IEICE Proceeding Series

# Tug-of-war Model for Competitive Multi-armed Bandit Problem: Amoeba-inspired Algorithm for Cognitive Medium Access

Song-Ju Kim, Masashi Aono, Etsushi Nameda, Masahiko Hara

Vol. 1 pp. 590-593
Publication Date: 2014/03/17
Online ISSN: 2188-5079

2012 International Symposium on Nonlinear Theory and its Applications
NOLTA2012, Palma, Majorca, Spain, October 22-26, 2012

# Tug-of-war Model for Competitive Multi-armed Bandit Problem: Amoeba-inspired Algorithm for Cognitive Medium Access

Song-Ju Kim[†], Masashi Aono[†], Etsushi Nameda[†], and Masahiko Hara[†]

†Flucto-Order Functions Research Team, RIKEN-HYU Collaboration Research Center, RIKEN Advanced Science Institute
Fusion Technology Center 5F, Hanyang University, 222 Wangsimni-ro, Seongdong-gu, Seoul 133-791, Korea
2-1 Hirosawa, Wako-shi, Saitama 351-0198, Japan
Email: songju@riken.jp

**Abstract**—The "tug-of-war (TOW) model" is a unique parallel search algorithm for solving the multi-armed bandit problem (BP), which was inspired by the photoavoidance behavior of a single-celled amoeboid organism, the true slime mold *Physarum polycephalum* [1, 2, 3, 4, 5, 6]. "The cognitive medium access", which refers to multiuser channel allocations in cognitive radio, can be interpreted as "competitive multi-armed bandit problem (CBP) [14]." Unlike the normal BP, the reward (free channel) probability of a channel selected by more than one user is evenly split between selecting users. In this study, we propose the "solid TOW (STOW) model" for the CBP toward developing cognitive medium access protocols in uncertain environments. The aim of this study is to explore how can the users achieve the "social maximum", which is the most desirable state to obtain the maximum total score, in a decentralized manner. We show that the performance of the STOW model is higher than that of the well-known UCB1-tuned algorithm in many cases.

## 1. Introduction

Recently, various biologically-inspired computing algorithms, such as ant colony optimization [7], bee colony optimization [8], and so on, have been studied actively. In this study, we were inspired by a unicellular amoeboid organism, the plasmodium of the true slime mold *Physarum polycephalum*, that exhibits rich spatiotemporal oscillatory behavior and sophisticated computational capabilities [9]. We are interested in how the volume conservation law affects the information processing capabilities of the amoeba, and formulated the tug-of-war (TOW) model [1, 2, 3, 10].

In the TOW model, a number of branches of the amoeba act as search agents to collect information on light stimuli while conserving the total sum of their resources. The resource conservation law produces nonlocally-correlated search movements of the branches. We showed that the nonlocal correlation can be advantageous to manage the "exploration–exploitation dilemma", which is the tradeoff between the accuracy and speed in solving the "multi-armed bandit problem (BP)" . In this study, we concentrate on the minimal instances of the BP, i.e., two-armed cases, stated as follows. Consider two slot machines. Both machines have individual reward probabilities $P_A$ and $P_B$. At each trial, a player selects one of machines and obtains some reward, for example, a coin, with the corresponding probability. The player wants to maximize the total reward sum obtained after a certain number of selections. However, it is supposed that the player does not know these probabilities. The problem is to determine the optimal strategy for selecting the machine which yields maximum rewards by referring to past experiences.

In our previous studies [1, 2, 3, 4, 5, 6], we showed that the TOW model is more efficient than other well-known algorithms such as the modified $\epsilon$-greedy algorithm and modified softmax algorithm, and comparable to the "upper confidence bound1-tuned (UCB1T) algorithm" which is known as the best algorithm among non-parameter algorithms [11]. The algorithms for solving the problem are applicable to various fields, such as the Monte-Carlo tree search which is used in algorithms for the "game of GO" [12, 13], the cognitive radio [14, 15], web advertising, and so on.

In this study, we present our algorithm that is applied to the cognitive radio, and make comparisons on the performances of our TOW model and the UCB1T algorithm. Recently, the "cognitive medium access" problem is one of the hottest topic in the field of mobile communications [14, 15]. The underlying idea is to allow unlicensed users (i.e., cognitive users) to access the available spectrum when the licensed users (i.e., primary users) are not active. The "cognitive medium access" is a new medium access paradigm in which the cognitive users should not interfere with the licenced users.

Figure 1 shows the channel model proposed by Lai et al. [14, 15]. There is a primary network consisting of N channels, each with bandwidth B. The users in the primary network are operated in a synchronous time-slotted fashion. It is assumed that at each time slot, channel $i$ is free with probability $P_i$. The cognitive users do not know $P_i$ a priori. At each time slot, the cognitive users attempt to exploit the availability of channels in the primary network by sensing the activity in this channel model. In this setting, a single cognitive user can access only a single channel at any given time. The problem is to derive an optimal accessing strategy for choosing channels that maximizes the

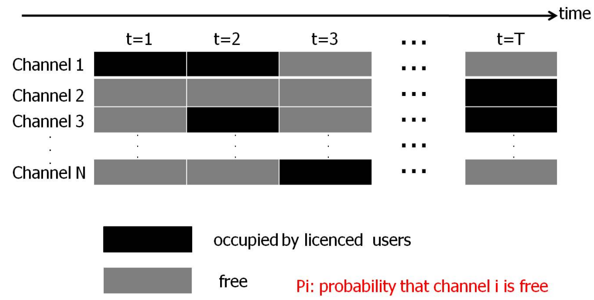
time
t=1
t=2
t=3
t=T
Channel 1
Channel 2
Channel 3
Channel N
occupied by licenced users
free
Pi: probability that channel i is free

Figure 1: Channel model.

expected throughput obtained by the cognitive user. This situation can be interpreted as the multiuser competitive bandit problem (CBP).

We consider the minimum CBP, i.e., 2 cognitive (unlicenced) users (1 and 2) and 2 channels (A and B). Each channel is not occupied by primary (licenced) users with the probability $P_i$. In the BP context, we assume that the user accessing a free channel can get some reward, for example a coin, with the probability $P_i$. Table 1 shows the pay-off matrix for user 1 and 2. If two cognitive users select the same channel, i.e. the collision occurs, the reward is evenly split between selecting users.

Table 1: Pay-off matrix for user 1 (user 2)

| | user 2: A | user 2: B |
|---|---|---|
| user 1: A | $P_A/2$ $(P_A/2)$ | $P_A$ $(P_B)$ |
| user 1: B | $P_B$ $(P_A)$ | $P_B/2$ $(P_B/2)$ |

In order to develop a unified framework for the design of efficient, and low complexity, cognitive medium access protocols, we have to seek an algorithm which can obtain the maximum total rewards (scores) in this context. We report the minimum results for the performance of the TOW model and the UCB1T algorithm as a candidate for the cognitive medium access in this study.

## 2. Solid TOW (STOW) Model

Many algorithms for the BP estimate the reward probability of each machine. In most cases, this "estimate" is updated only when the corresponding machine is selected. In contrast, the TOW model uses a unique learning method which is equivalent to that both estimates are updated simultaneously [5, 6]. The TOW model can imitate the system that determines its next moves at time $t+1$ in referring to the estimate of each machine even if it was not selected at time $t$, as if the two machines were selected simultaneously at time $t$. This unique feature is one of origins of the TOW's high performance.

In the previous study [6], we proposed the "solid" TOW (STOW) model which directly uses the advantage of the learning rule. Consider a rigid body like an iron bar, as shown in Fig. 2. Here, variable $X_k$ corresponds to the displacement of branch $k$ from an initial position, where $k \in \{A, B\}$. If $X_k$ is greater than 0, we consider that the body selects machine $k$. In the TOW model, the BP is represented in its inverse form, as we introduce "punishment" instead of "reward". That is, when machine $k$ is played, the player is "punished" with a probability $1 - P_k$.

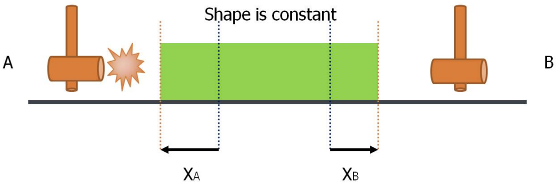

Figure 2: Solid TOW model.

We used the following estimate $Q_k$ $(k \in \{A, B\})$:

$$Q_k(t) = (N_k - L_k) - \sum_{\tau=0}^{t} \omega_e(\tau) l_k(\tau), \tag{1}$$

$$\omega_e(\tau) = \frac{2}{z(\tau)} - 1, \tag{2}$$

$$z(\tau) = \frac{L_1(\tau)}{N_1(\tau)} + \frac{L_2(\tau)}{N_2(\tau)}. \tag{3}$$

Here, $N_k$ is the number of playing machine $k$, and $L_k$ is the number of light stimuli (i.e., punishments) in $k$, where $l_k(t) = 1$ if light stimulus is applied at time $t$, otherwise 0.

The displacement $X_k$ $(k \in \{A, B\})$ is determined by the following difference equations:

$$X_A(t) = X_0 + Q_A(t) - Q_B(t) - \delta, \tag{4}$$

$$X_B(t) = X_0 + Q_B(t) - Q_A(t) + \delta, \tag{5}$$

$$\delta = \frac{a}{|d|} sin(\pi t + \pi/2), \tag{6}$$

$$d = \frac{N_A - L_A}{N_A} - \frac{N_B - L_B}{N_B}. \tag{7}$$

The body oscillates autonomously according to Eq. (6). The two parameters $X_0$ and $a$ are fixed as $X_0 = 0$ and $a = 0.35$ in this study. Consequently, +1 is added to $X_k$ if a reward (no light stimulus) occurs, or $-\omega_e(t)$ is added to $X_k$ if light stimulus is applied in each selected side.

## 3. Results: Performance Evaluation

### 3.1. easy problem instances

First, we consider problem instances such that $P_A < P_B$ and $P_A > P_B/2$, that are, $(P_A, P_B)$ = (0.2, 0.3), (0.3, 0.4), (0.4, 0.5), (0.5, 0.6), (0.6, 0.7), (0.7, 0.8), and (0.8, 0.9). When a user plays a machine which is different from the one that another user plays, we call this state "segregation", i.e., (user 1, user 2) = $(A, B)$ or $(B, A)$. The two users in the

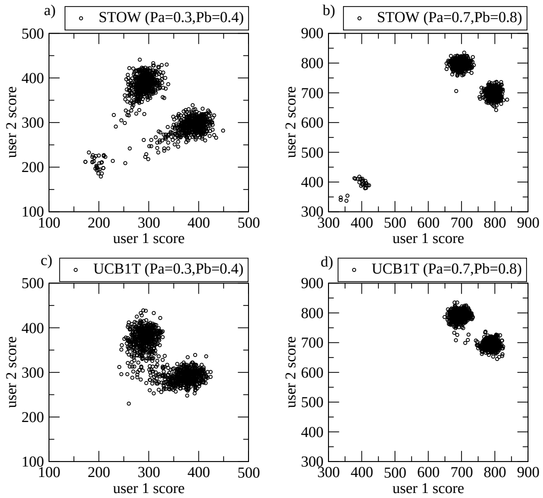

Figure 3: User scores of the STOW model and the UCB1T algorithm. An open circle denotes scores of user 1 (horizontal axis) and 2 (vertical axis) until 1000 selections for a sample. a) Scores of the STOW model for $P_A$=0.3 and $P_B$=0.4. b) Scores of the STOW model for $P_A$=0.7 and $P_B$=0.8. c) Scores of the UCB1T algorithm for $P_A$=0.3 and $P_B$=0.4. d) Scores of the UCB1T algorithm for $P_A$=0.7 and $P_B$=0.8.

segregation state will lose their rewards if they change their machines to play. Thus, the segregation state can be maintained stably as an "equilibrium". A state which gives the maximal total score, i.e., the maximal amount of rewards obtained by the two users, is called "social maximum" in the context of algorithmic game theory [16]. Here we designed each of the problem instances so that the segregation state corresponds to the social maximum.

The performance of each algorithm is evaluated in terms of the "score"; the accumulated amount of rewards that each user obtained over N plays. Figure 3 shows user scores of the STOW model (a) and b)) and the UCB1T algorithm (c) and d)) for $P_A$=0.3 and $P_B$=0.4, and $P_A$=0.7 and $P_B$=0.8, respectively. There are 1000 open circles for each figure because we used 1000 samples. An open circle denotes scores of user 1 (horizontal axis) and 2 (vertical axis) until 1000 selections for a sample.

There are two clusters of points in Fig. 3c and d. These clusters give the social maximum as they correspond to the segregation equilibrium such that (user 1, user 2) = ($A$, $B$) or ($B$, $A$). For $P_A$=0.3 and $P_B$=0.4 case (Fig. 3c), (user 1 score, user 2 score) = (300, 400) or (400, 300). For $P_A$=0.7 and $P_B$=0.8 case (Fig. 3d),(user 1 score, user 2 score) = (700, 800) or (800, 700).

On the other hand, there are three or four clusters in the STOW model. Larger two clusters correspond to the segregation equilibrium, and other smaller clusters correspond to the collision points due to some estimate-errors. The STOW model always estimates the $P_A + P_B$ by its own internal variables. Although this estimate generates the high

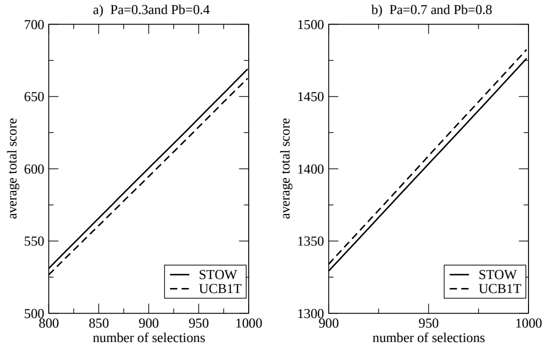

Figure 4: Average total scores of the STOW model (solid line) and the UCB1T algorithm (dashed line) for a) $P_A$=0.3 and $P_B$=0.4, and b) $P_A$=0.7 and $P_B$=0.8, respectively.

performance of the STOW model, estimate-errors occur in a small number of samples.

Despite the estimate-errors in the STOW model, total scores are comparable to the UCB1T algorithm as shown in Fig. 4. For $P_A$=0.3 and $P_B$=0.4 case, the average total score of the STOW model is higher than that of the UCB1T algorithm, while the average total score of the STOW model is lower than that of the UCB1T algorithm for $P_A$=0.7 and $P_B$=0.8 case.

Figure 5a also shows that the average total scores of the STOW model are comparable to those of the UCB1T algorithm. However, analyzing the results more precisely, we confirmed that the STOW is a superior algorithm than the UCB1T as shown in Fig. 5b. The superiority is defines as the difference of the average total scores between the two algorithms, divided by the average total score of the UCB1T algorithm. For ($P_A$, $P_B$) = (0.2, 0.3), (0.3, 0.4), and (0.4, 0.5) cases, the average total score of the STOW model is higher than that of the UCB1T algorithm. The superiority values are order of 0.01. Although the average total score of the STOW model is lower than that of the UCB1T algorithm for remained four cases, the superiority

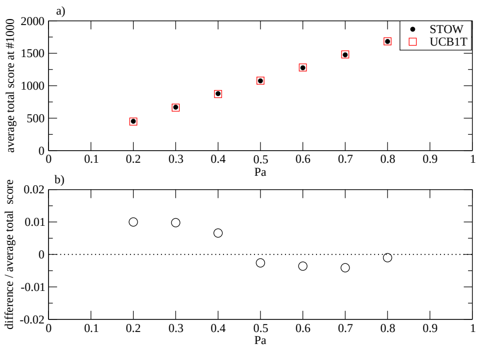

Figure 5: a) Average total scores of the STOW model (filled circle) and the UCB1T algorithm (open squre) until 1000 selections for each problem. b) The superiority of the STOW model compared to the UCB1T algorithm.

values are order of −0.001 which is small enough to consider as an accidental error. This means that the average total score of the STOW model is almost the same as that of the UCB1T algorithm for remained four cases.

### 3.2. hard problem instances

Secondly, we consider problem instances such that $P_A < P_B$ and $P_A < P_B/2$, that are, $(P_A, P_B)$ = (0.1, 0.3), (0.2, 0.5), (0.3, 0.7), and (0.4, 0.9). In contrast to the first instances in which the segregation states are equilibria and social maxima, these second instances were designed so that the segregation states are social maxima but cannot be the equilibria. Instead, the Nash equilibrium $(P_B/2, P_B/2)$ exists. The second instances are harder than the first ones, because the users need to avoid the Nash equilibrium to obtain the maximal total score.

Figure 6 shows that the average total scores of the STOW model (filled circle) and the UCB1T algorithm (open squre) until 1000 selections for each hard problem instance, and the superiority of the STOW model compared to the UCB1T algorithm (open circle). In all cases, the average total score of the STOW model is higher than that of the UCB1T algorithm except for (0.1, 0.3) case. [1] The superiority values are −0.02924, 0.01419, 0.05284, and 0.07911, respectively. These results imply that our STOW model is advantageous when used for harder problem instances in which naive methods to reach an equilibrium cannot achieve the maximal total score.

## 4. Conclusion

In this study, we presented a new form of the TOW model, the STOW model, which is applied to the cognitive radio. We showed that the performance of the STOW model is better than that of the UCB1T algorithm, especially for the hard problem instances in which the users should not be attracted to the Nash equilibria to achieve the social maximum. What kind of direct user interaction is needed for the realization of the social maximum? These are open questions left for us to design efficient and low-complexity cognitive medium access protocols in the future.

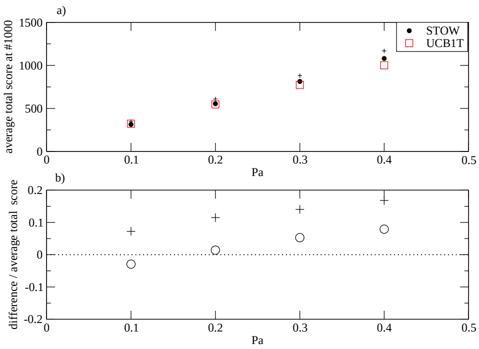

Figure 6: a) Average total scores of the STOW model (filled circle) and the UCB1T algorithm (open squre) until 1000 selections for each problem. b) The superiority of the STOW model compared to the UCB1T algorithm (open circle).

[1] The "plus" in figure 6b denotes the superiority with the another fixed parameter $a = 0.01$.

## References

[1] S. -J. Kim, M. Aono, M. Hara, *UC2009, LNCS 5715*, Springer, p.289, 2009.

[2] S. -J. Kim, M. Aono, M. Hara, *UC2010, LNCS 6079*, Springer, pp.69–80, 2010.

[3] S. -J. Kim, M. Aono, M. Hara, *BioSystems* vol.101, pp.29–36, 2010.

[4] S. -J. Kim, M. Aono, M. Hara, *Proc. of NOLTA2010*, pp.520–523, 2010.

[5] S. -J. Kim, E. Nameda, M. Aono, M. Hara, *Proc. of NOLTA2011*, pp.176–179, 2011.

[6] S. -J. Kim, M. Aono, E. Nameda, M. Hara, *Technical Report of IEICE (CCS-2011-025)*, pp.36–41, [in Japanese], 2011.

[7] M. Dorigo, L. M. Gambardella, *Artificial Life* vol.5 no. 2, pp.137–172, 1999.

[8] D. Karaboga, *Technical Report-TR06, Erciyes University*, 2005.

[9] M. Aono, Y. Hirata, M. Hara, K. Aihara, *New Generation Computing* vol.27, pp.129–157, 2009.

[10] M. Aono, Y. Hirata, M. Hara, K. Aihara, *UC2009, LNCS 5715*, Springer, pp.56–69, 2009.

[11] P. Auer, N. Cesa-Bianchi, P. Fischer, *Machine Learning* vol.47, pp.235–256, 2002.

[12] L. Kocsis, C. Szepesv*á*ri, *ECML2006, LNAI 4212*, Springer, pp.282–293, 2006.

[13] S. Gelly, Y. Wang, R. Munos, O. Teytaud, *RR-6062-INRIA*, pp.1–19, 2006.

[14] L. Lai, H. Jiang, H. V. Poor, *Proc. of IEEE 42nd Asilomar Conference on Signals, System and Computers*, pp.98–102, 2008.

[15] L. Lai, H. E. Gamal, H. Jiang, H. V. Poor, *IEEE Trans. on Mobile Computing*, vol.10 no.2, pp.239–253, 2011.

[16] T. Roughgarden, "Selfish routing and the price of anarchy", The MIT Press, Cambridge, (2005).