Bandit Algorithm Driven by a Classical Random Walk and a Quantum Walk

Tomoki Yamagami [1,*] , **Etsuo Segawa** [2] , **Takatomo Mihana** [1] , **André Röhm** [1] , **Ryoichi Horisaki** [1] **and Makoto Naruse** [1]

[1] Department of Information Physics and Computing, Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1 Hongo, Bunkyo, Tokyo 113-8656, Japan; takatomo_mihana@ipc.i.u-tokyo.ac.jp (T.M.); roehm@g.ecc.u-tokyo.ac.jp (A.R.); horisaki@g.ecc.u-tokyo.ac.jp (R.H.); makoto_naruse@ipc.i.u-tokyo.ac.jp (M.N.)

[2] Graduate School of Environment and Information Sciences, Yokohama National University, 79-1 Tokiwadai, Hodogaya, Yokohama 240-8501, Kanagawa, Japan; segawa-etsuo-tb@ynu.ac.jp

* Correspondence: yamagami-tomoki-qwb@g.ecc.u-tokyo.ac.jp

Abstract: Quantum walks (QWs) have a property that classical random walks (RWs) do not possess—the coexistence of linear spreading and localization—and this property is utilized to implement various kinds of applications. This paper proposes RW- and QW-based algorithms for multi-armed-bandit (MAB) problems. We show that, under some settings, the QW-based model realizes higher performance than the corresponding RW-based one by associating the two operations that make MAB problems difficult—exploration and exploitation—with these two behaviors of QWs.

Keywords: random walk; quantum walk; bandit algorithm; exploration–exploitation trade-off; decision-making

1. Introduction

Random walk (RW) is one of the most ubiquitous stochastic processes and is employed for both mathematical analyses and applications, such as describing real-world phenomena and constructing various algorithms. Meanwhile, along with the increasing interest in quantum mechanics from both theoretical and applied perspectives, the quantum counterpart of an RW, known as a quantum walk (QW), is also attracting attention [1–4]. A QW includes the effects of quantum superposition or time evolution. In classical RWs, a random walker (RWer) selects in which direction to go probabilistically at each time step, and thus, one can track where the RWer is at any time step. On the other hand, in QWs, one cannot tell where a quantum walker (QWer) exists during the time evolution, and the location is determined only after conducting the measurement.

QWs have a property that classical RWs do not possess: the coexistence of *linear spreading* and *localization* [5,6]. As a result, QWs show probability distributions that are totally different from those of random walks, which weakly converge to normal distributions. The former behavior, linear spreading, means that the standard deviation of the probability distribution of measurement of quantum walkers (QWers) grows in proportion to the run time t. In the case of discrete-time RWs on a one-dimensional lattice $\mathbb{Z}$, denoting the random variable of the position where a walker is measured at time $t \in \mathbb{N}_0 = \mathbb{N} \cup \{0\}$ by $X_t^{(\mathrm{RW})}$, then the standard deviation is $\mathbb{D}[X_t^{(\mathrm{RW})}] = \mathrm{O}(\sqrt{t})$. On the other hand, in discrete-time QWs on $\mathbb{Z}$, the standard deviation of a walker's position at time t is $\mathbb{D}[X_t^{(\mathrm{QW})}] = \mathrm{O}(t)$, and thus, discrete-time QWs outperform RWs in terms of the propagation velocity [7]. The latter behavior, localization, implies that the probability is distributed at a particular position no matter how long the walk runs. In the classical RWs, the probability distribution becomes flat despite keeping a bell-shaped curve; that is, localization is not observed.

QWs were first introduced in the field of quantum information theory [7–9]. The idea of weak convergence, which is frequently used in probability theory, was introduced to show the properties of QWs [10,11], and since then, quantum walks have been actively studied from both fundamental and applied perspectives. In fundamental fields, there have been many attempts to analyze these evolution models mathematically [6,12–22] due to varying behavior of QWs depending on the conditions or settings of time and space. In applied fields, their unique behavior is useful for implementing quantum structures or quantum analogs of existing models; therefore, various QW-based models have been considered for subjects such as time-series analysis [23], topological insulators [24,25], radioactive waste reduction [26,27], and optics [28,29]. In addition, the contribution to quantum information technology is becoming more prominent these days. QWs have been applied not only to the principle of technologies such as quantum search, quantum teleportation [30,31], and quantum key distribution [32] but also to the implementation of quantum gates themselves [33,34].

Throughout the extensive research conducted, considerable attention has also been devoted to the models of QWs themselves. Initially, QW models were introduced in the form of *coined* QWs, wherein the time evolution of walkers is considered through the utilization of unitary matrices called coin matrices, analogous to transitions in RWs [9]. According to the No-Go Lemma [35,36], we need the additional subspace, namely the coin space to each vertex, to construct a non-trivial unitary time evolution on the cycle, introduced later. This model remains one of the most intuitive models and is widely studied or applied even in contemporary times. On the other hand, depending on the research interests, some literature also explores QW models that do not incorporate coins [37–39], which represent a generalization of quantum cellular automata [35]. The coinless QW models seem to achieve reduced computational costs, but it is shown that such QW models are unitarily equivalent to coined QW models [40]. In this paper, we focus on using the coined QW models because adopting the coined QWs is reasonable when conducting a comparison with RWs. In the following, when referring to QWs, it indicates the *coined* QWs.

This paper proposes new solution schemes for multi-armed bandit (MAB) problems [41] using RWs and QWs. In the MAB problems, we consider a situation where there are multiple slot machines in an environment, each gives a reward with a probability allocated to it, and an agent iterates the selection of slot machines and probabilistic gain of the rewards and tries to maximize the total reward. Initially, the agent has no information about the probability of giving rewards, especially which slot machine has the maximum probability, which we call the best slot machine. Thus, it is required to accumulate such information through a certain number of selections, an action which we call *exploration*. On the other hand, too much exploration will use up the opportunities for selecting the better slot machines that have already been found; that is, it is also necessary to spend some rounds to bet on slot machines that are reliable based on the information obtained, which we call *exploitation*. The difficulty of MAB problems occurs under the balance between these two operations, known as the *exploration–exploitation trade-off* [42].

One of the purposes of this study is to show that, by utilizing QWs, we can construct an algorithm to solve the MAB problem. This algorithm can outperform models implemented by RWs in terms of the total rewards under some settings. Our idea to realize this is to address this dilemma derived from the exploration–exploitation trade-off by utilizing a unique property of QWs, i.e., the coexistence of linear spreading and localization. More precisely, we combine exploration with linear spreading, and exploitation with localization, as shown in Figure 1. By utilizing linear spreading, we intend to cover the whole environment and prevent us from missing some slot machines. In addition, by applying localization, we intend to mark slot machines that should be recommended with a high probability distribution. This paper introduces a QW-based algorithm for MAB problems, which realizes these combinations. Our study focuses on three-state site-dependent QWs on a cycle. The behavior of these walks corresponds to that of lazy random walks, with walkers moving clockwise, anti-clockwise, or staying in place in superposition. Giving

three states to QWs enables us to obtain a high existence probability at the initial position of a QWer. In addition, site-dependent coin matrices make it possible to trap or dam the QWer on certain vertices. By taking advantage of these, we attempt to bring about a high probability on a vertex whose slot machine should be recommended. To facilitate a clear comparison, we also construct an RW-based model wherein lazy RWs occur on cycles, and the transition probabilities depend on the position from which the walkers depart. While the QW-based model possesses the coexistence of linear spreading and localization, the RW-one does not; while the results depend on the specifics of the MAB problem, our study reveals that, for certain settings, the different properties of QWs and RWs also lead to a significant difference in total rewards between both algorithms.
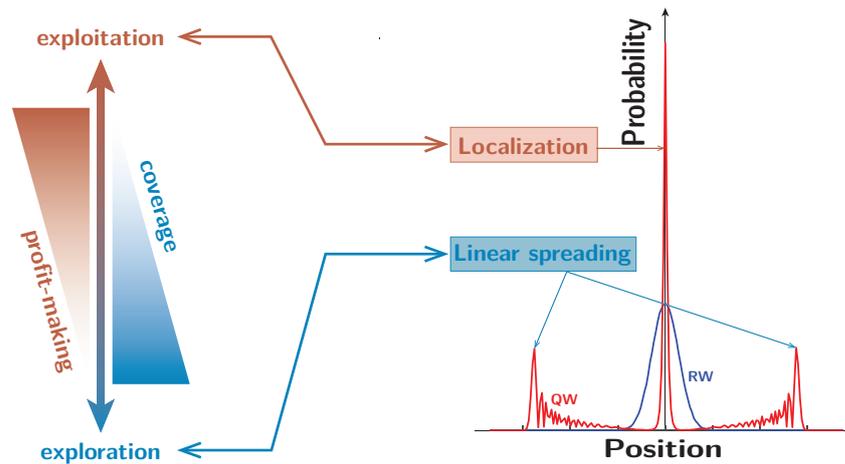
Figure 1. Association between the behaviors of quantum walks (linear spreading and localization) and the operations in MAB problems (exploration and exploitation).

The rest of this paper is organized as follows. First, in Section 2, we present an algorithm for MAB problems based on the RW, which is more intuitive than the QW-based one. Then, in Section 3, we introduce a system of discrete-time quantum walks on a cycle and the QW-based algorithm for MAB problems. In Section 4, we show some results for numerical simulations of the RW- and QW-based models and compare the performance between the two models. Section 5 concludes this paper and discusses the future possibilities of our work.

2. Random-Walk-Based Model for MAB Problem

This section presents an MAB algorithm implemented using a discrete-time random walk (RW) on cycles. The RW model presented in this paper describes the walkers that can stay at the same position, which we often call a *lazy random walk*. First, we present the mathematical system of the lazy RW, and then we construct the MAB algorithm based on it.

2.1. Random Walk on Cycles

Assume that cycle $\mathcal{C}_N$ is composed of N vertices and edges. Here vertices are labeled by set $V_N := \{0, 1, \cdots, N-1\}$, and the label is ordered clockwise. Thus, the set of edges is given by $E_N := \{\{x, x+1\} \mid x \in V_N\}$, where one applies addition and subtraction modulo N to V_N; i.e., $(N-1) + 1 \equiv 0$ and $0 - 1 \equiv N - 1$. In other words, V_N is isometric to $\mathbb{Z}/N\mathbb{Z}$.

We assume that the position of a walker is determined as follows:

- A walker initially exists at position $s \in V_N$.
- At each time step, a walker at position x moves one unit clockwise with probability $q(x)$, moves one unit anti-clockwise with probability $q(x)$ or stays at the current position with probability $1 - 2q(x)$.

Here the probabilities of moving clockwise and anti-clockwise are equal to each other in this paper. This is due to correspondence with the setting of the QW presented later;

therein, the setting of the initial state of the QW gives symmetric probability distributions when the coin matrix is homogeneous for the space. Note that $q(x)$ should satisfy $0 \leq q(x) \leq 1/2$ under this condition, and this setting is equivalent to the simple RW on cycles in the case of $q(x) = 1/2$ for all $x \in V_N$.

Such an RW is mathematically constructed as follows. For $\mathbb{N}_0 = \{0, 1, \cdots\}$, let $\{X_t\}_{t \in \mathbb{N}_0}$ be the sequence of random variables that represent the position of a walker at time step t. The next position X_{t+1} depends on the current one X_t, and the conditional probability is determined as follows:

$$\mathbb{P}(X_{t+1} = x + 1 \mid X_t = x) = \mathbb{P}(X_{t+1} = x - 1 \mid X_t = x) = q(x), \tag{1}$$

$$\mathbb{P}(X_{t+1} = x \mid X_t = x) = 1 - 2q(x) \tag{2}$$

with $x \in V_N$. We recall that $V_N \simeq \mathbb{Z}/N\mathbb{Z}$; the equation above includes

$$\mathbb{P}(X_{t+1} = N - 1 \mid X_t = 0) = q(0), \tag{3}$$

$$\mathbb{P}(X_{t+1} = 0 \mid X_t = N - 1) = q(N - 1). \tag{4}$$

Here we denote the probability that a walker is at position x at time step t by $\nu^{(t)}(x)$:

$$\nu^{(t)}(x) = \mathbb{P}(X_t = x). \tag{5}$$

Then, the relation

$$\nu^{(0)}(x) = \delta_s(x) \tag{6}$$

holds, where $\delta_{x'}(x)$ is the delta function: for $x' \in V_N$,

$$\delta_{x'}(x) = \begin{cases} 1 & (x = x') \\ 0 & (\text{otherwise}) \end{cases}. \tag{7}$$

Moreover, by Equations (1) and (2), $\nu^{(t)}(x)$ varies as follows:

$$\nu^{(t+1)}(x) = q(x+1)\nu^{(t)}(x+1) + (1 - 2q(x))\nu^{(t)}(x) + q(x-1)\nu^{(t)}(x-1). \tag{8}$$

2.2. Random-Walk-Based Algorithm

We consider an N-armed bandit problem with cycle $\mathcal{C}_N$; each vertex $x \in V_N$ is given a slot machine that gives a reward with the probability $p(x)$. In the following, each slot machine is identified by the same label as the corresponding slot machine; for example, we call the slot machine on the vertex x slot machine x. In addition, we call probability $p(x)$ the *success probability of slot machine x*. Moreover, we denote the slot machine with the best success probability in V_N by x^*; that is,

$$x^* = \arg\max_{x \in V_N} p(x), \tag{9}$$

and we call it the *best slot machine*.

The principle consists of the following four steps: **[STEP 0]** initializing the quantum walk settings, **[STEP 1]** running random walks, **[STEP 2]** playing the selected slot machine, and **[STEP 3]** updating the quantum walk settings. After finishing **[STEP 3]**, the process returns to **[STEP 1]**. We call the series of the last three steps (**[STEP 1–3]**, shown in Figure 2) a *decision*, and decisions are iterated J times over a run. Here, we use the following notations:
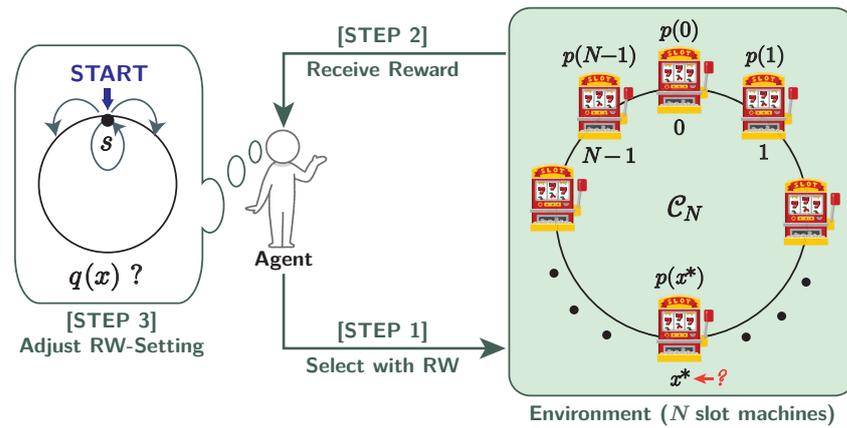
Figure 2. Single decision on the random-walk-based model for MAB problems.

- $s_j \in V_N$: Initial position of random walk in the j-th decision.
- $q_j(x) \in [0, 1/2]$: Clockwise-transition probability and anti-clockwise-transition probability in the j-th decision.
- $\hat{x}_j \in V_N$: Vertex (slot machine) measured in the j-th decision.
- $\hat{r}_j \in \{0, 1\}$: Reward on the j-th decision. This value is probabilistically determined by the Bernoulli distribution $\text{Ber}(p(\hat{x}_j))$; that is,

$$\hat{r}_j := \begin{cases} 1 & (\text{with prob. } p(\hat{x}_j)) \\ 0 & (\text{with prob. } 1 - p(\hat{x}_j)) \end{cases}. \tag{10}$$

- $H_j(x)$: Number of decisions where the slot machine x is selected until the j-th decision.
- $L_j(x)$: Number of decisions where the slot machine x gives the reward until the j-th decision.
- $\hat{p}_j(x)$: Empirical probability that the slot machine x gives the reward on the j-th decision:

$$\hat{p}_j(x) = \begin{cases} \dfrac{L_j(x)}{H_j(x)} & (H_j(x) \neq 0) \\ 0 & (H_j(x) = 0) \end{cases}. \tag{11}$$

[STEP 0] RW-setting initialization
For the first decision, the settings of the random walk are determined as follows:

- Initial position s_1: Probabilistically determined by the uniform distribution on V_N.
- Transition probability: $q_1(x) = q^\circ \in [0, 1/2]$ for all $x \in V_N$.

After finishing this step, the process iterates the following three steps.

[STEP 1] Random walk
Random walks are run over T time steps with the initial position s_j and transition probability $q_j(x)$, and the value $\hat{x}_j \in V_N$ is obtained following probability distribution $v_j^{(T)}(x)$.

[STEP 2] Slot machine play
The slot machine $\hat{x}_j \in V_N$ obtained at **[STEP 1]** is played. Then, the reward ($\hat{r}_j = 1$) is obtained with probability $p(\hat{x}_j)$.

Here H- and L-values are updated. First, the H-value on $\hat{x}_j$ is incremented:

$$H_j(\hat{x}_j) = H_{j-1}(\hat{x}_j) + 1. \tag{12}$$

If $\hat{r}_j = 1$, the L-value on $\hat{x}_j$ is also incremented (otherwise, the value is maintained):

$$L_j(\hat{x}_j) = \begin{cases} L_{j-1}(\hat{x}_j) + 1 & \text{(with prob. } p(\hat{x}_j)) \\ L_{j-1}(\hat{x}_j) & \text{(with prob. } 1 - p(\hat{x}_j)) \end{cases} . \tag{13}$$

For $x \neq \hat{x}_j$, the H- and L-values are maintained:

$$H_j(x) = H_{j-1}(x), \tag{14}$$

$$L_j(x) = L_{j-1}(x). \tag{15}$$

Based on that, $\hat{p}_j(x)$s are updated.

[STEP 3] RW-setting adjustment
Using the new $\hat{p}_j(x)$s, the settings of quantum walks are updated for the next decision. The new initial position is defined as

$$s_{j+1} = \underset{x \in V_N}{\arg\max} \, \hat{p}_j(x). \tag{16}$$

Moreover, the new transition probabilities are determined as

$$q_{j+1}(x) = q^\circ \exp(-a \cdot \hat{p}_j(x)^b) \tag{17}$$

where a, $b \geq 1$, and q° are defined in **[STEP 0]**. Note that the q-value monotonically decreases about the empirical success probability; that is, if $\hat{p}_j(x)$ is larger, then $q_{j+1}(x)$ is smaller. By setting the new initial state and q-value in these manners, we aim at confining walkers to the desired position while concurrently affording them opportunities to depart when the current decision is uncertain. The parameters a and b control the strength of the effect by $\hat{p}_j(x)$; the details are given in Appendix A.

After this step, the process returns to **[STEP 1]**.

3. Quantum-Walk-Based Model for MAB Problem

This section presents the MAB algorithm implemented by the discrete-time quantum walk (QW) on cycles. The difference between QW and RW lies in whether one handles the quantum superposition of states pertaining to the walker's positions. Herein, the transition of walkers at each time step is also superposed; that is, it is uncertain even after the time step which transition occurs: moving clockwise, anti-clockwise, or staying in place. We introduce probability amplitude vectors and coin matrices in the QW to describe the quantum superposition and its time evolution.

The QW model employed in our study is a three-state QW on a cycle, which can be naturally reduced to a finite space from the one-dimensional lattice model [5]; see, for example, [18,43,44]. First, we explain the definition of our QW in detail, and then we present the MAB algorithm based on it.

3.1. Quantum Walk on Cycles

Assume that cycle $\mathcal{C}_N$ is constructed in the same manner as in Section 2; that is, it is the graph established by the set of vertices $V_N := \{0, 1, \cdots, N - 1\}$ and edges $E_N := \{\{x, x + 1\} \mid x \in V_N\}$, where one applies addition and subtraction modulo N to V_N.

The space of the probability amplitude vectors driving our QW is defined in a compound Hilbert space consisting of the position Hilbert space $\mathcal{H}_P$ and the coin Hilbert space $\mathcal{H}_C$. The position Hilbert space $\mathcal{H}_P$ is spanned by the unit vectors corresponding to the vertices on $\mathcal{C}_N$; i.e., $\mathcal{H}_P = \text{span}\{|x\rangle \mid x \in V_N\}$. Here we require them to be mutually orthogonal, which is equivalent to satisfying the relation $\langle y|x \rangle = \delta_y(x)$ for any x, $y \in V_N$, where δ_y is the delta function defined by Equation (7). Then, $\mathcal{H}_P \simeq \mathbb{C}^N$ holds.

The coin Hilbert space $\mathcal{H}_C$ pertains to the internal state of walkers. In this model, we assume that there exist three internal states: clockwise $(+)$, anti-clockwise $(-)$, and staying (O). We define the three-dimensional unit vectors corresponding to them as $|-\rangle = [1\ 0\ 0]^T$, $|O\rangle = [0\ 1\ 0]^T$, and $|+\rangle = [0\ 0\ 1]^T$, where a superscript T on a matrix represents its transpose, and construct the coin Hilbert space as $\mathcal{H}_C = \text{span}\{|-\rangle, |O\rangle, |+\rangle\}$. Here you see that $\mathcal{H}_C = \mathbb{C}^3$. Based on $\mathcal{H}_P$ and $\mathcal{H}_C$, the whole system is described by

$$\mathcal{H}_{PC} = \mathcal{H}_P \otimes \mathcal{H}_C = \text{span}\{|x\rangle \otimes |\varepsilon\rangle \mid x \in V_N, \ \varepsilon \in \{\pm, O\}\}. \tag{18}$$

Then the total state of our QW at time $t \in \mathbb{N}_0$ is represented as follows: there exists $|\psi^{(t)}(x)\rangle \in \mathbb{C}^3$ for each $x \in V_N$ such that

$$|\Psi^{(t)}\rangle = \sum_{x \in V_N} |x\rangle \otimes |\psi^{(t)}(x)\rangle \in \mathcal{H}_{PC}. \tag{19}$$

Here, $t \in \mathbb{N}_0$ represents time step of QWs, and $|\psi^{(t)}(x)\rangle \in \mathbb{C}^3$ is called the probability amplitude vector at position $x \in V_N$ at run time t. We set the initial state as

$$|\Psi^{(0)}\rangle = |\Phi\rangle := |s\rangle \otimes |\varphi\rangle, \tag{20}$$

where $s \in V_N$, and $|\varphi\rangle \in \mathbb{C}^3$ is a constant vector with $\|\varphi\| = 1$. In this paper, we fix $|\varphi\rangle$ to $|O\rangle$, realizing a symmetric probability distribution about the initial position when the coin matrix defined later is homogeneous for positions.

Now, we introduce the time evolution of $|\Psi^{(t)}\rangle$ by

$$|\Psi^{(t+1)}\rangle = U|\Psi^{(t)}\rangle. \tag{21}$$

Here U is the unitary operator, referred to as the time evolution operator, and is composed of shift operator S and coin operator C:

$$U = SC, \tag{22}$$

and S and C are given by

$$S = S^\dagger \otimes |-\rangle\langle-| + I_N \otimes |O\rangle\langle O| + S \otimes |+\rangle\langle+| \tag{23}$$

and

$$C = \sum_{x \in V_N} \left(|x\rangle\langle x| \otimes C(x)\right). \tag{24}$$

Here S is defined as

$$S = \sum_{x \in V_N} |x+1\rangle\langle x| \tag{25}$$

and represents the clockwise transition, and then

$$S^\dagger = \sum_{x \in V_N} |x-1\rangle\langle x| \tag{26}$$

indicates the anti-clockwise transition. The identity matrix I_N corresponds to staying in place. Here, note that $N \equiv 0$ on $V_N (\simeq \mathbb{Z}/N\mathbb{Z})$; that is, for example, in the case of $N = 4$,

$$S = |1\rangle\langle 0| + |2\rangle\langle 1| + |3\rangle\langle 2| + |0\rangle\langle 3| \tag{27}$$

and

$$S^\dagger = |3\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 2| + |2\rangle\langle 3|. \tag{28}$$

$C(x)$ is a unitary matrix called a *coin matrix*, which is defined as follows:

$$C(x) = \begin{bmatrix} -\dfrac{1+\cos\theta(x)}{2} & \dfrac{\sin\theta(x)}{\sqrt{2}} & \dfrac{1-\cos\theta(x)}{2} \\ \dfrac{\sin\theta(x)}{\sqrt{2}} & \cos\theta(x) & \dfrac{\sin\theta(x)}{\sqrt{2}} \\ \dfrac{1-\cos\theta(x)}{2} & \dfrac{\sin\theta(x)}{\sqrt{2}} & -\dfrac{1+\cos\theta(x)}{2} \end{bmatrix} \tag{29}$$

with $\theta(x) \in [0, 2\pi)$ for all $x \in V_N$. Note that, in the case of $\cos\theta(x) = -1/3$, $C(x)$ is reduced to the Grover matrix, which is important in quantum searching [45].

Let us explain the equivalent expression for the time evolution operator U, which is useful to understand the dynamics of our QW: by applying the property of the Kronecker product, we have

$$U = \sum_{x \in V_N} \Big(|x-1\rangle\langle x| \otimes P(x) + |x\rangle\langle x| \otimes R(x) + |x+1\rangle\langle x| \otimes Q(x) \Big), \tag{30}$$

where

$$P(x) = |-\rangle\langle-| C(x) = \begin{bmatrix} -\dfrac{1+\cos\theta(x)}{2} & \dfrac{\sin\theta(x)}{\sqrt{2}} & \dfrac{1-\cos\theta(x)}{2} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{31}$$

$$Q(x) = |+\rangle\langle+| C(x) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \dfrac{1-\cos\theta(x)}{2} & \dfrac{\sin\theta(x)}{\sqrt{2}} & -\dfrac{1+\cos\theta(x)}{2} \end{bmatrix}, \tag{32}$$

$$R(x) = |O\rangle\langle O| C(x) = \begin{bmatrix} 0 & 0 & 0 \\ \dfrac{\sin\theta(x)}{\sqrt{2}} & \cos\theta(x) & \dfrac{\sin\theta(x)}{\sqrt{2}} \\ 0 & 0 & 0 \end{bmatrix}. \tag{33}$$

The matrices $P(x)$, $Q(x)$, and $R(x)$ are considered to be the decomposition elements of $C(x)$; that is, the relation $P(x) + Q(x) + R(x) = C(x)$ holds. They describe the matrix-valued weight of a clockwise transition, an anti-clockwise transition, and staying in place, respectively, corresponding to the transition probabilities of the RW, as shown in Figure 3.

By Equations (21) and (30), we have

$$|\psi^{(t+1)}(x)\rangle = P(x+1)\,|\psi^{(t)}(x+1)\rangle + R(x)\,|\psi^{(t)}(x)\rangle + Q(x-1)\,|\psi^{(t)}(x-1)\rangle. \tag{34}$$

Moreover, from the initial state (20), there exists a 2-dimensional matrix $\Xi^{(t)}(x)$ such that

$$|\psi^{(t)}(x)\rangle = \Xi^{(t)}(x)\,|\varphi\rangle. \tag{35}$$

Here $\Xi^{(t)}(x)$ describes the weight of all the possible paths from the origin to the position x at run time t. From Equation (34), the following relation holds:

$$\Xi^{(t+1)}(x) = P(x+1)\Xi^{(t)}(x+1) + R(x)\Xi^{(t)}(x) + Q(x-1)\Xi^{(t)}(x-1). \tag{36}$$
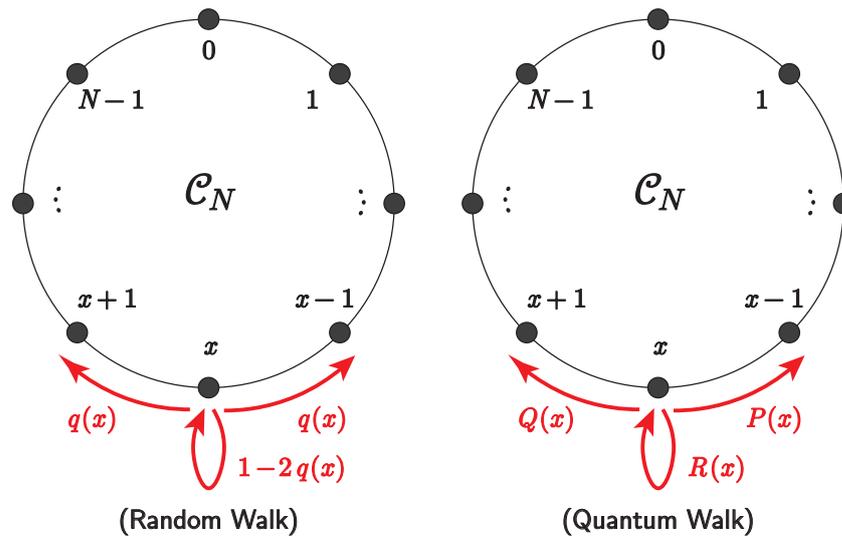
Figure 3. Transition probabilities of RW (**left panel**) and matrix-valued weights of QW (**right panel**).

Finally, the measurement probability of the particle at position x at run time t, denoted by $\mu^{(t)}(x)$, is given by

$$\mu^{(t)}(x) := \|\psi^{(t)}(x)\|^2. \tag{37}$$

Setting random variable X_t following the distribution $\mu^{(t)}$, we call X_t the position of a QWer at time t. This definition is based on the Born probability interpretation in quantum mechanics. Note that for any $t \in \mathbb{N}_0$, the following is satisfied:

$$\sum_{x \in V_N} \mu^{(t)}(x) = \sum_{x \in V_N} \|\psi^{(t)}(x)\|^2 = 1. \tag{38}$$

3.2. Quantum-Walk-Based Algorithm

We consider an N-armed bandit problem with cycle $\mathcal{C}_N$; each vertex $x \in V_N$ is given a slot machine that gives a reward with the probability $p(x)$, identically to the RW-based model in Section 3.

The principle is also similar to the RW-based model: First, the QW settings are initialized (**[STEP 0]**), and then decisions are iterated (**[STEP 1–3]**) J times over a run. As shown in Figure 4, the QW-based model controls the coin matrix $C(x)$ by adjusting the value of the parameter $\theta(x)$ instead of the transition probability $q(x)$ in the RW-based model. Here we use the following notations:

- $|\Phi_j\rangle \in \mathcal{H}_{PC}$: Initial state of quantum walk on the j-th decision.
- $s_j \in V_N$: Initial position of quantum walk on the j-th decision.
- $\theta_j(x) \in [0, 2\pi)$: Parameter of Equation (24) on vertex x on the j-th decision; then the coin matrix there is $C(x)$.
- $\hat{x}_j \in V_N$: Vertex (slot machine) measured on the j-th decision.
- $\hat{r}_j \in \{0, 1\}$: Reward on the j-th decision, which follows the Bernoulli distribution $\text{Ber}(p(\hat{x}_j))$:

$$\hat{r}_j := \begin{cases} 1 & (\text{with prob. } p(\hat{x}_j)) \\ 0 & (\text{with prob. } 1 - p(\hat{x}_j)) \end{cases}. \tag{39}$$

- $H_j(x)$: Number of decisions in which slot machine x is selected until the j-th decision.
- $L_j(x)$: Number of decisions in which slot machine x gives the reward until the j-th decision.

- $\hat{p}_j(x)$: Empirical probability that slot machine x gives the reward on the j-th decision:

$$\hat{p}_j(x) = \begin{cases} \dfrac{L_j(x)}{H_j(x)} & (H_j(x) \neq 0) \\ 0 & (H_j(x) = 0) \end{cases}. \tag{40}$$
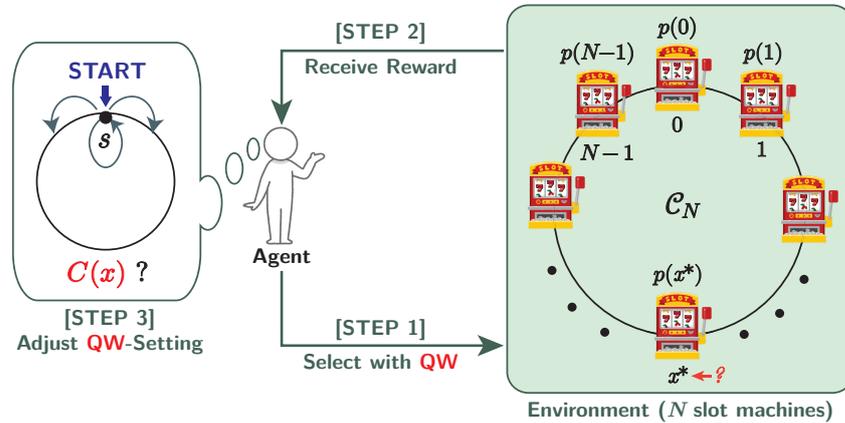
Figure 4. Single decision on the quantum-walk-based model for MAB problems

[STEP 0] QW-setting initialization

For the first decision, the settings of the quantum walk are determined as follows:

- Initial state: $|\Phi_1\rangle = |s_1\rangle \otimes |\mathrm{O}\rangle$. Here the initial position s_1 is probabilistically determined by the uniform distribution on V_N.
- Parameter of coin matrices: $\theta_1(x) = \theta^\circ \in [0, 2\pi)$ for all $x \in V_N$.

After finishing this step, the run iterates the following three steps.

[STEP 1] Quantum walk

Quantum walks are run over T time steps with the initial position s_j and the parameter $\theta_j(x)$. After running T steps of time evolution, the QWer is measured to obtain the value $\hat{x}_j \in V_N$ following probability distribution $\mu_j^{(T)}(x)$.

[STEP 2] Slot machine play

The slot machine $\hat{x}_j \in V_N$ obtained at **[STEP 1]** is played. Then, the reward ($\hat{r}_j = 1$) is obtained with probability $p(\hat{x}_j)$.

Here H- and L-values are updated. First, the H-value on $\hat{x}_j$ is incremented:

$$H_j(\hat{x}_j) = H_{j-1}(\hat{x}_j) + 1. \tag{41}$$

If $\hat{r}_j = 1$, the L-value on $\hat{x}_j$ is also incremented (otherwise, the value is maintained):

$$L_j(\hat{x}_j) = \begin{cases} L_{j-1}(\hat{x}_j) + 1 & (\text{with prob. } p(\hat{x}_j)) \\ L_{j-1}(\hat{x}_j) & (\text{with prob. } 1 - p(\hat{x}_j)) \end{cases}. \tag{42}$$

For $x \neq \hat{x}_j$, the H- and L-values are maintained:

$$H_j(x) = H_{j-1}(x), \tag{43}$$
$$L_j(x) = L_{j-1}(x). \tag{44}$$

Based on that, the $\hat{p}_j(x)$ values are updated.

[STEP 3] QW-setting adjustment

Using the new $\hat{p}_j(x)$s, the settings of quantum walks for the next decision are updated. The new initial state is defined as

$$|\Phi_{j+1}\rangle = |s_j\rangle \otimes |O\rangle, \tag{45}$$

where s_j is the provisionally best machine:

$$s_j = \arg\max_{x \in V_N} \hat{p}_j(x). \tag{46}$$

Moreover, the new parameters of the coin matrices are determined as

$$\theta_{j+1}(x) = \theta^\circ \exp(-a \cdot \hat{p}_j(x)^b) \tag{47}$$

where $a, b \geq 1$, and θ° are defined in **[STEP 0]**. Note that the θ-value is defined similarly to the q-value in the RW-based model; that is, if $\hat{p}_j(x)$ is larger, then $\theta_{j+1}(x)$ is smaller. When the θ-value at a certain position $x_{\rm L} \in V_N$ is updated, the difference between $C(x_{\rm L})$ and $C(x)$ with $x = x_{\rm L} \pm 1$ emerges, wherein $x_{\rm L}$ is often called a *defect*. If a defect exists at the initial position, the coin matrix, depending on the θ-value, controls the strength of localization there. Incidentally, when $\hat{p}_j(x)$ is large, $\theta_{j+1}(x)$ can be almost 0. If $\theta(x)$ in Equation (24) is exactly 0, then $C(x)$ is the identity matrix. This means that, if the initial position of the walker has a coin matrix with $\theta(x) = 0$, the walker is completely trapped there because the internal state is set to be $|O\rangle$. Thus, a larger empirical success probability indicates a strong localization if it is provisionally best. However, if $\hat{p}_j(x)$ is not large, this phenomenon is relaxed. In short, this θ-value plays a role corresponding to the q-value in the RW-based model; that is, it confines walkers to the desired position while concurrently affording them opportunities to depart when the current decision is uncertain. Regarding the analysis of θ-values, also see Appendix A.

After this step, the process returns to **[STEP 1]**.

4. Numerical Simulations

In this section, we give and compare simulation results for the RW- and QW-based models. Assume that the RW- and QW-based models are run in parallel K times, respectively, and each run is labeled by the set $\{1, 2, \cdots, K\}$. We indicate that the parameters are in the k-th run by a subscript next to the number of iterations; for example, the reward in the j-th decision in the k-th run is denoted by $\hat{r}_{j,k}$.

As figures-of-merit, we define quantities $M(j)$, $\rho(j)$, and CDR(j):

$$M(j) := \frac{1}{K} \sum_{k=1}^{K} \sum_{\ell=1}^{j} \hat{r}_{\ell,k}, \tag{48}$$

$$\rho(j) := \frac{1}{K} \sum_{k=1}^{K} \sum_{\ell=1}^{j} (p(x^*) - p(\hat{x}_{\ell,k})), \tag{49}$$

$$\text{CDR}(j) := \frac{1}{K} \sum_{k=1}^{K} \delta_{x^*}(\hat{x}_{j,k}). \tag{50}$$

$M(j)$ indicates the mean of *total rewards* until the j-th decision over K runs. The aim of the proposed models is to make $M(j)$ as large as possible. $\rho(j)$ is the mean of *cumulative regret* until the j-th decision over K runs. The cumulative regret is equal to the difference in expectations of total reward between the case where only the best machine is selected until the j-th decision and that of actual selections until then. CDR(j) is the *correct decision rate* of the j-th decision, which is the ratio of the number of runs in selecting the best slot

machine to the total number of runs K. Herein, δ_y for $y \in V_N$ is the delta function defined by Equation (7).

The parameter values used for this series of simulations are summarized in Table 1. The success probabilities of slot machines are given as shown in Figure 5; that is,

$$p(x) = \begin{cases} 0.9 & (x = 14) \\ 0.1 & (x = 15) \\ 0.7 & (x : \text{even except for } 14) \\ 0.5 & (x : \text{odd except for } 15) \end{cases}. \tag{51}$$

Herein, the best slot machine is $x^* = 14$. Recall that the agent cannot directly access all information regarding the success probabilities of the slot machines above. The tuples for the QW- and RW-based models are selected as one of the best performers in a certain range of parameters on each model. The details about parameter-dependencies of both models are found in Appendix A.

Table 1. Parameter values used for numerical simulation of decision making.

Parameter	Symbol	Value
Number of slot machines	N	32
Number of runs	K	500
Number of decisions for a single run	J	5000
Parameters for the QW-based model	(a, b, θ°)	$(5, 6, 5\pi/16)$
Parameters for the RW-based model	(a, b, q°)	$(9, 6, 0.5)$

The blue and orange curves in Figure 6a–c demonstrate the performances of RW- and QW-based models as the variations of the mean of total reward $M(j)$, the cumulative regret $\rho(j)$, and the maximum value of CDR over the number T of time steps of walks for single decision-making, respectively. The total reward and the cumulative regret are taken for the final decision $J = 5000$. The maximum value of CDR is taken over J decisions; that is,

$$\max(\text{CDR}) := \max_{j=1,\cdots,J} \text{CDR}(j). \tag{52}$$

For $T \geq 4$, we observe that $M(5000)$ and $\max(\text{CDR})$ of the QW-based model are larger than those of the RW-based model. On the other hand, for the cumulative regret $\rho(5000)$, the value for the QW-based model is lower than that for the RW-based model. Both results indicate that the performance of the QW-based model is superior to that of the RW-based model. You see the particular difference in the growth of $M(5000)$ and $\max(\text{CDR})$ over the variation of T between the QW- and RW-models; the gradient of the orange curves (QW) in the range of $2 \leq T \leq 8$ in $M(5000)$ and $\max(\text{CDR})$ is larger than that of the blue curve (RW). In addition, the QW-based model has higher suprema of $M(5000)$ and $\max(\text{CDR})$ than the RW-based model. Similar discussions are also made for regret $\rho(5000)$. In both the RW- and QW-models $\rho(5000)$ decreases over the variation of T in the range of $2 \leq T \leq 8$, but the gradient of the QW-model is steeper than that of the RW-one. Moreover, the QW-based model has a lower infimum than the RW-based model.

These results depend on a variety of choices; in particular, the casino setting and the parameters a and b. Indeed, the QW-based model shows a faster growth in reward for many of a and b compared to the RW-based model (see Appendix A). However, there are also settings where this relationship is reversed. We have also observed that in the limit of very large a and b, when fine-tuned to a specific casino setting, the performance of both RW and QW can grow even faster than the results shown in Figure 6. We speculate that linear spreading and localization may lose their advantage for certain settings.
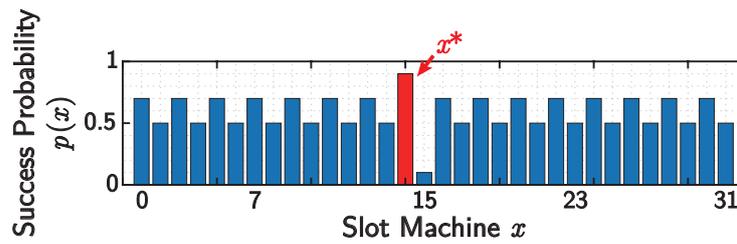
Figure 5. Success probability $p(x)$ for slot machine $x \in V_N$. The number of slot machines N is set to 32, and the best slot machine is $x^* = 14$.
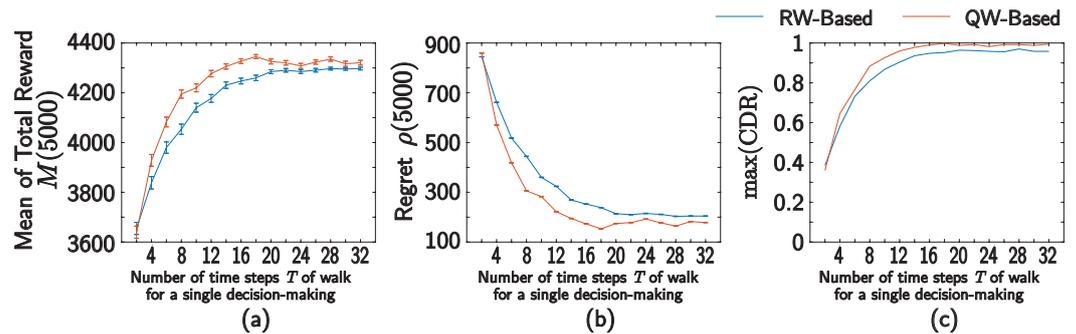
Figure 6. Comparison of (**a**) mean of total reward $M(J)$, (**b**) cumulative regret $\rho(J)$, and (**c**) the maximum value of CDR over the variation of final time step T of walks between the RW- and QW-based models. Parameters are determined as shown in Table 1.

You can see the contribution of linear spreading and localization from the behavior of variation of decision-making and probability distributions for making a decision, which is particularly apparent for smaller T. Figures 7 and 8, respectively, show the precise performances of runs of the RW- and QW-based models whose resultant total rewards $M(J)$ were almost equal to the average value. Herein, the number of time steps T is set to be 8 for each model, and the other parameters are set as in Table 1. Figure 7a,b indicate the relationship between the decision j and the selected slot machine $\hat{x}_j$ for the RW- and QW-based models, respectively. From this figure, you can see that the decision-making in the QW-based model almost converges to $x = x^*$ near $j = 1200$, while that in the RW-based model does so near $j = 1400$. This means that exploration in the QW-based model is more successful than that in the RW-based model. Linear spreading makes the probability distribution of QWs wider, whose variance is larger than that of RWs, which results in faster exploration of the QW-based model. Moreover, the behavior of the QW-based model after finding $x = x^*$ is more stable than that of the RW-based model, which indicates that the QW-based model also realizes more effective exploitation than the RW-based one for this set of parameters.

These behaviors are interpreted by the variations of the probability distributions of the RW($\nu_j^{(T)}$)- and QW($\mu_j^{(T)}$)-based models over decision j shown in Figure 8a,b, respectively. You see that, for smaller j, the probability of QW is more widely distributed than that of RW, although the values are quite small except for a certain position. It is important that the probabilities are distributed in a wider range even if they are quite small because it indicates that the agent has more selections, which is crucial to realize exploration. As a result, the QW-based model can obtain a high measurement probability of the best slot machine at $j = 1200$ at the latest, while the RW-based model does at $j = 1400$, which corresponds to the convergence of the decision-making shown in Figure 7. After beginning the concentrated investment to $x = x^*$, the measurement probability of walkers on the QW is almost 1, while the corresponding probability on the RW is around 0.9, which shows that strong localization occurs on vertex x^* after finding the slot machine there. This phenomenon contributes to exploitation; as you see in Figure 7, the QW-based model after finding the best slot machine is much more likely to select it than the RW-based one, and

we can understand that it comes from the difference in probability distributions between the two models.
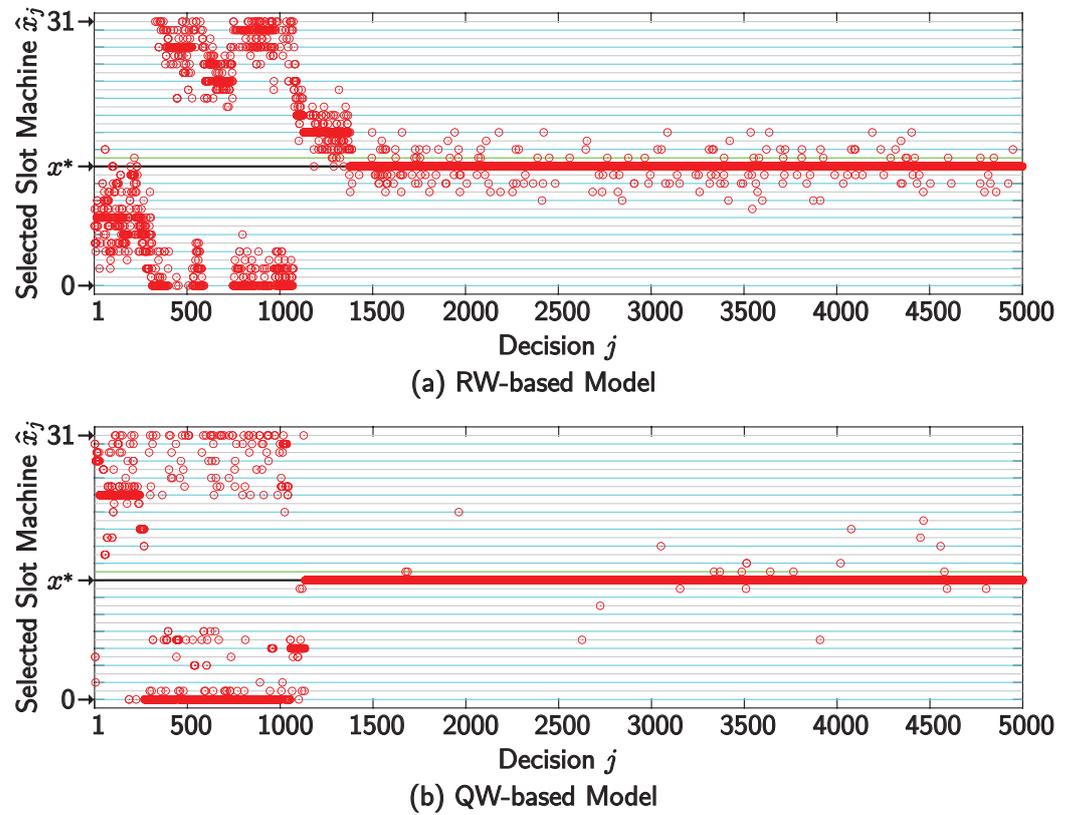
Figure 7. The red markers show the variation of the selected slot machine $\hat{x}_j$ over decision j for single runs of the (**a**) RW- and (**b**) QW-based models. For both settings, the number of time steps T is set to 8, and other parameters are determined as in Table 1. Each run is selected as the one whose resultant total rewards $M(J)$ were almost equal to the average value: $M(J) = 4063$ in (**a**), and $M(J) = 4200$ in (**b**). The black, sky blue, gray, and light green lines indicate the slot machines whose success probabilities are 0.9, 0.7, 0.5, and 0.1, respectively.
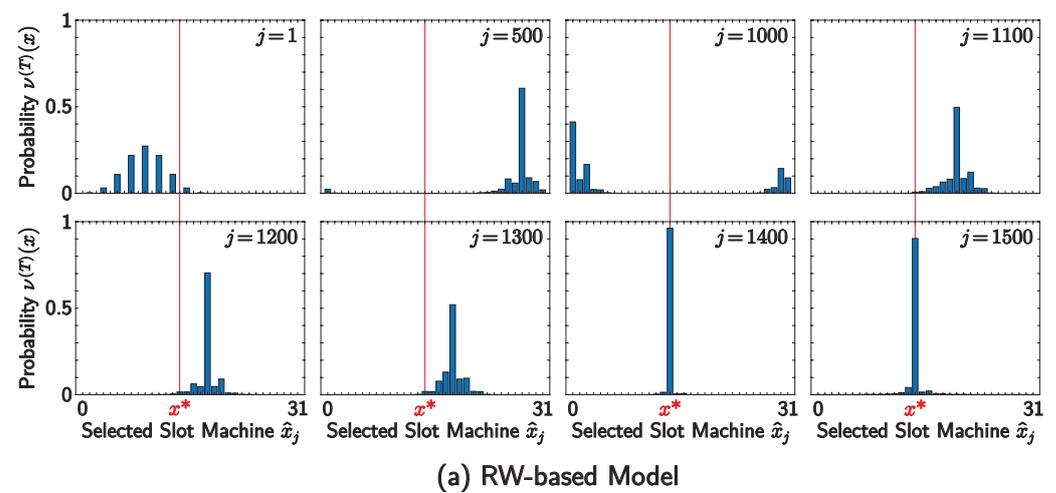
Figure 8. *Cont.*
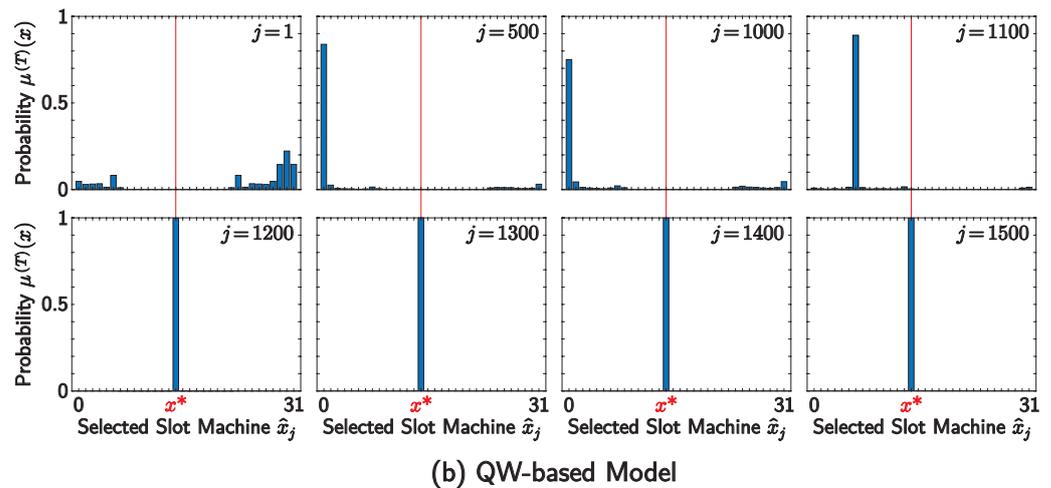
(b) QW-based Model

Figure 8. The probability distributions regarding the selected slot machine $\hat{x}_j$ where walkers exist after T steps of walk in the j-th decision with $j = 1$, 500, 1000, 1100, 1200, 1300, 1400, 1500 for single runs of the (**a**) RW- and (**b**) QW-based models. The settings are the same as in Figure 7; that is, for both settings, the number of time steps T is set to 8, other parameters are determined as in Table 1, and each run is selected as the one whose resultant total rewards $M(J)$ were almost equal to the average value: $M(J) = 4063$ in (**a**), and $M(J) = 4200$ in (**b**).

5. Conclusions and Discussion

This paper has proposed new solution schemes for multi-armed bandit (MAB) problems using random walks (RWs) and quantum walks (QWs). We demonstrated that we could find parameter regimes where the QW-based model performs better than the RW-based model by addressing the exploration–exploitation dilemma by utilizing a unique property of QWs, i.e., the coexistence of linear spreading and localization. Our idea was to combine exploration with linear spreading and exploitation with localization. By utilizing linear spreading, we expect the QWs to cover the whole environment to prevent it from missing some slot machines. In addition, by applying localization, we expect the QWs to identify the slot machine that should be recommended with a high probability distribution. Indeed, we showed that, under some settings, linear spreading contributes to exploring the environment and quickly finding the best slot machine, and localization contributes to exploiting the best slot machine more frequently.

The positive results obtained in this study open the possibility for further extensions of this approach. First, can we apply this algorithm to the case of multi-agent systems such as competitive or adversarial bandit problems [46–48] with some revision? Especially for the QW-based model, we will examine its application to the use of coin matrices implemented by multiple registers or to drive walkers on a torus. Moreover, there are possibilities for constructing application models in the single-agent case. For example, we may construct an evolved version of the QW-based model, including a quantum version of the optimal stopping problem.

Moreover, analyses of our models are also important. The performances obtained in Section 4 should depend on the number of slot machines (i.e., vertices) N, the true success probability $p(x)$, and the parameter settings (a, b, q°) or (a, b, θ°). Obtaining theoretical formulae for the figures-of-merit would be desirable, which would make our results more confident. Specifically, it is not immediately evident, given the parameter configurations outlined in this paper, that the QW-based model outperforms the RW-based model in general. Our investigations have focused solely on pairs (a, b) with small values; in the event that larger values are assigned to either a or b, the RW-based model can exhibit superior performance compared to any QW-based model (assuming the parameters are chosen correctly for a specific casino setting). Indeed, we have confirmed that the RW-based model with tuple $(a, b, q^\circ) = (50, 20, 0.5)$ performs as well as the QW-based one with

tuple $(a, b, \theta^\circ) = (50, 20, 29\pi/64)$, and in both results, the mean of total reward is around 4320 even in setting the number of time steps T to 8 (better than the results for small a and b as shown in Figure 6). However, sufficiently addressing the parameter dependency and its interplay with the casino settings is a highly complex problem at this stage. We provide our existing analysis in Appendix A.

Several factors contribute to the complexity of a rigid mathematical treatment of this problem, but the one that should be remarked on is the position-dependency of the coin matrices in the QW-based model. Solving QW-based models with site-dependent coins is very difficult, in general. While some studies have addressed this matter for the case where the coin matrix only on the origin differs from the others [6,22] or that the coin matrices are controlled by the trigonometric function whose input is in proportion to the label position [12,14,21], the generalized case remains an open problem. To conduct a thorough analysis of this model, it is necessary to accumulate analytical results for site-dependent quantum walks over an extended period of time.

Appendix A. Parameter-Dependencies of RW- and QW-Based Models

We analyze the (a, b, q°)-dependency of the RW-based model and the (a, b, θ°)-dependency of the QW-based model.

First, we introduce the function $f : [0, 1] \to \mathbb{R}$ defined as follows:

$$f(u) = c \exp(-au^b) \tag{A1}$$

with $a, b \geq 1$ and $c \geq 0$. Variance u corresponds to empirical success probability $\hat{p}_j(x)$ in both cases, and $f(u)$ to $q_{j+1}(x)$ and $\theta_{j+1}(x)$ in Equations (17) and (47), respectively. Note that $q_{j+1}(x)$ and $\theta_{j+1}(x)$ are controlled by position x through $\hat{p}_j(x)$; in that sense, they are the functions of the empirical success probability. Additionally, parameter c is represented by q° and θ° in the RW- and QW-based models, respectively.

By the property of exponential functions, $f(u)$ monotonically decreases; the maximum and minimum of $f(u)$ are $f(0) = c$ and $f(1) = c \exp(-a)$, respectively. This indicates that, if c is fixed, the minimum of $f(u)$ is determined by a and smaller in larger a as shown in Figure A1a. Furthermore, if both a and c are fixed, the maximum and minimum of $f(u)$ are constant; b solely governs how $f(u)$ decreases along with the growth of u. Precisely, larger b makes the gradient of $f(u)$ negatively larger as shown in Figure A1b.

In the following, we fix the number T of time steps of the walk for single decision-making to 32; the tendency is common for any T. The numbers of slot machines, runs, and decisions for a single run are equal to those of the simulation in Section 4; that is, $(N, K, J) = (32, 500, 5000)$.
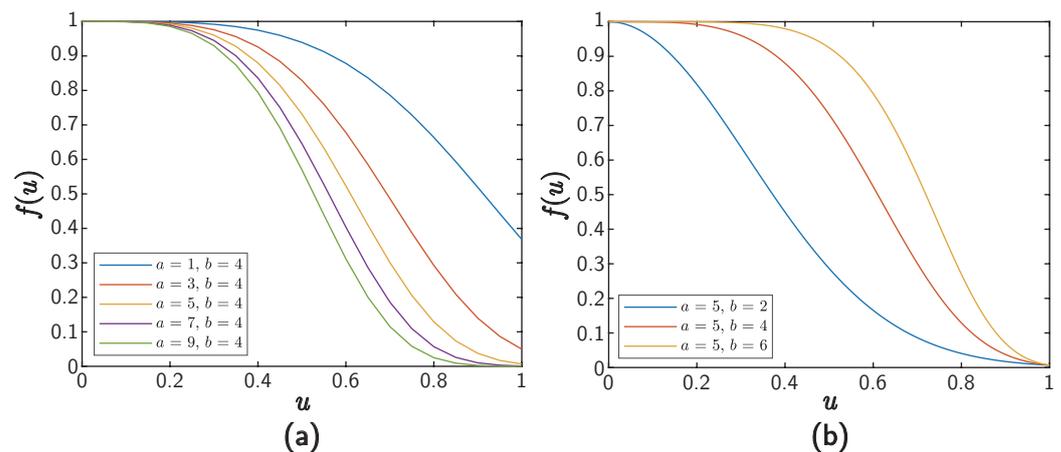
Figure A1. Variance of function $f(u)$. (**a**) Difference among the cases of $a = 1, 3, 5, 7, 9$ when b is fixed to 4. (**b**) Difference among the cases of $b = 2, 4, 6$ when a is fixed to 5. For both figures, c is fixed to 1.

In the RW-based model, $f(u)$ directly controls the behavior of walkers as the q-value given by Equation (17), which indicates the transition probability clockwise and anti-clockwise. By iterating the decision-making, the empirical success probability $\hat{p}_j(x)$ is likely to be $p(x)$ given by Equation (51). Especially, the algorithm should experience many plays with slot machine x with $p(x) = 0.9$ or 0.7; that is, considering $f(0.9)$ and $f(0.7)$ is particularly important. First, $f(0.9)$ should be smaller because one demands walkers stay at $x = x^*$ once they find the slot machine due to exploitation. To realize it, larger a and smaller b are desirable. On the other hand, $f(0.7)$ should be larger because one requires walkers to be active and explore more. To realize it, the opposite of the previous condition is desirable; i.e., smaller a and larger b. Here exploration–exploitation trade-off emerges as the balance of a and b. That is, it is required to choose a and b with an appropriate proportion (and this balance depends on the casino setting). If a is large, this leads to a small $f(u)$ on slots with high $\hat{p}_j(x)$, making it likely for the walker to stay on that slot and play that machine. This way, one can realize the exploitation of the best slot machine, but it may also lead to the over-exploitation of a non-best slot machine. Importantly, if b is large, one can save walkers from sticking to a non-best slot machine, but this also may result in over-exploration even after finding the best slot machine.

Figure A2 demonstrates the performance of the RW-based model depending on tuple (a, b, q°). You see that the larger b becomes, the larger a is required for improvement of the performance, which matches the analysis of $f(u)$ above. If the value of a is small under the large value of b, the algorithm conducts exploration too much. On the other hand, setting a too large is not demanded; this case results in over-exploitation. Indeed, you see that the performance of $(a, b, q^\circ) = (9, 2, 0.5)$ is worse than that of $(a, b, q^\circ) = (7, 2, 0.5)$. In our results, the best performance is obtained for the case of $(a, b) = (9, 6)$, where max(CDR) is over 0.95 for $q^\circ \geq 0.25$ in Figure A2(c3). In Section 4, we selected tuple $(a, b, q^\circ) = (9, 6, 0.5)$ as the representative of this range for the analysis in the main manuscript.

In the QW-based model, $f(u)$ contributes to controlling the behavior of walkers as the θ-value given by Equation (47), which plays the role of the phase of the coin matrix. The measurement probability on the defect strongly depends on the coin matrices on the defect and its neighborhood, and thus, θ° pertains to the performance much more than q° for the RW-based model. Moreover, the measurement probability of the defect is not monotonic over θ-values, whose analysis is still open. However, what we can say from $f(u)$ is that the exploitation will be frequently conducted if a is small b is large, which is the same as the RW-base model because the θ-value near 0 distributes the probability of almost 1 on the defect as mentioned on **[STEP 3]** in Section 3.2.
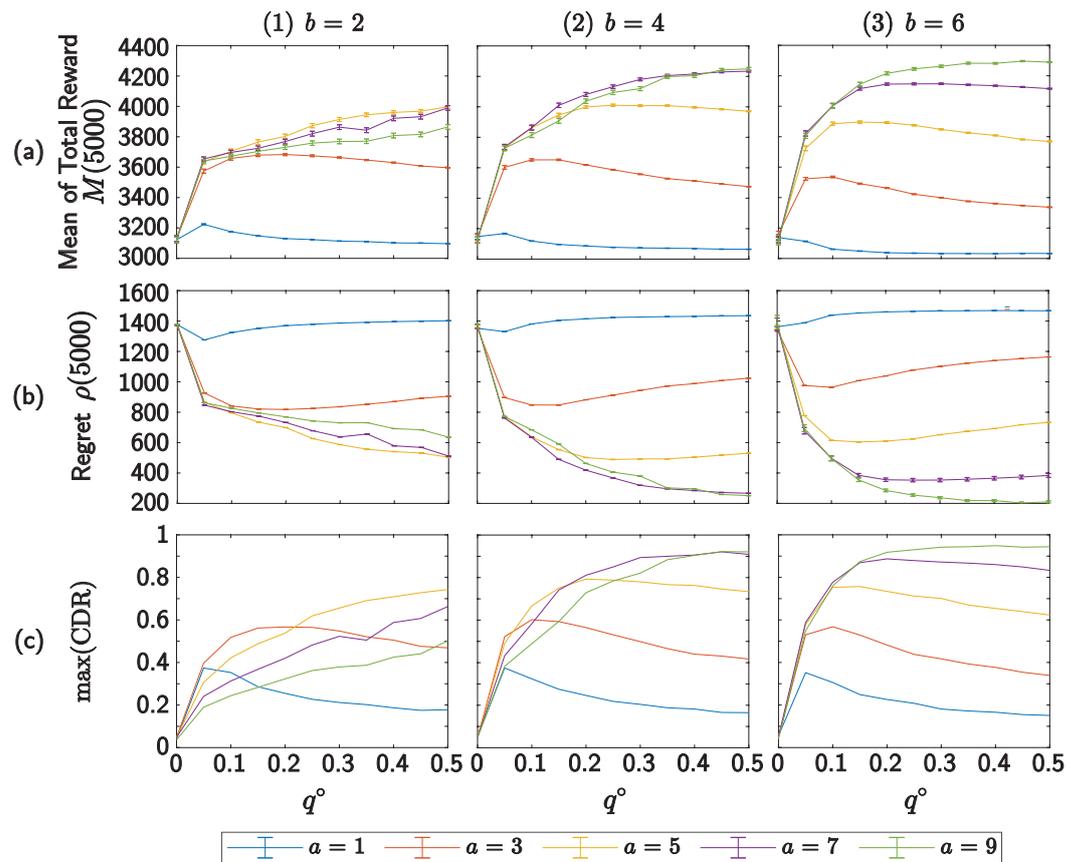
Figure A2. Comparison of the (**a**) mean of total reward $M(J)$, (**b**) cumulative regret $\rho(J)$, and (**c**) the maximum value of CDR on the RW-based model.

Figure A3 demonstrates the performance of the QW-based model depending on tuple (a, b, θ°). You see that some cases exhibit extreme results; the results can be both almost the same as the worst one in the RW-based model and better than any in the RW-based model. We observe that when setting $a = 1$, θ° makes the performance worse with violent oscillations. Especially, the case of $(a, b, \theta^\circ) = (1, 6, 29\pi/64)$ performs worst due to over-exploitation, which is almost the same as the worst one in the RW-based model. For other cases of a, performance tends to improve over the growth of θ°. Especially in the case of $b = 6$, max(CDR) is almost 1 for the sufficiently large θ° as shown in Figure A3(c3), which is better than any case in the RW-based model. This indicates that exploration should be introduced by large b-values. Particularly, the case of $(a, b) = (5, 6)$ realizes performs very well. Based on the above, it is important to choose the appropriate tuples for obtaining the result of the QW-based model that outperforms the RW-based model. In Section 4, we selected tuple $(a, b, \theta^\circ) = (5, 6, 5\pi/16)$ as the representative of this case.

As mentioned in the conclusions of the main manuscript, we have confirmed that the RW-based model with tuple $(a, b, q^\circ) = (50, 20, 0.5)$ performs as well as the QW-based one with tuple $(a, b, \theta^\circ) = (50, 20, 29\pi/64)$, and in both results, the mean of total reward is around 4320 even in setting the number of time steps T to 8 (better than the results for small a and b as shown in Figure 6. In this case, $f(u)$ is basically a step-function with $f(0.9) \simeq 0$ and $f(0.7) \simeq 1$, leading to walkers basically ignoring any position except the one optimal slot. It should be remarked that the discussion of $f(u)$ above strongly depends on the setting of success probability $p(x)$ given by Equation (51). For example, if we consider the setting where even the best slot machine has a success probability of 0.7, and the other ones have lower success probabilities, then $f(0.7)$ should be smaller. As mentioned in Section 5, the dependency on the settings regarding slot machines is a heavily complex problem and needs detailed discussions as future tasks.
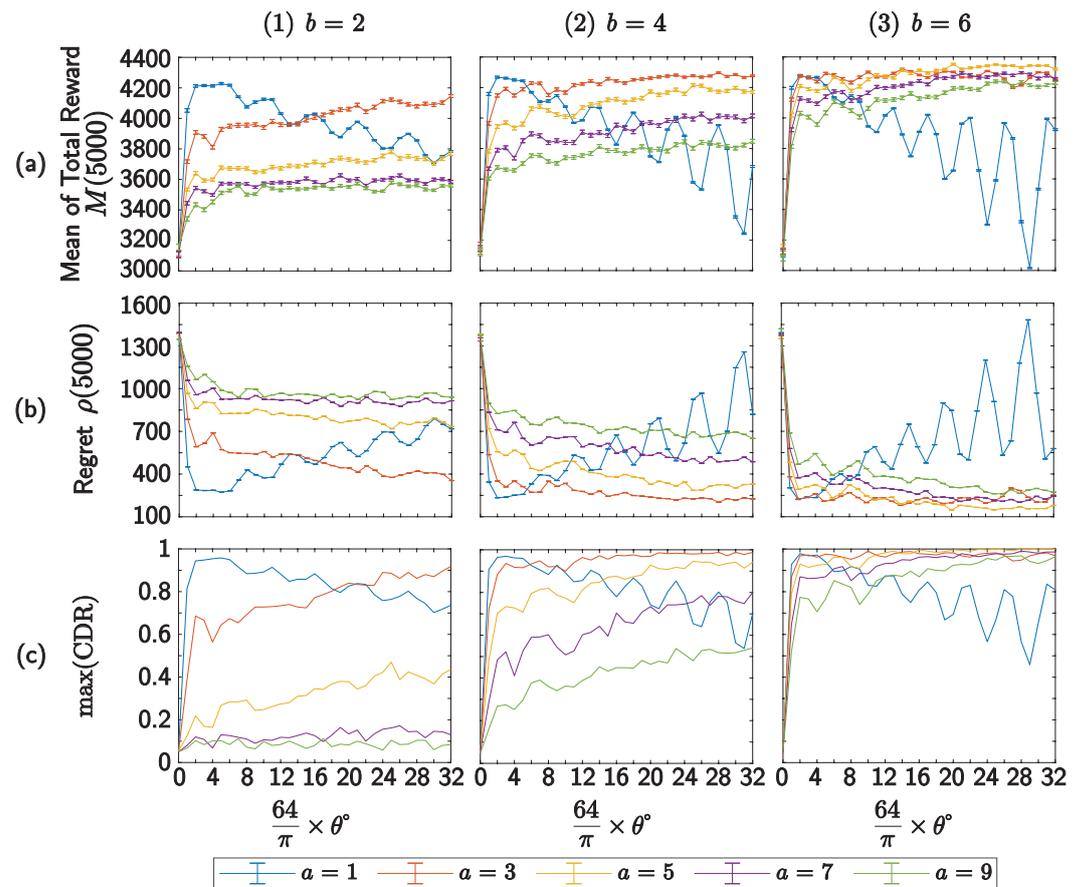
Figure A3. Comparison of the (**a**) mean of total reward $M(J)$, (**b**) cumulative regret $\rho(J)$, and (**c**) the maximum value of CDR on the QW-based model.

References

1. Konno, N. Quantum walks. In *Quantum Potential Theory*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 309–452.
2. Kempe, J. Quantum random walks: An introductory overview. *Contemp. Phys.* **2003**, *44*, 307–327. [CrossRef]
3. Venegas-Andraca, S.E. Quantum walks: A comprehensive review. *Quantum Inf. Process.* **2012**, *11*, 1015–1106. [CrossRef]
4. Kendon, V. Decoherence in quantum walks—A review. *Math. Struct. Comput. Sci.* **2007**, *17*, 1169–1220. [CrossRef]
5. Inui, N.; Konno, N.; Segawa, E. One-dimensional three-state quantum walk. *Phys. Rev. E* **2005**, *72*, 056112. [CrossRef]
6. Konno, N.; Łuczak, T.; Segawa, E. Limit measures of inhomogeneous discrete-time quantum walks in one dimension. *Quantum Inf. Process.* **2013**, *12*, 33–53. [CrossRef]
7. Ambainis, A.; Bach, E.; Nayak, A.; Vishwanath, A.; Watrous, J. One-dimensional quantum walks. In Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, Hersonissos, Greece, 6–8 July 2001; pp. 37–49.
8. Gudder, S.P. *Quantum Probability*; Elsevier Science: Amsterdam, The Netherlands, 1988.
9. Aharonov, Y.; Davidovich, L.; Zagury, N. Quantum random walks. *Phys. Rev. A* **1993**, *48*, 1687–1690. [CrossRef]
10. Konno, N. Quantum random walks in one dimension. *Quantum Inf. Process.* **2002**, *1*, 345–354. [CrossRef]
11. Konno, N. A new type of limit theorems for the one-dimensional quantum random walk. *J. Math. Soc. Jpn.* **2005**, *57*, 1179–1195. [CrossRef]
12. Linden, N.; Sharam, J. Inhomogeneous quantum walks. *Phys. Rev. A* **2009**, *80*, 052327. [CrossRef]
13. Konno, N. Localization of an inhomogeneous discrete-time quantum walk on the line. *Quantum Inf. Process.* **2010**, *9*, 405–418. [CrossRef]
14. Shikano, Y.; Katsura, H. Localization and fractality in inhomogeneous quantum walks with self-duality. *Phys. Rev. E* **2010**, *82*, 031122. [CrossRef]
15. Sunada, T.; Tate, T. Asymptotic behavior of quantum walks on the line. *J. Funct. Anal.* **2012**, *262*, 2608–2645. [CrossRef]
16. Bourgain, J.; Grünbaum, F.; Velázquez, L.; Wilkening, J. Quantum recurrence of a subspace and operator-valued Schur functions. *Commun. Math. Phys.* **2014**, *329*, 1031–1067. [CrossRef]
17. Suzuki, A. Asymptotic velocity of a position-dependent quantum walk. *Quantum Inf. Process.* **2016**, *15*, 103–119. [CrossRef]
18. Sadowski, P.; Miszczak, J.A.; Ostaszewski, M. Lively quantum walks on cycles. *J. Phys. Math. Theor.* **2016**, *49*, 375302. [CrossRef]
19. Godsil, C.; Zhan, H. Discrete-time quantum walks and graph structures. *J. Comb. Theory Ser. A* **2019**, *167*, 181–212. [CrossRef]

20. Cedzich, C.; Fillman, J.; Geib, T.; Werner, A. Singular continuous Cantor spectrum for magnetic quantum walks. *Lett. Math. Phys.* **2020**, *110*, 1141–1158. [CrossRef]
21. Ahmad, R.; Sajjad, U.; Sajid, M. One-dimensional quantum walks with a position-dependent coin. *Commun. Theor. Phys.* **2020**, *72*, 065101. [CrossRef]
22. Kiumi, C. Localization of space-inhomogeneous three-state quantum walks. *J. Phys. A Math. Theor.* **2022**, *55*, 225205. [CrossRef]
23. Konno, N. A new time-series model based on quantum walk. *Quantum Stud. Math. Found.* **2019**, *6*, 61–72. [CrossRef]
24. Asbóth, J.K.; Obuse, H. Bulk-boundary correspondence for chiral symmetric quantum walks. *Phys. Rev. B* **2013**, *88*, 121406. [CrossRef]
25. Obuse, H.; Asbóth, J.K.; Nishimura, Y.; Kawakami, N. Unveiling hidden topological phases of a one-dimensional Hadamard quantum walk. *Phys. Rev. B* **2015**, *92*, 045424. [CrossRef]
26. Matsuoka, L.; Ichihara, A.; Hashimoto, M.; Yokoyama, K. Theoretical study for laser isotope separation of heavy-element molecules in a thermal distribution. In Proceedings of the International Conference Toward and Over the Fukushima Daiichi Accident (GLOBAL 2011), Chiba, Japan, 11–16 December 2011; No. 392063.
27. Ichihara, A.; Matsuoka, L.; Segawa, E.; Yokoyama, K. Isotope-selective dissociation of diatomic molecules by terahertz optical pulses. *Phys. Rev. A* **2015**, *91*, 043404. [CrossRef]
28. Wang, J.; Manouchehri, K. *Physical Implementation of Quantum Walks*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 10.
29. Ide, Y.; Konno, N.; Matsutani, S.; Mitsuhashi, H. New theory of diffusive and coherent nature of optical wave via a quantum walk. *Ann. Phys.* **2017**, *383*, 164–180. [CrossRef]
30. Yamagami, T.; Segawa, E.; Konno, N. General condition of quantum teleportation by one-dimensional quantum walks. *Quantum Inf. Process.* **2021**, *20*, 224. [CrossRef]
31. Wang, Y.; Shang, Y.; Xue, P. Generalized teleportation by quantum walks. *Quantum Inf. Process.* **2017**, *16*, 221. [CrossRef]
32. Vlachou, C.; Krawec, W.; Mateus, P.; Paunković, N.; Souto, A. Quantum key distribution with quantum walks. *Quantum Inf. Process.* **2018**, *17*, 288. [CrossRef]
33. Childs, A.M. Universal computation by quantum walk. *Phys. Rev. Lett.* **2009**, *102*, 180501. [CrossRef] [PubMed]
34. Childs, A.M.; Gosset, D.; Webb, Z. Universal computation by multiparticle quantum walk. *Science* **2013**, *339*, 791–794. [CrossRef]
35. Meyer, D.A. From quantum cellular automata to quantum lattice gases. *J. Stat. Phys.* **1996**, *85*, 551–574. [CrossRef]
36. Meyer, D.A. On the absence of homogeneous scalar unitary cellular automata. *Phys. Lett. A* **1996**, *223*, 337–340. [CrossRef]
37. Szegedy, M. Quantum speed-up of Markov chain based algorithms. In Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, Rome, Italy, 17–19 October 2004; pp. 32–41.
38. Patel, A.; Raghunathan, K.; Rungta, P. Quantum random walks do not need a coin toss. *Phys. Rev. A* **2005**, *71*, 032347. [CrossRef]
39. Portugal, R.; Santos, R.A.; Fernandes, T.D.; Gonçalves, D.N. The staggered quantum walk model. *Quantum Inf. Process.* **2016**, *15*, 85–101. [CrossRef]
40. Konno, N.; Portugal, R.; Sato, I.; Segawa, E. Partition-based discrete-time quantum walks. *Quantum Inf. Process.* **2018**, *17*, 100. [CrossRef]
41. Robbins, H. Some aspects of the sequential design of experiments. *Bull. Am. Math. Soc.* **1952**, *58*, 527–535. [CrossRef]
42. Daw, N.D.; O'doherty, J.P.; Dayan, P.; Seymour, B.; Dolan, R.J. Cortical substrates for exploratory decisions in humans. *Nature* **2006**, *441*, 876–879. [CrossRef]
43. Sarkar, R.S.; Mandal, A.; Adhikari, B. Periodicity of lively quantum walks on cycles with generalized Grover coin. *Linear Algebra Its Appl.* **2020**, *604*, 399–424. [CrossRef]
44. Han, Q.; Bai, N.; Kou, Y.; Wang, H. Three-state quantum walks on cycles. *Int. J. Mod. Phys. B* **2022**, *36*, 2250075. [CrossRef]
45. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, Philadelphia, PA, USA, 22–24 May 1996; pp. 212–219.
46. Lai, L.; El Gamal, H.; Jiang, H.; Poor, H.V. Cognitive medium access: Exploration, exploitation, and competition. *IEEE Trans. Mob. Comput.* **2010**, *10*, 239–253.
47. Kim, S.J.; Naruse, M.; Aono, M. Harnessing the computational power of fluids for optimization of collective decision making. *Philosophies* **2016**, *1*, 245–260. [CrossRef]
48. Auer, P.; Cesa-Bianchi, N.; Freund, Y.; Schapire, R.E. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In Proceedings of the IEEE 36th Annual Foundations of Computer Science, Milwaukee, WI, USA, 23–25 October 1995; pp. 322–331.