i.e., the Moore and Repullo's theorem does not hold for the "bad" social choice rule f by virtue of the two-agent quantum mechanism $\hat{G}$.

4 A two-agent algorithmic mechanism

Following Ref. [6], in this section we will propose a two-agent algorithmic mechanism to help agents benefit from the two-agent quantum mechanism immediately.

4.1 Matrix representations of quantum states

In quantum mechanics, a quantum state can be described as a vector. For a two-level system, there are two basis vectors: $(1, 0)^T$ and $(0, 1)^T$. In the beginning, we define:

$$|C\rangle = [1, 0]^T, \ |D\rangle = [0, 1]^T, \ |CC\rangle = [1, 0, 0, 0]^T,$$

$$\hat{J} = \begin{bmatrix} \cos(\gamma/2) & 0 & 0 & i\sin(\gamma/2) \\ 0 & \cos(\gamma/2) & i\sin(\gamma/2) & 0 \\ 0 & i\sin(\gamma/2) & \cos(\gamma/2) & 0 \\ i\sin(\gamma/2) & 0 & 0 & \cos(\gamma/2) \end{bmatrix}, \ \gamma \in [0, \pi/2].$$
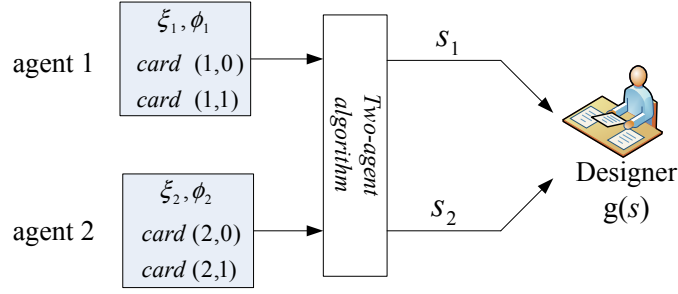
Fig. 2. The inputs and outputs of the two-agent algorithm.

For $\gamma = \pi/2$,

$$\hat{J}_{\pi/2} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 0 & i \\ 0 & 1 & i & 0 \\ 0 & i & 1 & 0 \\ i & 0 & 0 & 1 \end{bmatrix}.$$

$$|\psi_1\rangle = \hat{J}|CC\rangle = \begin{bmatrix} \cos(\gamma/2) \\ 0 \\ 0 \\ i\sin(\gamma/2) \end{bmatrix}$$

4.2   A two-agent algorithm

Following Ref. [6], here we will propose a two-agent version of algorithm that simulates the quantum operations and measurements in Step 4-7 of $\hat{G}$ given in Section 3.2. The entanglement measurement γ can be simply set as its maximum $\pi/2$. The inputs and outputs of the two-agent algorithm are shown in Fig. 2. The *Matlab* program is shown in Fig. 3(a)-(d).

Inputs:
1) (ξ_j, ϕ_j), $j = 1, 2$: the parameters of agent j's local operation $\hat{\omega}_j$, $\xi_j \in [0, \pi], \phi_j \in [0, \pi/2]$.
2) $card(j, 0), card(j, 1) \in S_j$, $j = 1, 2$: the information written on the two sides of agent j's card.

Outputs:
$s_j \in S_j$, $j = 1, 2$: the strategy of agent j that is sent to the designer.

Procedures of the algorithm:

Step 1: Reading two parameters ξ_j and ϕ_j from each agent j (See Fig. 3(a)).

Step 2: Computing the leftmost and rightmost columns of $\hat{\omega}_1 \otimes \hat{\omega}_2$ (See Fig. 3(b)).

Step 3: Computing the vector representation of $|\psi_2\rangle = [\hat{\omega}_1 \otimes \hat{\omega}_2]\hat{J}_{\pi/2}|CC\rangle$.

Step 4: Computing the vector representation of $|\psi_3\rangle = \hat{J}_{\pi/2}^+|\psi_2\rangle$.

Step 5: Computing the probability distribution $\langle\psi_3|\psi_3\rangle$ (See Fig. 3(c)).

Step 6: Randomly choosing a "collapsed" state from the set of all four possible states $\{|CC\rangle, |CD\rangle, |DC\rangle, |DD\rangle\}$ according to the probability distribution $\langle\psi_3|\psi_3\rangle$.

Step 7: For each $j \in I$, the algorithm sends $card(j,0)$ (or $card(j,1)$) as s_j to the designer if the j-th basis vector of the "collapsed" state is $|C\rangle$ (or $|D\rangle$) (See Fig. 3(d)).

4.3 A two-agent version of algorithmic mechanism

Given a two-agent algorithm that simulates the quantum operations and measurements, the two-agent quantum mechanism $\hat{G} : (\hat{\Omega}_1 \otimes \hat{\Omega}_2) \times (C_1 \times C_2) \to A$ can be updated to a two-agent algorithmic mechanism $\widetilde{G} : (\Xi_1 \times \Phi_1) \times (\Xi_2 \times \Phi_2) \times (C_1 \times C_2) \to A$, where $\Xi_1 = \Xi_2 = [0, \pi]$, $\Phi_1 = \Phi_2 = [0, \pi/2]$.

We use $\widetilde{S}_j$ to express $[0,\pi] \times [0,\pi/2] \times C_j$, and $\widetilde{S}_{-j}$ to express $[0,\pi] \times [0,\pi/2] \times C_k$ ($k \neq j$). And thus, a strategy profile is $\widetilde{s} = (\widetilde{s}_j, \widetilde{s}_{-j})$, where $\widetilde{s}_j = (\xi_j, \phi_j, c_j) \in \widetilde{S}_j$ and $\widetilde{s}_{-j} = (\xi_{-j}, \phi_{-j}, c_{-j}) \in \widetilde{S}_{-j}$. A *Nash equilibrium* of a two-agent algorithmic mechanism $\widetilde{G}$ played in a preference profile θ is a strategy profile $\widetilde{s}^* = (\widetilde{s}_1^*, \widetilde{s}_2^*)$ such that for any agent $j \in I$, $\widetilde{s}_j \in \widetilde{S}_j$, $\widetilde{G}(\widetilde{s}_1^*, \widetilde{s}_2^*)R_j(\theta)\widetilde{G}(\widetilde{s}_j, \widetilde{s}_{-j}^*)$.

Working steps of the two-agent algorithmic mechanism $\widetilde{G}$:

Step 1: Given an SCR f and a preference profile θ, if the two following conditions are satisfied, goto Step 3:

1) There exists $\theta' \in \Theta$, $\theta' \neq \theta$ such that $a'R_j(\theta)a$ (where $a' \in f(\theta')$, $a \in f(\theta)$) for each agent $j \in I$, and $a'P_k(\theta)a$ for at least one agent $k \in I$;

2) If there exists $\theta'' \in \Theta$, $\theta'' \neq \theta'$ that satisfies the former condition, then $a'R_j(\theta)a''$ (where $a' \in f(\theta')$, $a'' \in f(\theta'')$) for each agent $j \in I$, and $a'P_k(\theta)a''$ for at least one agent $k \in I$.

Step 2: Each agent j sets $card(j,0) = (\theta_j, a_j, b_j, n_j)$ and sends $card(j,0)$ as the strategy s_j to the designer. Goto Step 5.

Step 3: Each agent j sets $card(j,0) = (\theta', a', *, 0)$ and $card(j,1) = (\theta_j, a_j, b_j, n_j)$, then submits ξ_j, ϕ_j, $card(j,0)$ and $card(j,1)$ to the two-agent algorithm.

Step 4: The two-agent algorithm runs in a computer and outputs strategies s_1 and s_2 to the designer.

Step 5: The designer receives the overall strategy $s = (s_1, s_2)$ and let the final

9

outcome be $g(s)$ using rules (1)-(6) of the Moore-Repullo's mechanism. END.

4.4 New result for two-agent Nash implementation

As we have seen, in the two-agent algorithmic mechanism $\widetilde{G}$, the entanglement measurement γ is reduced to its maximum $\pi/2$. Hence, condition λ' shall be revised as $\lambda'^{\pi/2}$, where $\lambda_1'^{\pi/2}$, $\lambda_2'^{\pi/2}$ and $\lambda_3'^{\pi/2}$ are the same as λ_1', λ_2' and λ_3' respectively. $\lambda_4'^{\pi/2}$ is revised as follows:
$\lambda_4'^{\pi/2}$: For each agent $j \in I$, let him/her be the second agent and consider his/her payoff, $\$_{CC} > \$_{DC}$.

Proposition 2: For two agents, given a preference profile $\theta \in \Theta$ and an SCR f that satisfies condition $\mu2$:
1) If condition $\lambda'^{\pi/2}$ is satisfied, then f is not Nash implementable.
2) If condition $\lambda'^{\pi/2}$ is not satisfied, then f is Nash implementable. Put differently, the sufficient and necessary conditions for Nash implementation with two agents are updated as condition $\mu2$ and no-$\lambda'^{\pi/2}$.

The proof is straightforward according to Proposition 1 in Ref. [6]. Obviously, the two-agent algorithmic mechanism proposed here is a completely "*classical*" one that can be run in a computer.

5 Conclusions

This paper generalizes the quantum and algorithmic mechanisms in Refs. [5,6] to the case of two-agent Nash implementation. Although Moore and Repullo used the phrase "a full characterization" to claim that the problem of two-agent Nash implementation had been completely solved, we argue that there exists a new result as Proposition 2 specifies.

Since the two-agent quantum mechanism only requires two qubits to work, theoretically current experimental technologies of quantum information are adequate [7]. Moreover, the problem of time and space complexity existed in the algorithmic mechanism [6] does not exist here because the number of agents are exactly two. Therefore, the two-agent algorithmic mechanism can be applied to practical cases immediately. In this sense, the new result on two-agent Nash implementation holds not only in the quantum world, but also in the macro world.

Appendix

Consider the SCR f specified by Table 1. $I = \{1, 2\}$, $A = \{a^1, a^2, a^3, a^4\}$, $\Theta = \{\theta^1, \theta^2\}$. Let $B = A$ and $C_j(a, \theta) = L_j(a, \theta)$ for each $j \in I$, $\theta \in \Theta$, $a \in f(\theta)$, i.e.,

$$
\begin{aligned}
C_1(a^1, \theta^1) &= L_1(a^1, \theta^1) = \{a^1, a^2, a^4\}, \\
C_2(a^1, \theta^1) &= L_2(a^1, \theta^1) = \{a^1, a^3, a^4\}, \\
C_1(a^2, \theta^2) &= L_1(a^2, \theta^2) = \{a^2, a^3\}, \\
C_2(a^2, \theta^2) &= L_2(a^2, \theta^2) = \{a^2, a^4\}.
\end{aligned}
$$

Obviously,

$$
\begin{aligned}
a^1 &\in M_1(C_1(a^1, \theta^1), \theta^1) = \{a^1\}, \\
a^1 &\in M_2(C_2(a^1, \theta^1), \theta^1) = \{a^1\}, \\
a^2 &\in M_1(C_1(a^2, \theta^2), \theta^2) = \{a^2\}, \\
a^2 &\in M_2(C_2(a^2, \theta^2), \theta^2) = \{a^2\}.
\end{aligned}
$$

For each 4-tuple $(a, \theta, a', \theta') \in A \times \Theta \times A \times \Theta$, let

$$
\begin{aligned}
e(a^1, \theta^1, a^1, \theta^1) &= a^1 \in C_1(a^1, \theta^1) \cap C_2(a^1, \theta^1) = \{a^1, a^4\}, \\
e(a^1, \theta^1, a^2, \theta^2) &= a^4 \in C_1(a^1, \theta^1) \cap C_2(a^2, \theta^2) = \{a^2, a^4\}, \\
e(a^2, \theta^2, a^1, \theta^1) &= a^3 \in C_1(a^2, \theta^2) \cap C_2(a^1, \theta^1) = \{a^3\}, \\
e(a^2, \theta^2, a^2, \theta^2) &= a^2 \in C_1(a^2, \theta^2) \cap C_2(a^2, \theta^2) = \{a^2\}.
\end{aligned}
$$

Case 1): Consider $\theta^* = \theta^1$, $f(\theta^*) = \{a^1\}$.

For rule (i):

$$
\begin{aligned}
M_1(C_1(a^1, \theta^1), \theta^*) \cap M_2(C_2(a^1, \theta^1), \theta^*) &= \{a^1\} \cap \{a^1\} = \{a^1\}, \\
M_1(C_1(a^2, \theta^2), \theta^*) \cap M_2(C_2(a^2, \theta^2), \theta^*) &= \{a^3\} \cap \{a^2\} = \phi.
\end{aligned}
$$

Hence, rule (i) is satisfied.

For rule (ii):

$$
\begin{aligned}
M_1(C_1(a^1, \theta^1), \theta^*) \cap M_2(B, \theta^*) &= \{a^1\} \cap \{a^2\} = \phi, \\
M_1(C_1(a^2, \theta^2), \theta^*) \cap M_2(B, \theta^*) &= \{a^3\} \cap \{a^2\} = \phi, \\
M_2(C_2(a^1, \theta^1), \theta^*) \cap M_1(B, \theta^*) &= \{a^1\} \cap \{a^3\} = \phi, \\
M_2(C_2(a^2, \theta^2), \theta^*) \cap M_1(B, \theta^*) &= \{a^2\} \cap \{a^3\} = \phi.
\end{aligned}
$$

Hence, rule (ii) is satisfied.

For rule (iii):

$$M_1(B, \theta^*) \cap M_2(B, \theta^*) = \{a^3\} \cap \{a^2\} = \phi.$$

Hence, rule (iii) is satisfied.

For rule (iv):

$$e(a^1, \theta^1, a^1, \theta^1) = a^1, \ M_1(C_1(a^1, \theta^1), \theta^*) \cap M_2(C_2(a^1, \theta^1), \theta^*) = \{a^1\} \cap \{a^1\} = \{a^1\},$$
$$e(a^1, \theta^1, a^2, \theta^2) = a^4, \ M_1(C_1(a^1, \theta^1), \theta^*) \cap M_2(C_2(a^2, \theta^2), \theta^*) = \{a^1\} \cap \{a^2\} = \phi,$$
$$e(a^2, \theta^2, a^1, \theta^1) = a^3, \ M_1(C_1(a^2, \theta^2), \theta^*) \cap M_2(C_2(a^1, \theta^1), \theta^*) = \{a^3\} \cap \{a^1\} = \phi,$$
$$e(a^2, \theta^2, a^2, \theta^2) = a^2, \ M_1(C_1(a^2, \theta^2), \theta^*) \cap M_2(C_2(a^2, \theta^2), \theta^*) = \{a^3\} \cap \{a^2\} = \phi.$$

Hence, rule (iv) is satisfied.

Case 2): Consider $\theta^* = \theta^2$, $f(\theta^*) = \{a^2\}$.

For rule (i):

$$M_1(C_1(a^1, \theta^1), \theta^*) \cap M_2(C_2(a^1, \theta^1), \theta^*) = \{a^4\} \cap \{a^3\} = \phi,$$
$$M_1(C_1(a^2, \theta^2), \theta^*) \cap M_2(C_2(a^2, \theta^2), \theta^*) = \{a^2\} \cap \{a^2\} = \{a^2\}.$$

Hence, rule (i) is satisfied.

For rule (ii):

$$M_1(C_1(a^1, \theta^1), \theta^*) \cap M_2(B, \theta^*) = \{a^4\} \cap \{a^3\} = \phi,$$
$$M_1(C_1(a^2, \theta^2), \theta^*) \cap M_2(B, \theta^*) = \{a^2\} \cap \{a^3\} = \phi,$$
$$M_2(C_2(a^1, \theta^1), \theta^*) \cap M_1(B, \theta^*) = \{a^3\} \cap \{a^4\} = \phi,$$
$$M_2(C_2(a^2, \theta^2), \theta^*) \cap M_1(B, \theta^*) = \{a^2\} \cap \{a^4\} = \phi.$$

Hence, rule (ii) is satisfied.

For rule (iii):

$$M_1(B, \theta^*) \cap M_2(B, \theta^*) = \{a^4\} \cap \{a^3\} = \phi.$$

Hence, rule (iii) is satisfied.

For rule (iv):

$$e(a^1, \theta^1, a^1, \theta^1) = a^1, \ M_1(C_1(a^1, \theta^1), \theta^*) \cap M_2(C_2(a^1, \theta^1), \theta^*) = \{a^4\} \cap \{a^3\} = \phi,$$
$$e(a^1, \theta^1, a^2, \theta^2) = a^4, \ M_1(C_1(a^1, \theta^1), \theta^*) \cap M_2(C_2(a^2, \theta^2), \theta^*) = \{a^4\} \cap \{a^2\} = \phi,$$
$$e(a^2, \theta^2, a^1, \theta^1) = a^3, \ M_1(C_1(a^2, \theta^2), \theta^*) \cap M_2(C_2(a^1, \theta^1), \theta^*) = \{a^2\} \cap \{a^3\} = \phi,$$
$$e(a^2, \theta^2, a^2, \theta^2) = a^2, \ M_1(C_1(a^2, \theta^2), \theta^*) \cap M_2(C_2(a^2, \theta^2), \theta^*) = \{a^2\} \cap \{a^2\} = \{a^2\}.$$

Hence, rule (iv) is satisfied.

To sum up, the SCR f given in Table 1 satisfies condition $\mu 2$. Therefore, according to Moore-Repullo's theorem, it *should be* Nash implementable. However, as shown in Section 3 and 4, when condition λ' is satisfied, neither in the quantum world nor in the macro world will the SCR f be Nash implementable.

Acknowledgments

References

[1] E. Maskin, Nash equilibrium and welfare optimality, *Rev. Econom. Stud.* **66** (1999) 23-38.

[2] J. Moore and R. Repullo, Nash implementation: a full characterization, *Econometrica* **58** (1990) 1083-1099.

[3] B. Dutta and A. Sen, A necessary and sufficient condition for two-person Nash implementation, *Rev. Econom. Stud.* **58** (1991) 121-128.

[4] F. Busutto and G. Codognato, Reconsidering two-agent Nash implementation, *Social Choice and Welfare* **32** (2009) 171-179.

[5] H. Wu, Quantum mechanism helps agents combat "bad" social choice rules. *International Journal of Quantum Information*, 2010 (accepted).

http://arxiv.org/abs/1002.4294

[6] H. Wu, On amending the sufficient conditions for Nash implementation. *Theoretical Computer Science*, 2011 (submitted).

http://arxiv.org/abs/1004.5327

[7] T.D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe and J.L. O'Brien, Quantum computers, *Nature*, **464** (2010) 45-53.

[8] 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.

```
%*********************************************************
%    A two-agent algorithm
%*********************************************************

% Defining the array of (ξ_i, φ_i), i = 1,2
xi=zeros(2,1);
phi=zeros(2,1);

% Reading agent 1's parameters  (ξ_1, φ_1). For example, ŵ_1 = Ĉ = ŵ(0,π/2)
xi(1)=0;
phi(1)=pi/2;

% Reading agent 2's parameters  (ξ_2, φ_2). For example, ŵ_2 = Ĉ = ŵ(0,π/2)
xi(2)=0;
phi(2)=pi/2;
```

Fig. 3 (a). Reading each agent i's parameters ξ_i and ϕ_i, $i = 1,2$.

```
% Defining two 2*2 matrices A and B
A=zeros(2,2);
B=zeros(2,2);

% Let A represents the local operation ŵ_1 of agent 1.
A(1,1)=exp(i*phi(1))*cos(xi(1)/2);
A(1,2)=i*sin(xi(1)/2);
A(2,1)=A(1,2);
A(2,2)=exp(-i*phi(1))*cos(xi(1)/2);

% Let B represents the local operation ŵ_2 of agent 2.
B(1,1)=exp(i*phi(2))*cos(xi(2)/2);
B(1,2)=i*sin(xi(2)/2);
B(2,1)=B(1,2);
B(2,2)=exp(-i*phi(2))*cos(xi(2)/2);

% Computing the leftmost and rightmost columns of ŵ_1 ⊗ ŵ_2
C=zeros(4, 2);
for row=1 : 2
        C((row-1)*2+1, 1) = A(row,1) * B(1,1);
        C((row-1)*2+2, 1) = A(row,1) * B(2,1);
        C((row-1)*2+1, 2) = A(row,2) * B(1,2);
        C((row-1)*2+2, 2) = A(row,2) * B(2,2);
end
A=C;

% Now the matrix A contains the leftmost and rightmost columns of ŵ_1 ⊗ ŵ_2
```

Fig. 3 (b). Computing the leftmost and rightmost columns of $\hat{\omega}_1 \otimes \hat{\omega}_2$

```
% Computing |ψ₂⟩ = [ω̂₁ ⊗ ω̂₂]Ĵ_{π/2}|CC⟩
psi2=zeros(4,1);
for row=1 : 4
        psi2(row)=(A(row,1)+A(row,2)*i)/sqrt(2);
end

% Computing |ψ₃⟩ = Ĵ⁺_{π/2}|ψ₂⟩
psi3=zeros(4,1);
for row=1 : 4
        psi3(row)=(psi2(row) - i*psi2(5-row))/sqrt(2);
end

% Computing the probability distribution ⟨ψ₃|ψ₃⟩
distribution=psi3.*conj(psi3);
distribution=distribution./sum(distribution);
```

Fig. 3 (c). Computing $|\psi_2\rangle, |\psi_3\rangle, \langle\psi_3|\psi_3\rangle$.

```
% Randomly choosing a "collapsed" state according to the probability distribution ⟨ψ₃|ψ₃⟩
random_number=rand;
temp=0;
for index=1: 4
   temp = temp + distribution(index);
   if temp >= random_number
      break;
   end
end

% indexstr: a binary representation of the index of the collapsed state
%   '0' stands for |C⟩, '1' stands for |D⟩
indexstr=dec2bin(index-1);
sizeofindexstr=size(indexstr);

% Defining an array of strategies for two agents
strategy=cell(2,1);

% For each agent j ∈ I, the algorithm generates the strategy s_j
for index=1 : 2 - sizeofindexstr(2)
   strategy{index,1}=strcat('card(',int2str(index),',0)');
end
for index=1 : sizeofindexstr(2)
   if indexstr(index)=='0'      % Note: '0' stands for |C⟩
      strategy{2-sizeofindexstr(2)+index,1}=strcat('card(',int2str(2-sizeofindexstr(2)+index),',0)');
   else
      strategy{2-sizeofindexstr(2)+index,1}=strcat('card(',int2str(2-sizeofindexstr(2)+index),',1)');
   end
end

% The algorithm outputs the strategies s₁, s₂ to the designer
for index=1:2
   disp(strategy(index));
end
```

Fig. 3 (d). Computing the strategies s_1, s_2. This part corresponds to Step 7 of the two-agent quantum mechanism in Section 4.2.

15