PAPER

Efficient decision-making by volume-conserving physical object

Song-Ju Kim1, Masashi Aono2 and Etsushi Nameda3

1 WPI Center for Materials Nanoarchitectonics, National Institute for Materials Science, Tsukuba, Ibaraki 305-0044, Japan
2 Earth-Life Science Institute, Tokyo Institute of Technology, Tokyo 152-8550, and PRESTO JST, Japan
3 RIKEN, Wako, Saitama 351-0198, Japan

E-mail: KIM.Songju@nims.go.jp

Keywords: random walk, natural computing, decision-making

Abstract

Decision-making is one of the most important intellectual abilities of not only humans but also other biological organisms, helping their survival. This ability, however, may not be limited to biological systems and may be exhibited by physical systems. Here we demonstrate that any physical object, as long as its volume is conserved when coupled with suitable operations, provides a sophisticated decision-making capability. We consider the multi-armed bandit problem (MBP), the problem of finding, as accurately and quickly as possible, the most profitable option from a set of options that gives stochastic rewards. Efficient MBP solvers are useful for many practical applications, because MBP abstracts a variety of decision-making problems in real-world situations in which an efficient trial-and-error is required. These decisions are made as dictated by a physical object, which is moved in a manner similar to the fluctuations of a rigid body in a tug-of-war (TOW) game. This method, called 'TOW dynamics', exhibits higher efficiency than conventional reinforcement learning algorithms. We show analytical calculations that validate statistical reasons for TOW dynamics to produce the high performance despite its simplicity. These results imply that various physical systems in which some conservation law holds can be used to implement an efficient 'decision-making object'. The proposed scheme will provide a new perspective to open up a physics-based analog computing paradigm and to understanding the biological information-processing principles that exploit their underlying physics.

1. Introduction

The computing principles in modern digital paradigms have been designed to be dissociated from the underlying physics of natural phenomena [1]. In the construction of CMOS devices, wide-band-gap materials have been employed so that physical fluctuations such as thermal noise, which often violate logically-valid behavior, could be neglected [2]. Since electron dynamics constrained by physical laws cannot be controlled when only parameters of the same degree of freedom as those of logical input–output responses are modulated, considerably complicated circuits are required for implementing relatively simple logic gates such as NAND and NOR [3]. However, these efforts to circumvent the division between physics and computation are costly in terms of energy consumption and manufacturing resources. On the other hand, when we look at the natural world, information processing in biological systems is elegantly coupled with their underlying physical laws and fluctuations [4–6]. This suggests a potential for establishing a new physics-based analog-computing paradigm. In this paper, we show that a physical constraint, the conservation law for the volume of a rigid body, allows for efficient solving of decision-making problems when subjected to suitable operations involving fluctuations.

Suppose there are M slot machines, each of which returns a reward; for example, a coin, with a certain probability that is unknown to a player. Let us consider a minimal case: two machines A and B give rewards with individual probabilities $P_A$ and $P_B$, 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 multi-armed bandit
problem (MBP) is used to determine the optimal strategy for finding the machine with the highest reward probability as accurately and quickly as possible by referring to past experiences. The MBP is formulated as a mathematical problem without loss of generality and so is related to various stochastic phenomena. In fact, many application problems in diverse fields, such as communications (cognitive networks [7, 8]), commerce (advertising on the web [9]), entertainment (Monte Carlo tree search, which is used for computer games [10, 11]), can be reduced to MBPs. Particularly, the ‘upper confidence bound 1 (UCB1) algorithm’ for solving MBPs is used worldwide in many practical applications [17].

In the context of reinforcement learning, the MBP was originally described by Robbins [12], though the essence of the problem had been studied earlier by Thompson [13]. 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 [14, 15]. 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 [16, 17].

2. Methods

2.1. Tug-of-war (TOW) dynamics

Kim et al proposed an MBP solution using a dynamical system, called ‘TOW dynamics’; this algorithm was inspired by the spatiotemporal dynamics of a single-celled amoeboid organism (the true slime mold P. polycephalum) [18–23], which maintains a constant intracellular-resource volume while collecting environmental information by concurrently expanding and shrinking its pseudopod-like terminal parts. In this nature-inspired algorithm, the decision-making function is derived from its underlying physics, resembling that of a TOW game. The physical constraint in TOW dynamics, the conservation law for the volume of the amoeboid body, entails a non-local correlation among the terminal parts, that is, the volume increment in one part is immediately compensated by volume decrement(s) in the other part(s). In our previous studies [18–23], we showed that, owing to the non-local correlation derived from the volume-conservation law, TOW dynamics exhibit higher performance than other well-known algorithms such as the modified $\epsilon$-greedy algorithm and the modified softmax algorithm, which is comparable to the UCB1-tuned algorithm (seen as the best choice among parameter-free algorithms [17]). These observations suggest that efficient decision-making devices could be implemented using any physical object as long as it held some common physical attributes such as the conservation law. In fact, Kim et al demonstrated that optical energy-transfer dynamics between quantum dots, in which energy is conserved, can be exploited for the implementation of TOW dynamics [24, 25].

In this paper, we extract the most essential ingredients of the TOW dynamics and formulated its simplified version, called the ‘TOW principle’, in order to summarize the mechanism of high performance by showing analytical calculations of ‘regret’ that can characterize the performance of the multi-armed bandit algorithms. Consider a volume-conserving physical object; for example, a rigid body like an iron bar (the slot-machine’s handle), as shown in figure 1. Here, the variable $X_k$ represents the displacement of terminal $k$ from an initial position, where $k \in \{A, B\}$. If $X_k$ is a maximum, we assume that the body makes a decision to play machine $k$. In TOW dynamics, the MBP is represented in its inverse form: instead of ‘rewarding’ the player when the machine produces a coin with a probability $P_k$, we ‘punish’ the player when the machine gives no coin with a probability $P_{\bar{k}}$.

Figure 1. TOW dynamics. If machine $k$ ($k \in \{A, B\}$) is played at each time $t$, $+1$ and $-\omega$ are added to $X_k(t)$ for rewarding (a) and non-rewarding cases (b), respectively.
\[ X_A(t + 1) = Q_A(t) - Q_B(t) + \delta(t), \]
\[ Q_k(t) = N_k(t) - (1 + \omega)L_k(t). \]

Here, \( Q_k(t) \) is an estimate of information on past experiences accumulated from the initial time 1 to current time \( t \), \( N_k(t) \) counts the number of times that machine \( k \) has been played until time \( t \), \( L_k(t) \) counts the number of punishments when playing machine \( k \) until time \( t \), \( \delta(t) \) is an arbitrary fluctuation to which the body is subjected, and \( \omega \) is a weighting parameter to be described in detail later on in this paper. Equation (2) called the ‘learning rule’. Consequently the TOW dynamics evolve according to a particularly simple rule: in addition to the fluctuation, if machine \( k \) is played at each time \( t \), \( +1 \) and \( -\omega \) are added to \( X_k(t) \) when rewarded and non-rewarded, respectively (Figure 1).

3. Results

3.1. Solvability of the MBP

To explore the MBP solvability of TOW dynamics, let us consider a random-walk model. As shown in Figure 2(a), \( \alpha \) (right flight when rewarded) and \( \beta \) (left flight when non-rewarded) are the parameters. We assume that \( P_A > P_B \) for simplicity. After time step \( t \), the displacement \( R_k(t) \) can be described by
\[ R_k(t) = \alpha \left( N_k - L_k \right) - \beta L_k \]
\[ = \alpha N_k - (\alpha + \beta) L_k. \]

The expected value of \( R_k \) can be obtained from the following equation:
\[ E \left( R_k(t) \right) = \left\{ \alpha P_k - \beta \left(1 - P_k\right) \right\} N_k. \]

In the overlapping area between the two distributions shown in Figure 2(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:
\[ \alpha P_A - \beta \left(1 - P_A\right) > 0, \]
\[ \alpha P_B - \beta \left(1 - P_B\right) < 0. \]

These expressions can be rearranged into the form
\[ P_B < \frac{\beta}{\alpha + \beta} < P_A. \]

In other words, the parameters \( \alpha \) and \( \beta \) must satisfy the above conditions so that the random walk correctly represents the larger judgment.
We can easily confirm that the following form satisfies the above conditions:

\[ \frac{\beta}{\alpha + \beta} = \frac{P_A + P_B}{2} \]  

(8)

From \( R(t)/\alpha = Q_k(t) \), we obtain \( \omega = \frac{\beta}{\alpha} \). From this and equation (8), we obtain

\[ \omega_0 = \frac{\gamma}{2 - \gamma} \]  

(9)

\[ \gamma = P_A + P_B \]  

(10)

Here, we have set the parameter \( \omega \) to \( \omega_0 \). Therefore, we can conclude that the algorithm using the learning rule \( Q_k \) with the parameter \( \omega_0 \) can solve the MBP correctly.

### 3.2. TOW principle: origin of the high performance

In many popular algorithms such as the \( \epsilon \)-greedy algorithm, 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 reward probabilities \( \gamma = P_A + P_B \) is known to the player, we can update both of the two estimates simultaneously, even though only one of the machines was played.

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 \( Q'_j \) \( (k \in \{A, B\}) \) is given as follows:

\[
Q'_A = N_A \left( \frac{N_A - L_A}{N_A} + N_B \frac{\gamma - N_B - L_B}{N_B} \right) = N_A - L_A + (\gamma - 1)N_B + L_A, 
\]

\[
Q'_B = N_A \left( \frac{\gamma - N_A - L_A}{N_A} + N_B \frac{N_B - L_B}{N_B} \right) = N_B - L_B + (\gamma - 1)N_A + L_A. 
\]

These expected rewards, \( Q'_j \)'s, are not the same as those given by the learning rules of TOW dynamics, \( Q_j \) in equation (2). However, what we use substantially in TOW dynamics is the difference

\[
Q_A - Q_B = \left( N_A - N_B \right) - (1 + \omega) \left( L_A - L_B \right). 
\]

(13)

When we transform the expected rewards \( Q'_j \)'s into \( Q''_j = Q'_j/(2 - \gamma) \), we can obtain the difference

\[
Q''_A - Q''_B = \left( N_A - N_B \right) - \frac{2}{2 - \gamma} \left( L_A - L_B \right). 
\]

(14)

Comparing the coefficients of equations (13) and (14), the differences in their constituent terms are always equal when \( \omega = \omega_0 \) (equation (9)) is satisfied. Eventually, we can obtain the nearly optimal weighting parameter \( \omega_0 \) in terms of \( \gamma \).

This derivation implies that the learning rule for TOW dynamics is equivalent to that of the imaginary system in which both of the two estimates can be updated simultaneously. In other words, TOW dynamics 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 reward probabilities is given in advance, may be one of the origins of the high performance of TOW dynamics.

Monte Carlo simulations were performed it was verified that TOW dynamics with \( \omega_0 \) exhibits an exceptionally high performance, which is comparable to its peak performance—achieved with the optimal
parameter $\omega_{\text{opt}}$. To derive the optimal value $\omega_{\text{opt}}$ accurately, we need to take into account the fluctuation and other dynamics of terminals [21].

In addition, the essence of the process described here can be generalized to $K$-machine cases. To separate distributions of the top $M$th and top $(M + 1)$th machine, as shown in figure 2(b), all we need is the following:

$$
\omega_0 = \frac{\gamma'}{2 - \gamma'},
$$

(15)

$$
\gamma' = P_{(M)} + P_{(M+1)}.
$$

(16)

Here, $P_{(M)}$ denotes the top $M$th reward probability, and $M$ is any integer from 1 to $K - 1$. The MBP is a special case where $M = 1$. In fact, for $K$-machine and $M$-player cases, we have designed a physical system that can determine the overall optimal state, called the ‘social maximum’, quickly and accurately [26].

### 3.3. Performance characteristics

To characterize the high performance of TOW dynamics, let us consider the imaginary model for solving the MBP, called the ‘cheater algorithm’. The cheater algorithm selects a machine to play according to the following estimate:

$$
S_A = X_{A,1} + X_{A,2} + \cdots + X_{A,N},
$$

(17)

$$
S_B = X_{B,1} + X_{B,2} + \cdots + X_{B,N}.
$$

(18)

Here, $X_{k,i}$ is a random variable that takes either 1 (rewarded) or 0 (non-rewarded). 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 $t = N + 1$, 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 $X_A$ are defined as $E(X_A) = \mu_A$ and $V(X_A) = \sigma_A^2$. Here, $\mu_A$ is the same as the $P_i$ defined earlier. From the central-limit theorem, $S_A$ has a Gaussian distribution with $E(S_A) = \mu_A N$ and $V(S_A) = \sigma_A^2 N$. If we define a new variable $S = S_A - S_B$, $S$ has a Gaussian distribution and carries the following values:

$$
E(S) = \left(\mu_A + \mu_B\right)N,
$$

(19)

$$
V(S) = \left(\sigma_A^2 + \sigma_B^2\right)N,
$$

(20)

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

(21)

From figure 3, the probability of playing machine $B$, which has a lower reward probability, can be described as $Q\left(\frac{E(S)}{\sigma(S)}\right)$. Here, $Q(x)$ is a Q-function. We obtain

$$
P(t = N + 1, B) = Q\left(\phi \sqrt{N}\right).
$$

(22)

Here,

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

(23)

Using the Chernoff bound $Q(x) \leq \frac{1}{2} \exp\left(-\frac{x^2}{2}\right)$, we can calculate the upper bound of a measure, called the ‘regret’, which quantifies the accumulated losses of the cheater algorithm.
Note that the regret becomes constant as \( N \) increases.

Using the ‘cheated’ results, we can also calculate the regret of TOW dynamics in the same way. In this case,

\[
S_A = X_{A,1} + X_{A,2} + \cdots + X_{A,N_A} - \omega L_A, \\
S_B = X_{B,1} + X_{B,2} + \cdots + X_{B,N_B} - \omega L_B. 
\]

\( X_{k,i} \) is also a random variable that takes either 1 (rewarded) or 0 (non-rewarded). Here, we use \( L_A = (1 - \mu_k) N_k \). Then, we obtain \( E(S_k) = [\mu_k - (1 - \mu_k) \omega] N_k \) and \( V(S_k) = \sigma_k^2 N_k \).

Using the new variables \( S = S_A - S_B, N = N_A + N_B \), and \( D = N_A - N_B \), we also obtain

\[
E(S) = \frac{\mu_A - \mu_B}{2} (1 + \omega) N + \left\{ \frac{\mu_A + \mu_B}{2} (1 + \omega) - \omega \right\} D, \\
V(S) = \frac{\sigma_A^2 + \sigma_B^2}{2} N + \frac{\sigma_A^2 - \sigma_B^2}{2} D. 
\]

If the conditions \( \omega = \omega_0 \) and \( \sigma_A = \sigma_B \equiv \sigma \) are satisfied, we then obtain

\[
E(S) = \frac{\mu_A - \mu_B}{2} (1 + \omega_0) N, \\
V(S) = \sigma^2 N, 
\]

and

\[
P(t = N + 1, B) = Q(\phi_T \sqrt{N}). 
\]

Here,

\[
\phi_T = \frac{(\mu_A - \mu_B)(1 + \omega_0)}{2 \sigma}. 
\]

We can then calculate the upper bound of the regret for TOW dynamics

\[
E(N_B) = \sum_{t=0}^{N-1} Q(\phi_T \sqrt{t}) 
\leq \frac{1}{2} - \frac{1}{\phi_T^2} \left( \exp \left\{ -\frac{\phi_T^2}{2} (N - 1) \right\} - 1 \right) 
\]

\[
= \frac{1}{2} + \frac{1}{\phi_T^2}. 
\]

Note that the regret for TOW dynamics also becomes constant as \( N \) increases.

It is known that optimal algorithms for the MBP, defined by Auer et al, have a regret proportional to \( \log(N) \) [17]. The regret 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 means that the probability of incorrect judgment remains non-zero in TOW dynamics, 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 crucial for many practical purposes. For this reason, TOW dynamics would be more suited to real-world applications.
4. Conclusion and discussion

In this paper, we proposed TOW dynamics for solving the MBP and analytically validated that their high efficiency in making a series of decisions for maximizing the total sum of stochastically obtained rewards is embedded in any volume-conserving physical object when subjected to suitable operations involving fluctuations. In conventional decision-making algorithms for solving the MBP, the parameter for adjusting the ‘exploration time’ must be optimized. This exploration parameter often reflects the difference between the rewarded experiences, i.e., $|P_A - P_B|$. In contrast, TOW dynamics demonstrates that a higher performance can be achieved by introducing a weighting parameter $\omega$ that refers to the sum of the rewarded experiences, i.e., $P_A + P_B$. Owing to this novelty, the high performance of TOW dynamics can be reproduced when implementing these dynamics with various volume-conserving physical objects. 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.

Acknowledgments

This work was partially undertaken when the authors belonged to the RIKEN Advanced Science Institute, which was reorganized and integrated into RIKEN as of the end of March 2013. We thank Professor Masahiko Hara for valuable discussions.

References

[1] Herken R 1995 *The Universal Turing Machine A Half-Century Survey* 2nd edn (Berlin: Springer)
[2] Sze S M 1981 *Physics of Semiconductor Devices* 2nd edn (New York: Wiley)
[3] Baker R J 2010 *CMOS Circuit Design, Layout, and Simulation* 3rd edn (New York: Wiley)
[4] Castro L N 2007 *Phys. Life Rev.* 4 1–36
[5] Kari L and Rozenberg G 2008 *Commun. ACM* 51 72–83
[6] Lee J, Imai K and Zhu Q 2012 *Inf. Sci.* 187 266–77
[7] Lai L, Jiang H and Poor H V 2008 *System and computers Proc. IEEE 42nd Asilomar Conference on Signals* pp 98–102
[8] Lai L, Gamal H E, Jiang H and Poor H V 2011 *IEEE Trans. Mobile Comput.* 10 239–53
[9] Agarwal D, Chen B C and Elango P 2009 *Proc. ICDM2009* pp 1–10
[10] Kocsis L and Szepesvári C 2006 *ECML2006, LNAI 4212* (Berlin: Springer) pp 282–93
[11] Gelly S, Wang Y, Munos R and Teytaud O 2006 *RR-6062-INRIA* pp 1–19
[12] Robbins H 1952 *Bull. Am. Math. Soc.* 58 527–36
[13] Thompson W 1933 *Biometrika* 25 285–94
[14] Gittins J and Jones D A 1974 *Progress in Statistics* (Amsterdam: North-Holland) pp 241–66
[15] Gittins J 1979 *J. R. Stat. Soc.* B 41 148–77
[16] Agrawal R 1995 *Adv. Appl. Probab.* 27 1054–78
[17] Auer P, Cesa-Bianchi N and Fischer P 2002 *Mach. Learn.* 47 235–56
[18] Kim S-J, Aono M and Hara M 2010 *UC2010, LNCS 6079* (Berlin: Springer) pp 69–80
[19] Kim S-J, Aono M and Hara M 2010 *BioSystems* 101 29–36
[20] Kim S-J, Nanesa E, Aono M and Hara M 2011 *Proc. NOLTA2011* pp 76–9
[21] Kim S-J, Aono M, Nanesa E and Hara M 2011 *Technical Report of IEICE* (CCS-2011-025) pp 36–41 (in Japanese)
[22] Aono M, Kim S-J, Hara M and Munakata T 2014 *BioSystems* 117 1–9
[23] Kim S-J and Aono M 2014 *NOLTA 5* 198–209
[24] Kim S-J, Naruse M, Aono M, Ohtsu M and Hara M 2013 *Sci. Rep.* 3 2370
[25] Naruse M et al 2014 *J. Appl. Phys.* 116 154303
[26] Kim S-J and Aono M 2015 *Spec. Issue Adv. Sci. Technol. Environmental*. B11 41–5