Materials Science

http://www.aimspress.com/journal/Materials/

AIMS Materials Science, 3(1): 245–259.
DOI: 10.3934/matersci.2016.1.245
Received: 12 July 2015
Accepted: 22 January 2016
Published: 25 February 2016

*Research article*

# Decision maker based on atomic switches

**Song-Ju Kim[1]*, Tohru Tsuruoka[1], Tsuyoshi Hasegawa[2], Masashi Aono[3,4], Kazuya Terabe[1] , and Masakazu Aono[1]**

[1] WPI Center for Materials Nanoarchitectonics, National Institute for Materials Science, 1-1 Namiki, Tsukuba, Ibaraki 305–0044, Japan

[2] Department of Applied Physics, Waseda University, 3-4-1 Ookubo, Shinjuku-ku, Tokyo 169-8555, Japan

[3] Earth-Life Science Institute, Tokyo Institute of Technology, Tokyo 152–8550, Japan

[4] PRESTO JST, Japan

* **Correspondence:** E-mail: KIM.Songju@nims.go.jp; Tel: +81-29-851-3351; Fax: +81-29-860-4790.

**Abstract:** We propose a simple model for an atomic switch-based decision maker (ASDM), and show that, as long as its total number of metal atoms is conserved when coupled with suitable operations, an atomic switch system provides a sophisticated "decision-making" capability that is known to be one of the most important intellectual abilities in human beings. We considered a popular decision-making problem studied in the context of reinforcement learning, the multi-armed bandit problem (MAB); the problem of finding, as accurately and quickly as possible, the most profitable option from a set of options that gives stochastic rewards. These decisions are made as dictated by each volume of precipitated metal atoms, which is moved in a manner similar to the fluctuations of a rigid body in a tug-of-war game. The "tug-of-war (TOW) dynamics" of the ASDM exhibits higher efficiency than conventional reinforcement-learning algorithms. We show analytical calculations that validate the statistical reasons for the ASDM to produce such high performance, despite its simplicity. Efficient MAB solvers are useful for many practical applications, because MAB abstracts a variety of decision-making problems in real-world situations where an efficient trial-and-error is required. The proposed scheme will open up a new direction in physics-based analog-computing paradigms, which will include such things as "intelligent nanodevices" based on self-judgment.

**Keywords:** natural computing; atomic switch; tug-of-war dynamics; amoeba-inspired computing; multi-armed bandit problem; reinforcement learning

## 1. Introduction

Many natural phenomena, including the physical, chemical, and biological, can be viewed as computing processes [1, 2, 3]. Inspired by such natural phenomena, many search algorithms for combinatorial optimization problems have been proposed to quickly obtain high quality solutions, such as simulated annealing [4], neural networks [5], genetic algorithms [6], and DNA computing [7]. In this paper, we propose a new computing architecture that utilizes the desirable physical properties of "atomic switches" [8, 9] to solve a decision-making problem. If this architecture could be implemented in physical devices in such a way that their computing processes are elegantly coupled with their underlying physics, we would be able to utilize their abilities to make accurate and speedy decisions in uncertain environments [10].

Decision-making is one of the most important intellectual abilities of human beings. In the context of reinforcement learning, the multi-armed bandit problem (MAB) was originally described by Robbins [11], although the essence of the problem had been studied earlier by Thompson [12]. Suppose there are $M$ slot machines, each of which returns a reward; for example, coins, with a certain probability density function (PDF) that is unknown to a player. Let us consider a minimal case: two machines $A$ and/or $B$ give rewards with individual PDF whose mean reward is $\mu_A$ and $\mu_B$, respectively. The player makes a decision on which machine to play at each trial, trying to maximize the total reward obtained after repeating several trials. The MAB is used to determine the optimal strategy for playing machines as accurately and quickly as possible by referring to past experience.

The optimal strategy, called the "Gittins index," is known only for a limited class of problems in which the reward distributions are assumed to be known to the players [13, 14]. Even in this limited class, in practice, computing the Gittins index becomes intractable for many cases. For the algorithms proposed by Agrawal and Auer et al., another index was expressed as a simple function of the reward sums obtained from the machines [15, 16]. In particular, the "upper confidence bound 1 (UCB1) algorithm" for solving MABs is used worldwide in many practical applications [16]. The MAB is formulated as a mathematical problem without loss of generality and, as such, is related to various stochastic phenomena. In fact, many application problems in diverse fields, such as communications (cognitive networks [17, 18]), commerce (advertising on the web [19]), entertainment (Monte-Carlo tree search, which is used for computer games [20, 21]), can be reduced to MABs.

A proposal was made about ten years ago for a conceptually novel switching device called the "atomic switch," which is based on metal ion migration and electrochemical reactions in solid electrolytes (SEs) [8, 9]. Because its resistance state is controlled continuously by the movement of a limited number of metal ions/atoms, the atomic switch can be regarded as a physics-based analog-computing element. Nanoarchitectonic designs using such atomic switches have recently been proposed for natural computing [22, 23]. In this paper, using two atomic switches that interact with each other, we show that a physical constraint, the conservation law, allows for the efficient solving of decision-making problems.

This paper consists of five sections. In Sec. 2, a brief summary of an atomic-switch-based decision maker (ASDM) model is given, and its operating principle is described. The theoretical analyses of the dynamics underlying the ASDM are presented in Sec. 3. The simulation results based on the ASDM and the SOFTMAX algorithm, which is a well-known algorithm for solving the MABs [24], are compared in Sec. 4. Section 5 presents the conclusion with a short discussion.

## 2. Model

Recently, Kim et al. proposed a MAB solver [25, 26] that reflects the behavior of a single-celled amoeboid organism (the true slime mold *P. polycephalum*), which maintains a constant intracellular-resource volume while collecting environmental information by concurrently expanding and shrinking its pseudopod-like terminal parts. In this bio-inspired algorithm, the decision-making function is derived from its underlying "tug-of-war (TOW) game"-like dynamics. The physical constraint in TOW dynamics, the conservation law for the volume of the amoeboid body, entails a nonlocal correlation among the terminal parts. That is, a volume increment in one part is immediately compensated for by volume decrement(s) in another part(s). Owing to the nonlocal correlation, the TOW dynamics exhibit higher performance than other well-known algorithms, such as the modified $\epsilon$-GREEDY algorithm and the modified SOFTMAX algorithm [25, 26, 10]. These observations suggest that efficient decision-making devices could be implemented using any physical object as long as it holds some physical conservation law. In fact, Kim et al. theoretically and experimentally demonstrated that optical energy-transfer dynamics between quantum dots, in which energy is conserved, can be used for the implementation of TOW dynamics [27, 28, 29].

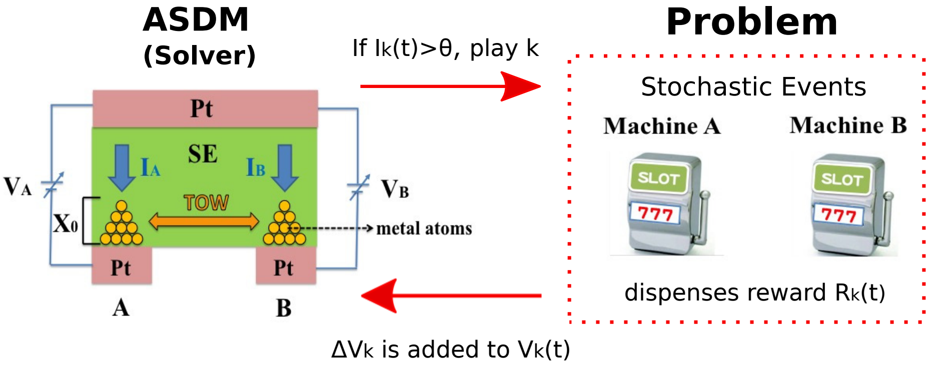

**Figure 1. The ASDM using gapless-type atomic switches. The ASDM decides which machine (*A* or/and *B*) is to be played at time *t* according to whether the current $I_k$ is larger than $\theta$ or not.**

Here, we propose a simplified model for an ASDM based on the TOW dynamics. The ASDM consists of two atomic switches (named *A* and *B*) located close to each other, as shown in Figure 1, in which a SE including metal ions is sandwiched between one Pt electrode on the top side and two Pt electrodes on the bottom side. The electrode material does not have to be Pt, but it should be an inert metal. Each atomic switch is operated in a metal/ionic conductor/metal (MIM) configuration, which can be referred to as a "gapless-type atomic switch [30]" because each MIM switch can form a metal filament between the top and bottom electrodes by precipitation on an inert electrode. In the initial state, we consider the situation where metal ions are distributed uniformly in the SE and the total number of metal ions is constant. First, a bias voltage of $-V_0$ is applied to both switches *A* and *B*, i.e., to the bottom Pt electrodes relative to the top Pt electrode ($V_A$ and $V_B = -V_0$), and the current $I_k$ passing through the respective switch is measured with a time step increment of $t_s$. Under these circumstances,

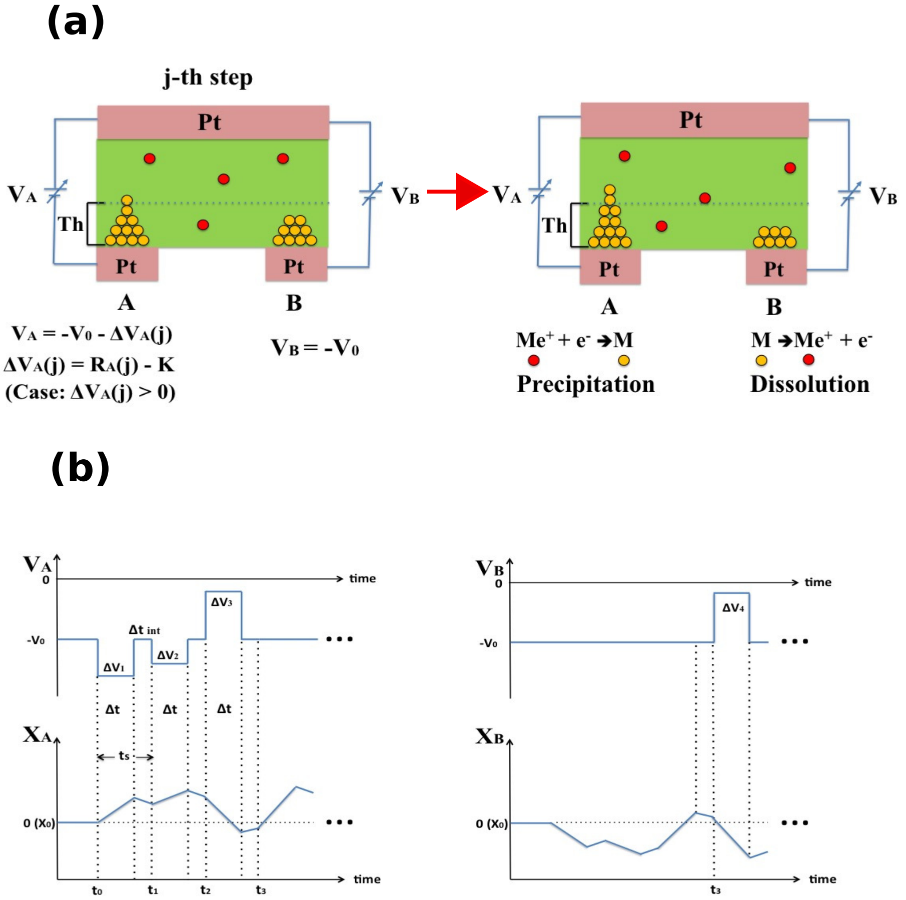

**Figure 2. (a) Tug-of-war (TOW) dynamics in the ASDM. (b) An example of expected behavior of the ASDM, showing the relationship between voltage $V_k$ and displacement $X_k$. Here, added voltage $\Delta V_k(j)$ is determined by each reward $R_k(j)$ (Eq.(1)) at play $j$ ($I_k > \theta$). The ASDM selected machine $A$, $A$, $A$, $B$, $\cdots$ in this figure.**

metal ions migrate to the respective bottom electrodes and are precipitated on them by the reduction reaction ($Me^+ + e^- \to M$). Here, we assume that the same amounts of metal atoms are precipitated on both bottom electrodes, and the heights of these initial precipitations are defined by $X_0$, as shown in Figure 1. Because of the stochastic nature of precipitation phenomena, the current $I_k$ should fluctuat with time even after reaching the equilibrium state.

In the next procedure, the ASDM compares the measured current $I_k$ of both switches $A$ and $B$ with a threshold $\theta$. If the current $I_k$ becomes larger than $\theta$ at a certain time step, the ASDM chooses the corresponding slot machine(s) $k$ (A and/or B), and sends this information to the problem side. On the problem side, a reward $R_k(j)$ generated from each "unknown PDF" (the mean reward $\mu_k$ is also supposed to be unknown) is obtained as a result of stochastic events by playing the machine, where $j$ is the step number. Depending on the reward, the added voltage $\Delta V_k(j)$ is determined by

$$\Delta V_k(j) = R_k(j) - K, \qquad (1)$$

and is returned to the ASDM. Here, $R_k(j)$ is an arbitrary real value, and $K$ is a parameter that will be described in detail later on in this paper. As a result, the total voltage applied to the respective switch at the j-th step is changed to

$$V_k = -(V_0 + \Delta V_k(j)) . \tag{2}$$

In response to the change in the applied voltages, the height of precipitations in switches $A$ and $B$ may vary, and each displacement from the initial height $X_0$ at time $t$ is given by $X_k(t)$ ($k \in \{A, B\}$). The total height becomes $X_0 + X_k(t)$. This procedure is repeated for the number of steps specified by the problem side or until a metal filament in either switch is formed between the top and bottom electrodes.

During the operation, the following conditions are assumed:

1. The SE is nearly empty of metal ions to be precipitated. Owing to the conservation law of the total number of metal ions/atoms, precipitation of metal atoms ($Me^+ + e^- \rightarrow M$) in one switch occurs together with the dissolution of metal atoms ($M \rightarrow Me^+ + e^-$) in the other switch. This means that the height increment of one precipitation ($X_A$ or $X_B$) is compensated by a decrement in the other. Eq.(3) represents this condition.
2. The time step $t_s$ consists of the time duration of applying the added voltage $\Delta t$ and the interval time $\Delta t_{int}$, as shown in Figure 2(b). In addition, $\Delta t$ is sufficiently larger than $\Delta t_{int}$ to ignore the displacement of precipitations during $\Delta t_{int}$. This means that once displacements of precipitations take place in $\Delta t$, this status is maintained in the subsequent $\Delta t_{int}$ after the application of $\Delta V_k$.
3. The difference in precipitation height of switch $k$ from the $(j-1)$-th step to the $j$-th step is proportional to $\Delta V_k(j)$. For simplicity, the shape of the precipitated atoms is ignored.
4. $I_k$ is also proportional to $X_0 + X_k(t)$. This assumption implies that there is also a threshold for the displacement of precipitations $Th$, which corresponds to the threshold $\theta$ of the current $I_k$. If $I_k$ is larger than $\theta$, the condition $X_0 + X_k(t) > Th$ is fulfilled. If the $Th$ is set to be smaller than $X_0$, the ASDM dynamics works from the initial state without fluctuations.

According to the foregoing assumptions, the displacement $X_A$ $(= -X_B)$ can be described by the following equations:

$$X_A(t_{j+1}) = Q_A(t_j) - Q_B(t_j) + \delta(t_j), \tag{3}$$

$$Q_k(t_j) = \sum_{j=1}^{N_k} \Delta V_k(j). \tag{4}$$

Here, $Q_k(t_j)$ ($k \in \{A, B\}$) is an “index” for information of past experiences accumulated from the initial time 1 to the current time $t_j$, $N_k$ counts the number of times that machine $k$ has been played, $\Delta V_k$ is the added voltage when playing machine $k$, and $\delta(t_j)$ is an arbitrary fluctuation to which the body is subjected, and $K$ is a parameter. Eqs.(1) and (4) are called the “learning rule.” Consequently, the ASDM evolves according to a particularly simple rule: in addition to the fluctuation, if machine $k$ is played at each time $t$, $R_k - K$ is added to determine $X_k(t_j)$.

The basic behavior of the ASDM underlying TOW dynamics is illustrated in Figure 2(a). Consider that the $X_A$ of switch $A$ is higher than $Th$ while the $X_B$ of switch $B$ is lower than $Th$ at time step $j$. This situation corresponds to $I_A > \theta$ and $I_B < \theta$. Under these circumstances, the ASDM chooses slot machine $A$, and a reward $R_A(j)$ is obtained by playing machine $A$, which is determined as a result of a stochastic event. Then, $V_A = -(V_0 + \Delta V_A(j))$ is applied to switch $A$, whereas switch $B$ is kept $V_B = -V_0$. In the case of $\Delta V_A(j) > 0$, the precipitation of switch $A$ is enhanced by the reduction reaction $Me^+ + e^- \rightarrow M$. On the other hand, the precipitation of switch $B$ is reduced by the oxidation reaction $M \rightarrow Me^+ + e^-$, owing to the conservation law of the total number of atoms and ions. As a result, the height difference between the precipitations of switches $A$ and $B$ increases, as shown in Figure 2(a). Note that $\Delta V_k(j)$ can take both polarities because of the stochastic event. If $\Delta V_A(j)<0$, $V_A$ and $V_B$ are applied to decrease the height difference between the two precipitations. However, as the number of time steps increases, the ASDM finally decides to select one of switches ($A$ or $B$). We illustrate schematically the expected behavior of the ASDM in Figure 2(b). Here, $Th=X_0$ is assumed. Even under this condition, the current $I_k$ (correspondingly $X_k$) fluctuates around $\theta$ ($Th$) with time. At each time step $j$, the ASDM detects the switch showing a higher current than $\theta$ (larger precipitation than $Th$) and selects the corresponding slot machine. The voltages applied to both switches are then updated according to the reward obtained from the slot machines. The ASDM does not always select one machine. Selection of both machines or neither of them is also possible.

## 3. Theoretical Analyses

Theoretical analyses of the TOW dynamics for a Bernoulli type MAB, in which a reward is limited to 0 or 1, are described in [10]. In this section, theoretical analyses of the ASDM are described for a general MAB where a reward is not limited to 0 or 1 and can take an arbitrary value.

### 3.1. MAB Solvability

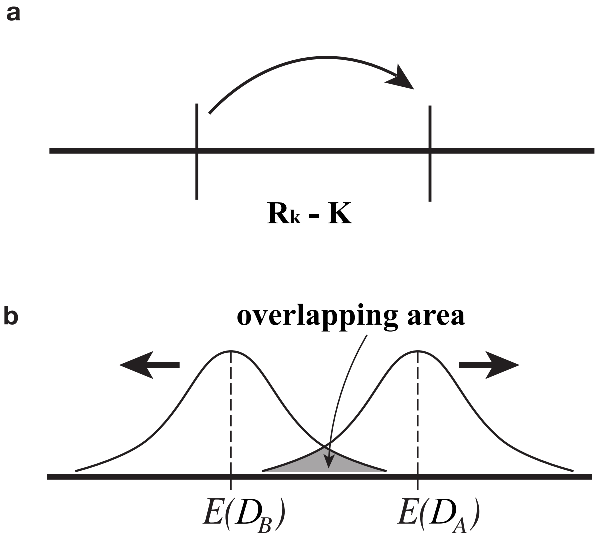

**Figure 3. (a) Random walk: flight $R_k(t) - K$ . Here, the probability density function of $R_k$ has the mean $\mu_k$. (b) Probability distributions of two random walks.**

To explore the MAB solvability of the ASDM using the learning rule $Q_k$ (Eqs.(1) and (4)), let us consider a random-walk model as shown in Fig. 3(a). Here, $R_k(t)$ ($k \in \{A, B\}$) is a reward at time $t$, and $K$ is a parameter (see Eq.(1)). We assume that means of the probability density function of $R_k$ satisfy $\mu_A > \mu_B$ for simplicity. After time step $t$, the displacement $D_k(t)$ ($k \in \{A, B\}$) can be described by

$$D_k(t) = \sum_{j=1}^{N_k(t)} R_k(j) - K\, N_k(t). \tag{5}$$

The expected value of $D_k$ can be obtained from the following equation:

$$E(D_k(t)) = (\mu_k - K)\ N_k(t). \tag{6}$$

In the overlapping area between the two distributions shown in Fig. 3(b), we cannot accurately estimate which is larger. The overlapping area should decrease as $N_k$ increases so as to avoid incorrect judgments. This requirement can be expressed by the following forms:

$$\begin{aligned} \mu_A - K &> 0, \qquad (7) \\ \mu_B - K &< 0. \qquad (8) \end{aligned}$$

These expressions can be rearranged into the form

$$\mu_B < K < \mu_A. \tag{9}$$

In other words, the parameter $K$ must satisfy the above conditions so that the random walk correctly represents the larger judgment.

We can easily confirm that the following form, which we call $K_0$, satisfies the above conditions:

$$\begin{aligned} K_0 &= \frac{\gamma}{2}, \qquad (10) \\ \gamma &= \mu_A + \mu_B. \qquad (11) \end{aligned}$$

Therefore, we can conclude that the ASDM using the learning rule $Q_k$ with the parameter $K_0$ can solve the MAB correctly.

### 3.2. *Origin of the high performance*

In many popular algorithms such as the $\epsilon$-GREEDY algorithm [24], at each time $t$, an estimate of reward probability is updated for either of the two machines being played. On the other hand, in an imaginary circumstance in which the sum of the mean rewards $\gamma = \mu_A + \mu_B$ is known to the player, we can update both of the two estimates simultaneously, even though only one of the machines was played.

**Table 1. Estimates for each mean reward based on the knowledge that machine $A$ was played $N_A$ times and that machine $B$ was played $N_B$ times—on the assumption that the sum of the mean rewards $\gamma = \mu_A + \mu_B$ is known.**

| | | | |
|---|---|---|---|
| $A$: | $\frac{\sum_{j=1}^{N_A} R_A(j)}{N_A}$ | $B$: | $\gamma - \frac{\sum_{j=1}^{N_A} R_A(j)}{N_A}$ |
| $A$: | $\gamma - \frac{\sum_{j=1}^{N_B} R_B(j)}{N_B}$ | $B$: | $\frac{\sum_{j=1}^{N_B} R_B(j)}{N_B}$ |

The top and bottom rows of Table 1 provide estimates based on the knowledge that machine $A$ was played $N_A$ times and that machine $B$ was played $N_B$ times, respectively. Note that we can also update the estimate of the machine that was not played, owing to the given $\gamma$.

From the above estimates, each expected reward $Q'_k$ ($k \in \{A, B\}$) is given as follows:

$$\begin{aligned} Q'_A &= N_A \frac{\sum_{j=1}^{N_A} R_A(j)}{N_A} + N_B\,(\gamma - \frac{\sum_{j=1}^{N_B} R_B(j)}{N_B}) \\ &= \sum_{j=1}^{N_A} R_A(j) - \sum_{j=1}^{N_B} R_B(j) + \gamma N_B, \qquad (12) \\ Q'_B &= N_A\,(\gamma - \frac{\sum_{j=1}^{N_A} R_A(j)}{N_A}) + N_B \frac{\sum_{j=1}^{N_B} R_B(j)}{N_B} \\ &= \sum_{j=1}^{N_B} R_B(j) - \sum_{j=1}^{N_A} R_A(j) + \gamma\, N_A. \qquad (13) \end{aligned}$$

These expected rewards, $Q'_j$s, are not the same as those given by the learning rules of TOW dynamics, $Q_j$s in Eqs.(1) and (4). However, what we use substantially in TOW dynamics is the difference

$$Q_A - Q_B = (\sum_{j=1}^{N_A} R_A(j) - \sum_{j=1}^{N_B} R_B(j)) - K\,(N_A - N_B). \qquad (14)$$

When we transform the expected rewards $Q'_j$s into $Q''_j = Q'_j/2$, we can obtain the difference

$$Q''_A - Q''_B = (\sum_{j=1}^{N_A} R_A(j) - \sum_{j=1}^{N_B} R_B(j)) - \frac{\gamma}{2}\,(N_A - N_B). \qquad (15)$$

Comparing the coefficients of Eqs.(14) and (15), the two differences are always equal when $K = K_0$ (Eq.(10)) is satisfied. Eventually, we can obtain the nearly optimal weighting parameter $K_0$ in terms of $\gamma$.

This derivation implies that the learning rule for the ASDM is equivalent to that of the imaginary system in which both of the two estimates can be updated simultaneously. In other words, the ASDM imitates the imaginary system that determines its next move at time $t + 1$ in referring to the estimates of the two machines, even if one of them was not actually played at time $t$. This unique feature in the learning rule, derived from the fact that the sum of mean rewards is given in advance, may be one of the origins of the high performance of the ASDM.

Monte Carlo simulations were performed it was verified that the ASDM with $K_0$ exhibits an exceptionally high performance, which is comparable to its peak performance—achieved with the optimal parameter $K_{opt}$. To derive the optimal value $K_{opt}$ accurately, we need to take into account the fluctuations.

In addition, the essence of the process described here can be generalized to $M$-machine cases. To separate distributions of the top $m$-th and top $(m + 1)$-th machine, as shown in Fig. 3(b), all we need is the following $K_0$:

$$\begin{aligned} K_0 &= \frac{\gamma'}{2}, \qquad (16) \\ \gamma' &= \mu_{(m)} + \mu_{(m+1)}. \qquad (17) \end{aligned}$$

Here, $\mu_{(m)}$ denotes the top $m$-th mean, and $m$ is any integer from 1 to $M-1$. The MBP is a special case where $m = 1$. In fact, for $M$-machine and $X$-player cases, we have designed a physical system that can determine the overall optimal state, called the "social maximum [31, 32]," quickly and accurately [33, 34].

### 3.3. Performance characteristics

In this section, we calculate a performance measure, called the "regret," to characterize the high performance of the ASDM. We consider the "cheater algorithm," an imaginary model for solving the MAB, because the regret of this algorithm can be easily calculated.

The cheater algorithm selects a machine to play according to the following estimate $S_k$ ($k \in \{A, B\}$)

$$S_A = X_{A,1} + X_{A,2} + \cdots + X_{A,N}, \quad (18)$$

$$S_B = X_{B,1} + X_{B,2} + \cdots + X_{B,N}. \quad (19)$$

Here, $X_{k,i}$ is a random variable. If $S_A > S_B$ at time $t = N$, machine $A$ is played at time $t = N + 1$. If $S_B > S_A$ at time $t = N$, machine $B$ is played at time $t = N + 1$. If $S_A = S_B$ at time $t = N$, a machine is played randomly at time $t = N + 1$. Note that the algorithm refers to results of both machines at time $t$ without any attention to which machine was played at time $t - 1$. In other words, the algorithm "cheats" because it plays both machines and collects both results, but declares that it plays only one machine at a time.

The expected value and the variance of $X_k$ are defined as $E(X_k) = \mu_k$ and $V(X_k) = \sigma_k^2$. Here, $\mu_k$ is the same as the $P_k$ defined earlier. From the central-limit theorem, $S_k$ has a Gaussian distribution with $E(S_k) = \mu_k N$ and $V(S_k) = \sigma_k^2 N$. If we define a new variable $S = S_A - S_B$, $S$ has a Gaussian distribution and carries the following values:

$$E(S) = (\mu_A + \mu_B)N, \quad (20)$$

$$V(S) = (\sigma_A^2 + \sigma_B^2)N, \quad (21)$$

$$\sigma(S) = \sqrt{\sigma_A^2 + \sigma_B^2}\,\sqrt{N}. \quad (22)$$

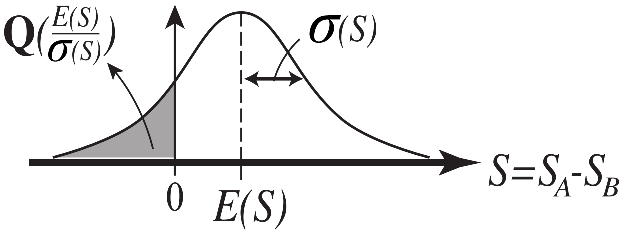

**Figure 4. $\mathbf{Q}(\frac{E(S)}{\sigma(S)})$: probability of selecting the lower-reward machine in the cheater algorithm**

From Fig. 4, the probability of playing machine $B$, which has a lower reward probability, can be described as $\mathbf{Q}(\frac{E(S)}{\sigma(S)})$. Here, $\mathbf{Q}(x)$ is a $\mathbf{Q}$-function. We obtain

$$P(t = N + 1, B) = \mathbf{Q}(\phi\sqrt{N}). \quad (23)$$

Here,

$$\phi = \frac{\mu_A - \mu_B}{\sqrt{\sigma_A^2 + \sigma_B^2}}. \tag{24}$$

Using the Chernoff bound $\mathbf{Q}(x) \leq \frac{1}{2}\exp(-\frac{x^2}{2})$, we can calculate the upper bound of a measure, called the "regret," which quantifies the accumulated losses of the algorithm.

$$regret = (\mu_A - \mu_B)E(N_B). \tag{25}$$

$$\begin{aligned}
E(N_B) &= \sum_{t=0}^{N-1} \mathbf{Q}(\phi\sqrt{t}) \\
&\leq \sum_{t=0}^{N-1} \frac{1}{2}\exp(-\frac{\phi^2}{2}t) \\
&= \frac{1}{2} + \sum_{t=1}^{N-1} \frac{1}{2}\exp(-\frac{\phi^2}{2}t) \\
&\leq \frac{1}{2} + \int_0^{N-1} \frac{1}{2}\exp(-\frac{\phi^2}{2}t)dt \\
&= \frac{1}{2} - \frac{1}{\phi^2}\left(\exp(-\frac{\phi^2}{2}(N-1)) - 1\right) \qquad (26) \\
&\to \frac{1}{2} + \frac{1}{\phi^2}. \qquad (27)
\end{aligned}$$

Note that the regret becomes constant as $N$ increases.

Using the "cheated" results, we can also calculate the regret for the ASDM in the same way. In this case,

$$S_A = X_{A,1} + X_{A,2} + \cdots + X_{A,N_A} - KN_A, \tag{28}$$

$$S_B = X_{B,1} + X_{B,2} + \cdots + X_{B,N_B} - KN_B. \tag{29}$$

$X_{k,i}$ is also a random variable. Then, we obtain

$$E(S_k) = (\mu_k - K)N_k, \tag{30}$$

$$V(S_k) = \sigma_k^2 N_k. \tag{31}$$

Using the new variables $S = S_A - S_B$, $N = N_A + N_N$, and $D = N_A - N_N$, we also obtain

$$E(S) = \frac{\mu_A - \mu_B}{2}N + \left(\frac{\mu_A + \mu_B}{2} - K\right)D, \tag{32}$$

$$V(S) = \frac{\sigma_A^2 + \sigma_B^2}{2}N + \frac{\sigma_A^2 - \sigma_B^2}{2}D. \tag{33}$$

If the conditions $K = K_0$ and $\sigma_A = \sigma_B \equiv \sigma$ are satisfied, we then obtain

$$E(S) = \frac{\mu_A - \mu_B}{2}N, \tag{34}$$

$$V(S) = \sigma^2 N, \tag{35}$$

and

$$P(t = N+1, B) = \mathbf{Q}(\phi_T \sqrt{N}). \tag{36}$$

Here,

$$\phi_T = \frac{\mu_A - \mu_B}{2\sigma}. \tag{37}$$

We can then calculate the upper bound of the regret for the ASDM

$$\begin{aligned} E(N_B) &= \sum_{t=0}^{N-1} \mathbf{Q}(\phi_T \sqrt{t}) \\ &\leq \frac{1}{2} - \frac{1}{\phi_T^2}\left(\exp(-\frac{\phi_T^2}{2}(N-1)) - 1\right) \quad (38) \\ &\rightarrow \frac{1}{2} + \frac{1}{\phi_T^2}. \quad (39) \end{aligned}$$

Note that the regret for the ASDM also becomes constant as $N$ increases.

It is well known that optimal algorithms for the MAB, defined by Auer et al. [16], have a regret proportional to $\log(N)$. The regret for the optimal algorithms has no finite upper bound as $N$ increases because it continues to require playing the lower-reward machine to ensure that the probability of incorrect judgment goes to zero. A constant regret for the ASDM means that the probability of incorrect judgment remains non-zero, although this probability is nearly equal to zero. However, it would appear that the reward probabilities change frequently in actual decision-making situations, and their long-term behavior is not important for many practical purposes. For this reason, the ASDM would be more suited to real-world applications.

## 4. Simulation Results

In this section, to show the effectiveness of the ASDM, we investigate the performance comparison between the ASDM and the SOFTMAX algorithm which is a well-known algorithm for efficient decision-making [24] (see Appendix). From computer simulations, we confirmed that, in almost all cases, an ASDM with the parameter $K_0$ $(=\frac{\mu_A+\mu_B}{2})$ can acquire more rewards than a SOFTMAX algorithm with the optimized parameter $\tau_{opt}$, although SOFTMAX is well known as a good algorithm [35]. Here, the parameter $K_0$ is nearly optimal as shown in Fig. 5(a). Regret, as defined in the previous section, is a performance measure where a lower value indicates higher performance (more rewards). Figure 5(b) shows an ASDM/SOFTMAX performance comparison. The vertical axis denotes the regret (mean values of 1000 samples), and the horizontal axis denotes the number of plays. The blue dotted line denotes the upper bound of the ASDM with $K_0$ (Eq.(39)). For the reward PDFs, we used normal distributions $N(\mu_A, \sigma^2)$ and $N(\mu_B, \sigma^2)$, where $\mu_A$=0.6, $\mu_B$=0.5, and $\sigma$=0.2. Computer simulations were executed under the condition that $Th$=$X_0$ and $\delta$=$sin(\pi/2 + \pi t)$.

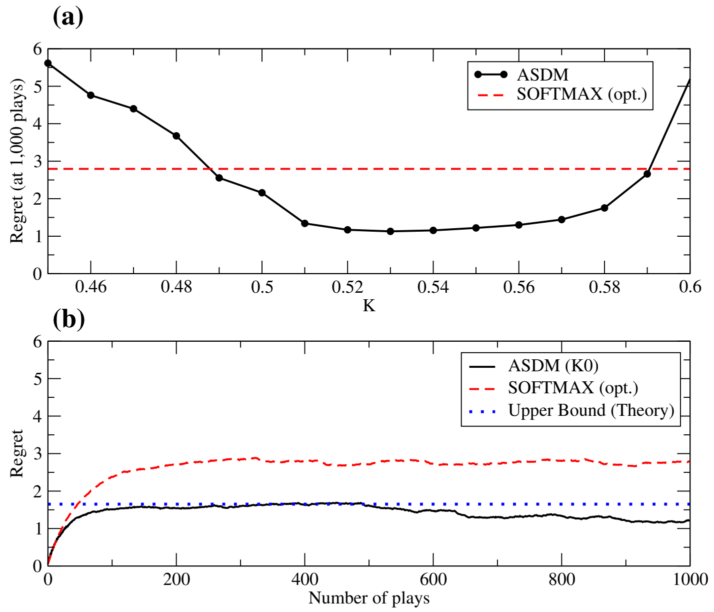

**Figure 5. (a) The regret of the ASDM with $K$ until $1,000$ plays (black solid line). The red dashed line denotes the regret of SOFTMAX with $\tau_{opt}$=0.3 (b) Performance comparison between the ASDM (black solid line) and SOFTMAX [24] (red dashed line). We used $K_0$=0.55 for the ASDM and $\tau_{opt}$=0.3 for SOFTMAX. The blue dotted line denotes the upper bound of the ASDM with $K_0$ (Eq.(39)).**

## 5. Discussion and Conclusion

In this study, we proposed an ASDM for solving the MAB, and analytically validated their high efficiency in making decisions. In conventional decision-making algorithms for solving MABs, the parameter for adjusting the "exploration time" must be optimized. This exploration parameter always reflects the difference between the rewarded experiences, i.e., $|\mu_A - \mu_B|$. In contrast, the ASDM demonstrates that higher performance can be achieved by introducing a parameter $K_0$ that refers to the sum of the rewarded experiences, i.e., $\mu_A + \mu_B$. This type of optimization, using the sum of the rewarded experiences, is particularly useful for time varying environments (reward probability or reward PDF) [26]. Owing to this novelty, the high performance of the TOW dynamics can be reproduced when implementing these dynamics with atomic switches.

The ASDM proposed in this paper is a simple "ideal model." While the assumptions used for constructing the model may contain some points that do not match real experimental situations, we can more accurately extend the model so that the modified assumptions do match real experimental situations. As we extend the model to treat more than two options, it may be found that there are some experimental limitations in implementing TOW dynamics when more than two atomic switches are used. As long as the TOW dynamics between atomic switches is implemented, high performance decision-making can be guaranteed even in the extended model.

The ASDM will introduce a new physics-based analog-computing paradigm, which will include such things as "intelligent nanodevices" based on self-judgment. Thus, our proposed physics-based analog-computing paradigm would be useful for a variety of real-world applications and for understanding the biological information-processing principles that exploit their underlying physics.

## Appendix

### *SOFTMAX algorithm*

The SOFTMAX algorithm is a well-known algorithm for solving MABs [24]. In this algorithm, the probability of selecting A or B, $P'_A(t)$ or $P'_B(t)$, is given by the following Boltzmann distributions:

$$P'_A(t) = \frac{\exp[\beta \cdot Q_A(t)]}{\exp[\beta \cdot Q_A(t)] + \exp[\beta \cdot Q_B(t)]}, \tag{40}$$

$$P'_B(t) = \frac{\exp[\beta \cdot Q_B(t)]}{\exp[\beta \cdot Q_A(t)] + \exp[\beta \cdot Q_B(t)]}, \tag{41}$$

where $Q_k(t)$ ($k \in \{A, B\}$) is given by $\frac{\sum_{j=1}^{N_k(t)} R_k(j)}{N_k(t)}$. Here, $\beta$ is a time-dependent form in our study, as follows:

$$\beta(t) = \tau \cdot t. \tag{42}$$

$\beta = 0$ corresponds to a random selection, and $\beta \to \infty$ corresponds to a greedy action.

## Acknowledgments

The authors thank Dr. Etsushi Nameda and Prof. Masahiko Hara for valuable discussions in the early stages of this work. This work was supported in part by the Sekisui Chemical Grant Program for "Research on Manufacturing Based on Innovations Inspired by Nature."

## Conflict of Interest

The authors declare that there is no conflicting of interest regarding the publication of this paper.

## References

1. Castro LN (2007) Fundamentals of natural computing: an overview. *Physics of Life Reviews* 4: 1-36.
2. Kari L, Rozenberg G (2008) The many facets of natural computing. *Communications of the ACM* 51: 72-83.
3. Rozenberg G, Back T, Kok J (2012) *Handbook of natural computing*, Springer-Verlag.
4. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization in simulated annealing. *Science* 220: 671-680.
5. Hopfield JJ, Tank DW (1985) Neural computation of decisions in optimization problems. *Biological Cybernetics* 52: 141-152.
6. Brady RM (1985) Optimization strategies gleaned from biological evolution. *Nature* 317: 804-806.
7. Adelman LM (1994) Molecular computation of solutions to combinatorial problems. *Science* 266: 1021-1024.

8. Terabe K, Hasegawa T, Nakayama T, et al. (2001) Quantum point contact switch realized by solid electrochemical reaction. *RIKEN Review* 37: 7-8.

9. Terabe K, Hasegawa T, Nakayama T, et al. (2005) Quantized conductance atomic switch. *Nature* 433: 47-50.

10. Kim S-J, Aono M, Nameda E (2015) Efficient decision-making by volume-conserving physical object. *New J Phys* 17: 083023.

11. Robbins H (1952) Some aspects of the sequential design of experiments. *Bull Amer Math Soc* 58: 527-536.

12. Thompson W (1933) On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25: 285-294.

13. Gittins J, Jones D (1974) Dynamic allocation index for the sequential design of experiments, In: Gans, J. *Progress in Statistics* North Holland, 241-266.

14. Gittins J (1979) Bandit processes and dynamic allocation indices. *J R Stat Soc B* 41: 148-177.

15. Agrawal R (1995) Sample mean based index policies with O(log n) regret for the multi-armed bandit problem. *Adv Appl Prob* 27: 1054-1078.

16. Auer P, Cesa-Bianchi N, Fischer P (2002) Finite-time analysis of the multiarmed bandit problem. *Machine Learning* 47: 235-256.

17. Lai L, Jiang H, Poor HV (2008) Medium access in cognitive radio networks: a competitive multi-armed bandit framework. *Proc. of IEEE 42nd Asilomar Conference on Signals, System and Computers*, 98-102.

18. Lai L, Gamal HE, Jiang H, et al. (2011) Cognitive medium access: exploration, exploitation, and competition. *IEEE Trans. on Mobile Computing* 10: 239-253.

19. Agarwal D, Chen BC, Elango P (2009) Explore/exploit schemes for web content optimization. *Proc of ICDM2009*, `http://dx.doi.org/10.1109/ICDM.2009.52`.

20. Kocsis L, Szepesvári C. (2006) Bandit based monte-carlo planning, In: Carbonell, J. G. et al., *17th European Conference on Machine Learning, Lecture Notes in Artificial Intelligence* 4212, Springer, 282-293.

21. Gelly S, Wang Y, Munos R, et al. (2006) Modification of UCT with patterns in Monte-Carlo Go. *RR-6062-INRIA*, 1-19.

22. Demis EC, Aguilera R, Sillin HO, et al. (2015) Atomic switch networks - nanoarchitectonic design of a complex system for natural computing. *Nanotechnology* 26: 204003.

23. Avizienis AV, Sillin HO, Martin-Olmos C, et al. (2012) Neuromorphic atomic switch networks. *PLoS ONE* 7: e42772.

24. Sutton R, Barto A (1998) *Reinforcement Learning: An Introduction*, MIT Press.

25. Kim S-J, Aono M, Hara M (2010) Tug-of-war model for multi-armed bandit problem, In: Calude C. et al. *Unconventional Computation, Lecture Notes in Computer Science* 6079, Springer, 69-80.

26. Kim S-J, Aono M, Hara M (2010) Tug-of-war model for the two-bandit problem: Nonlocally-correlated parallel exploration via resource conservation. *BioSystems* 101: 29-36.

27. Kim S-J, Naruse M, Aono M, et al. (2013) Decision maker based on nanoscale photo-excitation transfer. *Sci Rep* 3: 2370.

28. Naruse M, Nomura W, Aono M, et al. (2014) Decision making based on optical excitation transfer via near-field interactions between quantum dots. *J Appl Phys* 116: 154303.

29. Naruse M, Berthel M, Drezet A, et al. (2015) Single photon decision maker *Sci Rep* 5: 13253.

30. Tsuruoka T, Hasegawa T, Terabe K, et al. (2012) Conductance quantization and synaptic behavior in a $Ta_2O_5$-based atomic switch. *Nanotechnology* 23: 435705.

31. Roughgarden T (2005) *Selfish routing and the price of anarchy*, MIT Press, Cambridge.

32. Nisan N, Roughgarden T, Tardos E, et al. (2007) *Algorithmic Game Theory*, Cambridge Univ. Press.

33. Kim S-J, Aono M (2015) Decision maker using coupled incompressible-fluid cylinders. Special issue of *Advances in Science, Technology and Environmentology* B11: 41-45, Available from: http://arxiv.org/abs/1502.03890.

34. Kim S-J, Naruse M, Aono M (2015) Harnessing natural fluctuations: analogue computer for efficient socially-maximal decision-making. *eprint arXiv*, Available from: http://arxiv.org/abs/1504.03451.

35. Vermorel J, Mohri M (2005) Multi-armed bandit algorithms and empirical evaluation, In: Gama J., et al. *16th European Conference on Machine Learning. Lecture Notes in Artificial Intelligence* 3720, Springer, 437-448.