Fluctuation and Noise Letters
Vol. 5, No. 4 (2005) L471–L478

World Scientific
www.worldscientific.com

QUANTUM IMPLEMENTATION OF PARRONDO'S PARADOX

PIOTR GAWRON
The Institute of Theoretical and Applied Informatics of the Polish Academy of Sciences, Bałtycka 5, 44-100 Gliwice, Poland
gawron@iitis.gliwice.pl

JAROSŁAW A. MISZCZAK
The Institute of Theoretical and Applied Informatics of the Polish Academy of Sciences, Bałtycka 5, 44-100 Gliwice, Poland
and
Institute of Physics, University of Silesia, Uniwersytecka 4, 40-007 Katowice, Poland
miszczak@iitis.gliwice.pl

Received 28 February 2005
Revised 22 August 2005
Accepted 23 August 2005
Communicated by Andreas Klappenecker

We propose a quantum implementation of a capital-dependent Parrondo's paradox that uses $O(\log_2(n))$ qubits, where n is the number of Parrondo games. We present its implementation in the quantum computer language (QCL) and show simulation results.

Keywords: Quantum games; Parrondo paradox.

1. Introduction

Quantum game theory [1–3] is a new field of science having its roots in both game theory and quantum information theory. For about a decade quantum computer scientists have been searching for new methods of quantum algorithm design. Thorough investigation of different quantum games may bring new insight into the development of quantum algorithms.

It was shown that Grover's algorithm [4] can be treated as an example of a quantum Parrondo's paradox [5, 6]. Operators used in Grover's algorithm can be treated as Parrondo games having separately zero expected values, however, if they are interwired, the expected value fluctuates. This effect is well known in Grover's algorithm.

Implementation of a quantum Parrondo's paradox has been described in papers [7–10]. In this paper we present a new implementation scheme of a capital-dependent Parrondo's paradoxical games on a relatively small number of qubits.

2. Parrondo's Paradox

2.1. *Classical version*

Parrondo's paradox consists of a sequence of games, where each game can be interpreted as a toss of an asymmetrical coin. Every success means that the player gains one dollar, every loss means that the player loses one dollar. There are two games. Game $\mathbb{A}$ has probability of winning $1/2 - \epsilon$. Game $\mathbb{B}$ depends on the amount of capital accumulated by player. If his capital is a multiple of three, the player tosses coin B_1, which has probability of winning $1/10 - \epsilon$, otherwise the player tosses coin B_2 which has probability of winning $3/4 - \epsilon$. Originally $\epsilon = 0.005$, but generally it can be any small real number.

Both games $\mathbb{A}$ and $\mathbb{B}$ are biased and have negative expected gain. But when a player has the option to choose which game he wants to play at each step of the sequence, he can choose such a combination of games which allows him to obtain positive expected gain.

It is known that sequences $(\mathbb{ABBAB})+$ or $(\mathbb{AABB})+$ give relatively high expected gain. This fact is known as *Parrondo's paradox*.

3. Proposed Quantum Implementation

3.1. *Overview*

In [7, 10, 11] the quantum versions of Parrondo games were proposed. The scheme introduced in [7] realizes a history-dependent version of Parrondo's paradox. Its disadvantage is that it needs a large number of qubits to store the history of the games. On the other hand, the scheme by Meyer and Blumer [10] uses Brownian motion of particles in one dimension and does not consume large amounts of quantum resources. The scheme presented by Flitney, Abott and Johnson in [11] is based on multi-coin discrete quantum history-dependent random walk.

The implementation of the capital-dependent quantum Parrondo's paradox introduced in this paper uses only $O(\log_2(n))$ qubits, where n is the number of games played. This allows to perform simulation even when a relatively large number of games are played; for instance, if a strategy consists of three elementary games then 400 iterations require only 15 qubits.

3.2. *Implementation*

3.2.1. *Gates and parameters*

To implement games $\mathbb{A}$ and $\mathbb{B}$, three arbitrary-chosen one-qubit quantum gates A, B_1 and B_2 are used. Each gate is described by four real parameters and our scheme as a whole is described by set of parameters:

- $\{\delta_A, \alpha_A, \beta_A, \theta_A, \delta_{B_1}, \alpha_{B_1}, \beta_{B_1}, \theta_{B_1}, \delta_{B_2}, \alpha_{B_2}, \beta_{B_2}, \theta_{B_2}\}$: real numbers describing gates A, B_1, B_2;
- $\mathbb{S}$: strategy – any sequence of games $\mathbb{A}$, $\mathbb{B}$;
- n: size of $|\$\rangle$ – outcome register (see below);
- *offset*: initial capital offset.

Each gate is composed of elementary gates as presented in Eq. (1):

$$G(\delta_G, \alpha_G, \theta_G, \beta_G) = R_z(\beta_G) R_y(\theta_G) R_z(\alpha_G) Ph(\delta_G), \tag{1}$$

where $G \in \{A, B_1, B_2\}$ and

$$Ph(\xi) = \begin{pmatrix} e^{i\xi} & 0 \\ 0 & e^{i\xi} \end{pmatrix}, R_y(\xi) = \begin{pmatrix} \cos(\frac{\xi}{2}) & -\sin(\frac{\xi}{2}) \\ \sin(\frac{\xi}{2}) & \cos(\frac{\xi}{2}) \end{pmatrix}, R_z(\xi) = \begin{pmatrix} e^{-i(\frac{\xi}{2})} & 0 \\ 0 & e^{i(\frac{\xi}{2})} \end{pmatrix}.$$

3.2.2. *Registers*

The quantum register used to perform this scheme consists of three subregisters:

- $|c\rangle$: one-qubit register representing the coin,
- $|\$\rangle$: n-qubit register storing player's capital,
- $|o\rangle$: three-qubit auxiliary register.

Register $|c\rangle$ holds the state of the *quantum coin*. Gates A, B_1 and B_2 acting on this register represent *quantum coin tosses*. One should note that the register $|c\rangle$ does not store information about history of the games.

After every execution of gates A, B_1 and B_2, the state of the register $|\$\rangle$ is changed according to the result of the *quantum coin toss*. This register is responsible for storing the history of the games, that is, player's capital.

Register $|o\rangle$ is an ancillary register, which one need to check if the state of the $|\$\rangle$ register is a multiple of three. At the beginning of the scheme and after the application of the games' gates this register is always set to $|000\rangle$.

3.2.3. *Games*

Games $\mathbb{A}$ and $\mathbb{B}$ are implemented using the conditional incrementation–decrementation (CID) gate and gates A, B_1 and B_2 described above. In addition, game $\mathbb{B}$ uses gate $mod3$.

Gate $mod3$ sets $|o_1\rangle$ and $|o_2\rangle$ registers to state $|1\rangle$ iff the $|\$\rangle$ register contains a number that is a multiple of three:

$$mod3|a\rangle|0\rangle = |a\rangle|a \text{ (mod 3)}\rangle. \tag{2}$$

The CID gate is responsible for increasing and decreasing the player's capital. The circuit for this gate is presented in Fig. 1(b). This gate increments register $|\$\rangle$ if $|c\rangle$ is in state $|1\rangle$ and decrements if it is in state $|0\rangle$.

Game $\mathbb{A}$ is directly implemented by gate A as presented in Fig. 1(a).

Game $\mathbb{B}$, presented in Fig. 1(c), is more complicated. It uses gate $mod3$ to check if the player's capital is a multiple of three. If it is the case gate B_2 is applied to register $|c\rangle$, otherwise, B_1 is applied.

One can easily check that all gates used in this scheme are unitary because they are composed of elementary unitary operations.

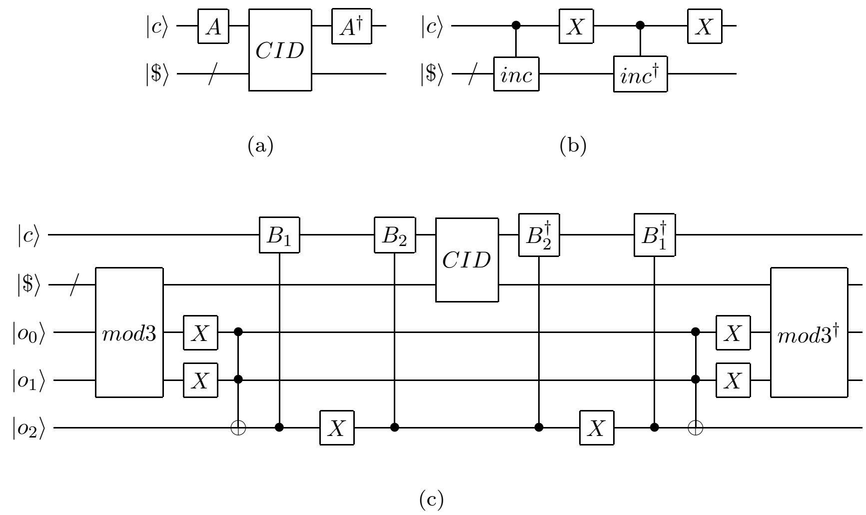


Fig. 1. Gates used to implement Parrondo's paradox. (a) Circuit for the game $\mathbb{A}$. (b) Conditional incrementation–decrementation (CID) circuit. (c) Circuit for the game $\mathbb{B}$.

3.2.4. *Sequence of games*

The game procedure is composed of the following steps:

1. Preparation of $|c\rangle$ in state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.
2. Preparation of $|\$\rangle$ in state $|(2^{(n-1)} + \textit{offset})\rangle$, where *offset* is a small integer number,
3. Preparation of $|o_1 o_2 o_3\rangle$ in $|000\rangle$ state.
4. Application of gates A and B in some chosen order $\mathbb{S}$.

After each application of gate A or B the number stored in register $|\$\rangle$ is either incremented or decremented. The initial state of register $|\$\rangle$ must be chosen in such way that integer overflow is avoided. The maximum number of elementary games cannot exceed the capacity of the register $|\$\rangle$.

3.2.5. *Outcome of games*

If our scheme is implemented on a physical quantum device it should be finalized by measurement. This would give a single outcome representing the final capital. Thus, to obtain expected gain, the experiment should be repeated several times.

Simulation allows to observe the state vector of the quantum system. Using this property the expected gain is calculated as the average value of σ_z in state $|\$\rangle\langle\$| = \mathrm{Tr}_{|c\rangle\otimes|o\rangle}(|c,\$,o\rangle\langle c,\$,o|)$ obtained after tracing out the register with respect to coin and auxiliary subregisters:

$$\langle\$\rangle = \mathrm{Tr}(\sigma_z^{\otimes n}|\$\rangle\langle\$|). \tag{3}$$

4. Simulation

Simulations of a quantum Parrondo's paradox presented in this article were performed using QCL [12]. The source code of the implementation can be found on the webpage listed in Ref. [13].

4.1. *Parameters*

To carry out the simulation, gates A, B_1 and B_2 were prepared with coefficients listed in Table 1. Those coefficients were chosen arbitrarily.

Table 1. Coefficients of the experiment.

δ_A	α_A	β_A	θ_A
0	1	0	$2(\frac{\pi}{2}+0.01)$
δ_{B_1}	α_{B_1}	β_{B_1}	θ_{B_1}
0	1	0	$2(\frac{\pi}{10}+0.01)$
δ_{B_2}	α_{B_2}	β_{B_2}	θ_{B_2}
0	1	0	$2(\frac{3\pi}{4}+0.01)$

4.2. *Results of simulation*

In Fig. 2 the selection of results is presented. As one can see there are strategies that give positive expected values. For *offset* $= 0$, strategy $\mathbb{ABBAB}$ gives a gain of ~ 5.43 after 400 steps. For *offset* $= 3$, strategy $\mathbb{BABBB}$ gives a gain of ~ 13.69 after 400 steps.

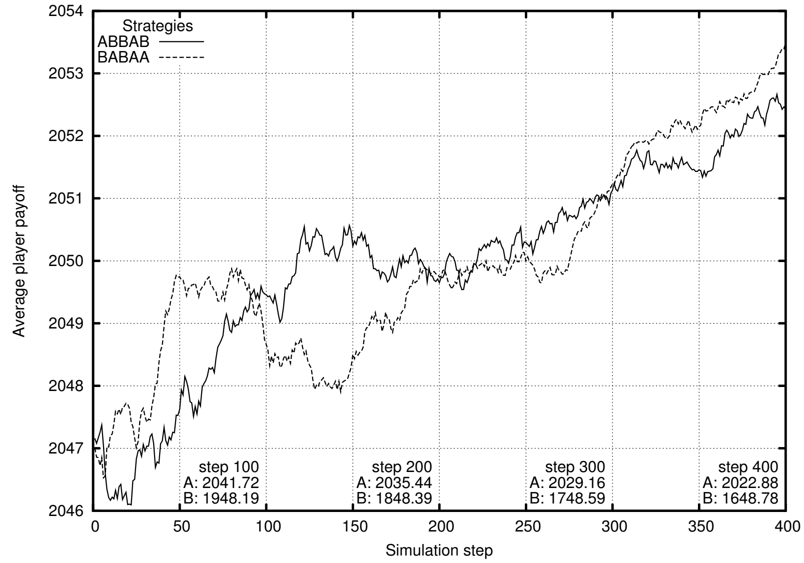
Simulations have shown that finding the winning strategy for a given initial set of parameters is not trivial because they are uncommon. We found that the initial value kept in register $|\$\rangle$ heavily influences the outcome, for example, see Fig. 2. For different offsets different winning strategies can be found. One should note that behavior of the strategy can change if the initial offset is altered, for example, see Fig. 3.

5. Conclusions

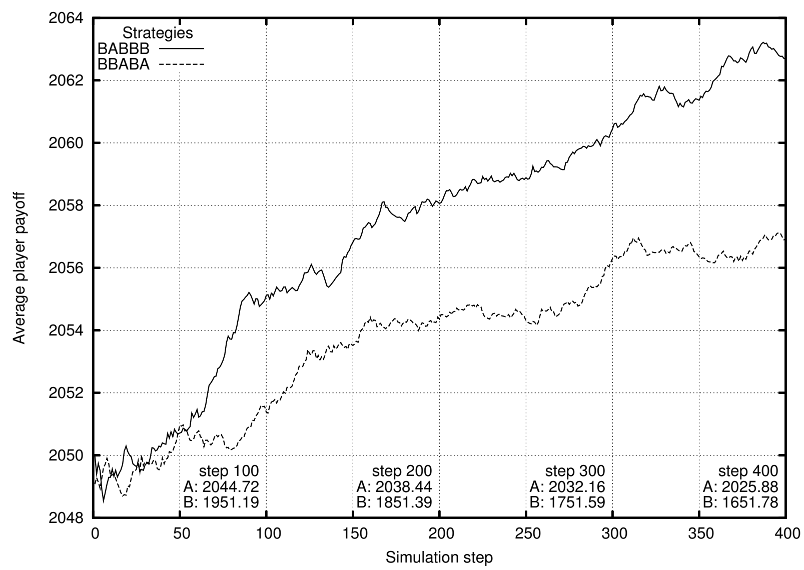
We have shown that it is possible to create a capital-driven scheme for quantum Parrondo games using less than 20 qubits. The main advantage of this scheme is that the size of the register grows as $O(\log_2(n))$, where n is the number of steps. We have found that the initial value of the register $|\$\rangle$ is important for selection of strategy. Simulations have shown that for an analyzed set of strategies composed of five elementary games, winning strategies are uncommon.

Acknowledgments

This paper has been supported by the Polish Ministry of Scientific Research and Information Technology under the (solicited) grant No. PBZ-MIN-008/P03/2003.



(a)



(b)

Fig. 2. Initial offset heavily influences the expected gain. For different offsets we have found different best strategies. The behavior of strategies $\mathbb{A}$ and $\mathbb{B}$ does not depend on initial *offset*. (a) Comparison of two best-found winning strategies for $offset = 0$. Mean values for strategies $\mathbb{A}$ and $\mathbb{B}$ are also noted. (b) Comparison of two best-found winning strategies for $offset = 3$.

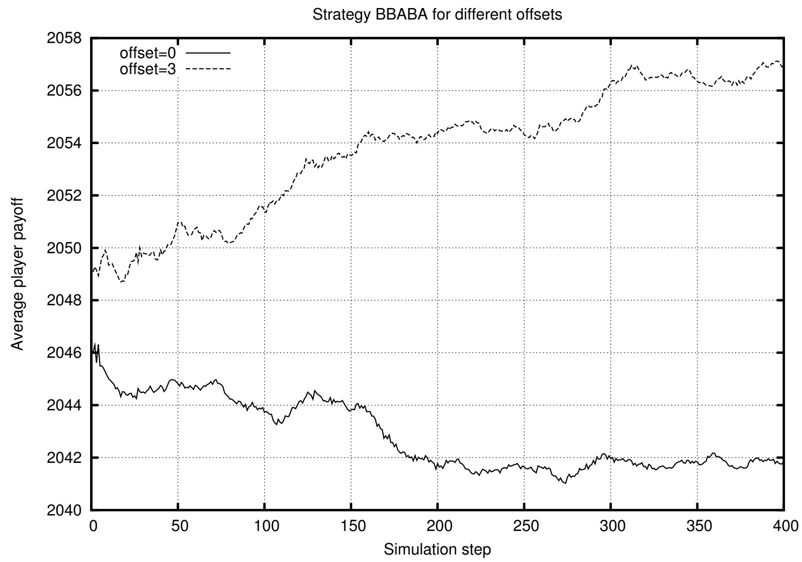


Fig. 3. For different offsets the strategy can behave differently. In this case the strategy $\mathbb{BBABA}$ is winning for *offset* = 3 and losing for *offset* = 0.

References

[1] E. W. Piotrowski and J. Sładkowski, An invitation to quantum game theory, *Int. J. Theor. Phys.* **42** (2003) 1089.

[2] E. W. Piotrowski and J. Sładkowski, The next stage: quantum game theory, in *Mathematical Physics Research at the Cutting Edge* (Nova Science Publishers, Inc., 2004), quant-ph/0308027.

[3] A. P. Flitney and D. Abbott, An introduction to quantum game theory, *Fluct. Noise Lett.* **2** (2002) R175.

[4] L. Grover, A fast quantum mechanical algorithm for database search, in *Proc. 28th Annual ACM Symposium on the Theory of Computation* (ACM Press, New York, 1996), pp. 212–219.

[5] C. F. Lee and N. Johnson, Parrondo games and quantum algorithms, quant-ph/0203043 (2002).

[6] C. F. Lee and N. Johnson, Exploiting randomness in quantum information processing, *Phys. Lett. A* **301** (2002) 343.

[7] A. P. Flitney, J. Ng and D. Abbott, Quantum Parrondo's games, *Physica A* **314** (2002) 35.

[8] G. P. Harmer, D. Abbott and P. G. Taylor, The paradox of Parrondo's games, *Proc. R. Soc. Lond. Ser. A* **456** (2000) 247.

[9] J. M. R. Parrondo, G. P. Harmer and D. Abbott, New paradoxical games based on Brownian ratchets, *Phys. Rev. Lett.* **85** (2000) 5226..

[10] D. Meyer and H. Blumer, Parrondo games as lattice gas automata, *J. Stat. Phys.* **107** (2002) 225.

[11] A. P. Flitney, D. Abbott and N. F. Johnson, Quantum random walks with history dependence, *J. Phys. A* **37** (2004) 7581.

[12] B. Oemer, Quantum programming in QCL, Master's thesis, Technische Universität Wien (2000).

[13] http://www.iitis.gliwice.pl/zksi/.