A Two-Armed Bandit Based Scheme for Accelerated Decentralized Learning

Ole-Christoffer Granmo and Sondre Glimsdal

Department of ICT, University of Agder, Grimstad, Norway

Abstract. The two-armed bandit problem is a classical optimization problem where a decision maker sequentially pulls one of two arms attached to a gambling machine, with each pull resulting in a random reward. The reward distributions are unknown, and thus, one must balance between exploiting existing knowledge about the arms, and obtaining new information. Bandit problems are particularly fascinating because a large class of real world problems, including routing, QoS control, game playing, and resource allocation, can be solved in a decentralized manner when modeled as a system of interacting gambling machines.

Although computationally intractable in many cases, Bayesian methods provide a standard for optimal decision making. This paper proposes a novel scheme for decentralized decision making based on the Goore Game in which each decision maker is inherently Bayesian in nature, yet avoids computational intractability by relying simply on updating the hyper parameters of sibling conjugate priors, and on random sampling from these posteriors. We further report theoretical results on the variance of the random rewards experienced by each individual decision maker. Based on these theoretical results, each decision maker is able to accelerate its own learning by taking advantage of the increasingly more reliable feedback that is obtained as exploration gradually turns into exploitation in bandit problem based learning.

Extensive experiments demonstrate that the accelerated learning allows us to combine the benefits of conservative learning, which is high accuracy, with the benefits of hurried learning, which is fast convergence. In this manner, our scheme outperforms recently proposed Goore Game solution schemes, where one has to trade off accuracy with speed. We thus believe that our methodology opens avenues for improved performance in a number of applications of bandit based decentralized decision making.

Keywords: Bandit Problems, Goore Game, Bayesian Learning.

1 Introduction

The conflict between exploration and exploitation is a well-known problem in reinforcement learning, and other areas of artificial intelligence. The *Two-Armed Bandit* (TAB) problem captures the essence of this conflict. In brief, a decision maker sequentially pulls one of two arms attached to a gambling machine, with

K.G. Mehrotra et al. (Eds.): IEA/AIE 2011, Part II, LNAI 6704, pp. 532–541, 2011.

each pull resulting in a random reward. The reward distributions are unknown, and thus, one must balance between exploiting existing knowledge about the arms, and obtaining new information. Multiple *interacting* bandits problems are particularly fascinating because they can be used to model and efficiently solve a large class of real world decentralized decision making problems, such as QoS-control in sensor networks [1].

In [2] we proposed a *Bayesian* technique for solving bandit like problems, akin to the *Thompson Sampling* [3] principle, leading to novel schemes for handling multi-armed and non-stationary (restless) bandit problems [4,5]. Empirical results demonstrated the advantages of these techniques over established top performers. Furthermore, we provided theoretical results stating that the original technique is instantaneously self-correcting and that it converges to only pulling the optimal arm with probability as close to unity as desired. Later on, as a further testimony to the renewed importance of the Thompson Sampling principle, a modern Bayesian look at the multi-armed bandit problem was also taken in [6,7].

In decentralized decision making problems, however, a certain phenomenon renders current bandit problem based solutions sub-optimal. Specifically, multiple decentralized decision makers are simultaneously exploring a collection of interacting bandits. This means that the variances of the reward distributions of each bandit problem are governed by the current level of exploration being manifested in the system as a whole. In other words, the variance of the reward distributions will be fluctuating with the degree of exploration taking place. Thus, initially, when exploration typically is significant, each decision maker should be correspondingly more conservative or cautious when interpreting the received rewards. Otherwise, by being too reckless, the decision maker may be led astray early on, converging to a sub-optimal decision.

The traditional approach to dealing with the above described fluctuation of reward distribution variance is to make learning sufficiently conservative. The purpose is to minimize the chance of each decision maker converging prematurely. Obviously, the disadvantages of this approach is the corresponding loss in learning speed caused by being too conservative also when exploration calms down. A recent approach deals with this problem indirectly by incorporating a Kalman filter into the decision making [5], allowing each decision maker to track changing reward distributions. Thus, too reckless learning initially is offset by the "forgetting" mechanism of the Kalman filter. This means that premature convergence is hindered. Yet, this tracking of changing reward distributions also means that exploration never stops. The decision makers will, as a result, never converge to a single optimal decision.

In this paper, we propose a novel scheme for solving one particular class of decentralized decision making problems, namely, the *Goore Game* (GG) [8]. The novel scheme we present here addresses directly and specifically fluctuating reward distribution variances. When a decision maker chooses which arm to pull, it also submits a measurement of its degree of exploration, which we refer to as *arm selection variance*. In turn, along with the random reward it receives from the arm pull, it also receives a signal that reflects the current aggregated level of exploration being manifested in the system. Based on this signal, each decision maker is effectively made able to *accelerate* its learning, taking advantage of the increasingly more reliable feedback that can be obtained when exploration gradually turns into exploitation.

2 The Goore Game (GG)

One of the most fascinating games studied in the field of artificial games is the GG. We describe it using the following informal formulation given in [9].

> *Imagine a large room containing N cubicles and a raised platform. One person (voter) sits in each cubicle and a Referee stands on the platform. The Referee conducts a series of voting rounds as follows. On each round the voters vote "Yes" or "No" (the issue is unimportant) simultaneously and independently (they do not see each other) and the Referee counts the fraction, λ, of "Yes" votes. The Referee has a uni-modal performance criterion $G(\lambda)$, which is optimized when the fraction of "Yes" votes is exactly λ^*. The current voting round ends with the Referee awarding a dollar with probability $G(\lambda)$ and assessing a dollar with probability $1 - G(\lambda)$ to every voter independently. On the basis of their individual gains and losses, the voters then decide, again independently, how to cast their votes on the next round.*

The game has many interesting features which render it both non-trivial and intriguing. It is a essentially a *distributed* game. Furthermore, the players of the game are ignorant of all of the parameters of the game. All they know is that they have to make a choice, for which they are either rewarded or penalized. They have no clue as to how many other players there are, how they are playing, or even of how/why they are rewarded/penalized. Finally, the stochastic function used to reward or penalize the players can be completely arbitrary, as long as it is uni-modal. The literature concerning the GG is sparse. It was initially studied in the general learning domain, and, as far as we know, was for a long time merely considered as an interesting pathological game. Recently, however, the GG has found important applications within two main areas, namely, QoS (Quality of Service) support in wireless sensor networks [10] and within cooperative mobile robotics as summarized in [11].

3 Accelerated Decentralized Learning in Two-Armed Bandit Based Decision Making (ADL-TAB)

This paper proposes a novel scheme for decentralized decision making in which each decision maker is inherently Bayesian in nature, yet avoids computational intractability by relying simply on updating the hyper parameters of sibling conjugate priors, and on random sampling from these posteriors. Based on the sibling conjugate priors, we also measure the current degree of exploration and exploitation being manifested in the system as a whole. This allows each decision maker to accelerate its learning by taking advantage of the increasingly more reliable feedback that can be obtained when exploration gradually turns into exploitation.

3.1 Bayesian Sampling for Two-Armed Normal Bandits (BS-TANB)

At the heart of our decentralized decision making scheme, we find a Bayesian Sampling approach to Two-Armed Normal Bandits (BS-TANB) problems. A unique feature of BS-TANB is its computational simplicity, achieved by relying *implicitly* on Bayesian reasoning principles. Possessing a bell-shaped probability density function with mean μ and standard deviation σ

$$f(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{(x-\mu)^2}{\sigma^2}\right)},$$

the normal distribution, $N(\mu,\sigma)$, is central to BS-TANB. Essentially, BS-TANB uses the normal distribution for two purposes. First of all, it is used to provide a *Bayesian estimate* of the reward probabilities associated with each of the available bandit arms. Secondly, a pertinent feature of BS-TANB is that it uses the normal distribution as the basis for a *randomized arm selection mechanism*. The following algorithm contains the essence of BS-TANB (see [5] for further details).

Algorithm: BS-TANB
Input: Observation noise σ_{ob}^2.
Initialization: $\mu_0[1] = \mu_1[1] = A$; $\sigma_0[1] = \sigma_1[1] = B$; # *Typically, A can be set to* 0*, with B being sufficiently large.*
Method:
For $t = 1, 2, \ldots$ **Do**

1. For each *Arm*, $j \in \{0,1\}$, draw a value x_j randomly from the associated *normal* distribution, $N(\mu_j[t], \sigma_j[t])$.
2. Pull the *Arm* i whose drawn value x_i is the largest one:
$$\alpha[t] = i = \underset{j\in\{0,1\}}{\text{argmax}}\ x_j.$$
3. Receive a reward $\tilde{r}_i$ from pulling *Arm* i, and update parameters as follows:
   – *Arm* i:
$$\mu_i[t+1] = \frac{\sigma_i^2[t]\cdot\tilde{r}_i + \sigma_{\text{ob}}^2\cdot\mu_i[t]}{\sigma_i^2[t] + \sigma_{\text{ob}}^2}$$
$$\sigma_i^2[t+1] = \frac{\sigma_i^2[t]\sigma_{\text{ob}}^2}{\sigma_i^2[t] + \sigma_{\text{ob}}^2}$$
   – *Arm* $j \neq i$:
$$\mu_j[t+1] = \mu_j[t]$$
$$\sigma_j^2[t+1] = \sigma_j^2[t]$$

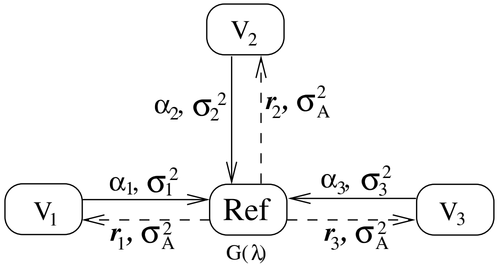

Fig. 1. Decentralized decision making with accelerated learning

As seen from the above BS-TANB algorithm, t is a discrete time index and the parameters $\phi^t = \langle(\mu_0[t], \sigma_0[t]), (\mu_1[t], \sigma_1[t])\rangle$ form an infinite 4-dimensional continuous state space, with each pair $(\mu_i[t], \sigma_i[t])$ giving the prior distribution of the unknown reward r_i associated with *Arm* i. Within Φ the BS-TANB navigates by transforming each prior distribution into a posterior distribution, based on the rewards $\tilde{r}_i$ obtained from selecting *Arm* i, $\alpha[t] = i$, as well as the observation noise σ^2_{ob}, given as an input parameter to the algorithm. Essentially, the algorithm uses observation noise σ^2_{ob} to determine how much emphasis to put on the reward $\tilde{r}_i$, which is a crucial property that we will now take advantage of.

In the interest of notational simplicity, let *Arm* 1, $\alpha[t] = 1$, be the arm under investigation. Then, for any parameter configuration $\phi^t \in \Phi$ we can state, using a generic notation[1], that the probability of selecting *Arm* 1, $\alpha[t] = 1$, is equal to the probability $P(X_1 > X_0|\phi^t)$ — the probability that a randomly drawn value $x_1 \in X_1$ is greater than the other randomly drawn value $x_0 \in X_0$ at time step t. Since the associated stochastic variables X_0 and X_1 are normally distributed, with parameters $(\mu_0[t], \sigma_0[t])$ and $(\mu_1[t], \sigma_1[t])$, respectively, we have that:

$$P(\alpha[t] = 1) = P(X_1 \geq X_0|\phi^t) = \int_{-\infty}^{0} f(x; \mu_0[t] - \mu_1[t], \sqrt{\sigma_0^2[t] + \sigma_1^2[t]}) \quad (1)$$

In the following, we will let $p[t]$ denote this latter probability.

3.2 BS-TANB Based Decentralized Decision Making

The overall decentralized decision making scheme is illustrated in Fig. 1. On each round t, the n decision makers $V_q \in \{V_1, \ldots, V_n\}$ choose one of two arms $\alpha_q[t] = i \in \{0, 1\}$ simultaneously and independently (they do not see each other), with $\alpha_q[t] = 0$ refering to a "No"-vote and $\alpha_q[t] = 1$ refering to a "Yes"-vote.

Let $p_q[t] = P(\alpha_q[t] = 1)$ be the probability that decision maker V_q casts a "Yes" vote on round t. Then $1 - p_q[t]$ is the probability that V_q casts a "No" vote, and each voting $\alpha_q[t]$ can be seen as a Bernoulli trial in which a "Yes" vote is a success and a "No" vote is a failure. Note that the concrete instantiation of

[1] By this we mean that P is not a fixed function. Rather, it denotes the probability function for a random variable, given as an argument to P.

the arm selection probability $p_q[t]$ is governed by the learning scheme applied, which in our case is BS-TANB.

Definition 1 (Arm Selection Variance). *In a two-armed bandit problem where the current arm selection probability is p, we define* Arm Selection Variance*, σ^2, to be the variance, $p(1-p)$, of the outcome of the corresponding Bernoulli trial.*

As seen in Fig. 1, in addition to casting a vote $\alpha_q[t]$, each decision maker V_q also submits its present *Arm Selection Variance*, $\sigma_q^2[t]$, in order to signal its level of exploration. Thus, as in the traditional Goore Game setup, a Referee calculates the fraction, $\lambda[t]$, of "Yes" votes. In addition, it now also calculates the variance $\sigma_A^2[t]$ of the total number of "Yes" votes, which simply is the sum of the variances of the independently cast votes (cf. Bienayme formula): $\sigma_A^2[t] = \sum_{q=1}^{n} \sigma_q^2[t]$. Note that in practice, such as in QoS control in sensor networks [1], this operation is conducted by the so-called base station of the network.

The Referee has a uni-modal *normally distributed* performance criterion $G(\lambda[t]; \mu_G, \sigma_G)$, where μ_G is the mean and σ_G^2 is the variance, which is thus optimized when the fraction of "Yes" votes is exactly μ_G, $\lambda[t] = \mu_G$. The current voting round ends with the Referee awarding a reward $\tilde{r}_i$ to each voter, with the reward being of magnitude $G(\lambda[t]; \mu_G, \sigma_G)$. Additionally, white noise $N(0, \sigma_W)$ is independently added to the reward received by each voter.

On the basis of their individual gains, the voters then decide, again independently, how to cast their votes on the next round.

3.3 Measuring Fluctuating Observation Noise in Goore Games

In order to develop a decentralized BS-TANB based scheme for solving the above problem, whose accuracy does not rely merely on conservative learning, it is crucial that we are able to determine the observation noise (σ_{ob}^2), needed by BS-TANB for its Bayesian computations.

From the perspective of voter V_q, let $Y_q = \sum_{r \neq q} \alpha_r[t]$ be the total number of "Yes" votes found among the $n-1$ votes cast by the other voters ($r \neq q$). According to our Bayesian bandit scheme, each voter V_q, at any given iteration t of the game, cast its vote according to a Bernoulli distribution with success probability $p_q[t] = P(\alpha_q[t] = 1) = P(X_1 > X_0 | \phi_q^t)$ — the probability of voting "Yes". Furthermore, initially, all voters vote "Yes" with probability $p_q[1] = 0.5$, and based on Bayesian computations, gradually shift their probability of voting "Yes" towards either 0 or 1, as learning proceeds. This leads us to design a solution for the case where Y_q is a sum of independent random variables of similar magnitude, in other words, where Y_q is approximately normally distributed for large n, $Y_q \sim N(\mu_F^q, \sigma_F^q)$. Since each term in the summation is Bernoulli distributed, the mean of the sum becomes $\mu_F^q = \sum_{r \neq q} p_r[t]$ while the variance becomes ${\sigma_F^q}^2 = \sum_{r \neq q} p_r[t](1 - p_r[t])$. The above means that, essentially, we may assume that each voter V_q decides whether to add an additional "Yes" vote or not to a random sum of yes votes, $Y_q \sim N(\mu_F^q, \sigma_F^q)$. That is, the reward that voter V_q receives when he votes either "Yes" ($\alpha_q[t] = 1$) or "No" ($\alpha_q[t] = 0$), becomes a function $G(\frac{Y_q+\alpha_q[t]}{n})$ governed by the random variable $Y_q \sim N(\mu_F^q, \sigma_F^q)$ as well as the decision α_q of voter V_q.

Thus $E[G(\frac{Y_q+\alpha_q[t]}{n})]$ is the expected reward received by voter V_q when pulling arm $\alpha_q[t]$ and $Var[G(\frac{Y_q+\alpha_q[t]}{n})]$ is the variance of the reward, which we will refer to as observation noise, σ_{ob}.

Lemma 1. *Let X be a normally distributed random variable, $X \sim N(\mu_F, \sigma_F)$. The expected value $E[G(X)]$ of a deterministic function $G(X) \sim N(\mu_G, \sigma_G)$ of X then becomes:*

$$E[G(X)] = \frac{1}{\sqrt{2\pi(\sigma_G^2 + \sigma_F^2)}} e^{-\frac{(\mu_G-\mu_F)^2}{2(\sigma_G^2+\sigma_F^2)}} \tag{2}$$

Proof. The proof is found in [12] and is omitted here in the interest of brevity.

Lemma 2. *A deterministic function $G(X) \sim N(\mu_G, \sigma_G)$ of a normally distributed random variable, $X \sim N(\mu_F, \sigma_F)$, has the variance:*

$$Var[G(X)] = \frac{\sqrt{\sigma_F^2\sigma_G^2}}{2\pi\sigma_F\sigma_G^2\sqrt{\sigma_G^2+2\sigma_F^2}} e^{-\frac{(\mu_G-\mu_F)^2}{\sigma_G^2+2\sigma_F^2}} - \frac{e^{-\frac{(\mu_G-\mu_F)^2}{\sigma_G^2+\sigma_F^2}}}{2\pi(\sigma_G^2+\sigma_F^2)} \tag{3}$$

Proof. The proof is found in [12] and is omitted here in the interest of brevity.

Since both the mean of G, μ_G, and the mean of F, μ_F, are unknown, the latter equation cannot be used directly to guide the bandit based learning. Instead, we consider the maximum of $Var[G(X)]$ in terms of $\mu_F \in (0, 1)$. By considering the maximum, learning accuracy is prioritized, at the potential cost of reduced learning speed. It turns out that both μ_F and μ_G is eliminated from the equation when only considering the maximum of $Var[G(X)]$ with respect to μ_F, as follows.

Theorem 1. *The maximum of the variance $Var[G(X)]$ with respect to μ_F of the function $G(X) \sim N(\mu_G, \sigma_G)$, with $X \sim N(\mu_F, \sigma_F)$, is:*

$$\max_{\mu_F \in (0,1)} Var[G(X)] = \sigma_F^2 \left(\sigma_G^2+\sigma_F^2\right)^2 \frac{e^{\frac{\sigma_G^2\left(\log\left(|\sigma_F|\left(\sigma_G^4+2\,\sigma_F^2\,\sigma_G^2+\sigma_F^4\right)|\sigma_G|\right)-\log\left(\sigma_F\,\sigma_G^2\left(\sigma_G^2+2\,\sigma_F^2\right)^{\frac{3}{2}}\right)\right)}{\sigma_F^2}}}{2\,\pi\,\sigma_G^2\left(\sigma_G^2+2\,\sigma_F^2\right)^3} \tag{4}$$

Proof. The proof is found in [12] and is omitted here in the interest of brevity.

In other words, since σ_F in the above equation can be approximated based on the feedback σ_A from the Referee (see Fig. 1), we can find the worst case observation noise based on Theorem 1. Thus, essentially, we have found a closed form formula that approximates the worst case observation noise σ_{ob} that each voter can apply adaptively in its Bayesian computations.

4 Empirical Results

In this section we evaluate the ADL-TAB scheme by comparing it with the currently best performing algorithm — the family of *Bayesian* techniques reported in [2]. Based on our comparison with these "reference" algorithms, it should be quite straightforward to also relate the ADL-TAB performance results to the performance of other similar algorithms.

We have conducted numerous experiments using various reward distributions, including a wide range of $G(\lambda)$-functions and a wide range of voters, under varying degrees of observation noise. The full range of empirical results are reported in [12], and they all show the same trend. Thus, in this paper, we report performance on a representative subset of the experiment configurations, involving the 3, 5, and 10 player Goore Game. Performance is measured in terms of *Regret — the difference between the sum of rewards expected after N successive rounds of the GG, and what would have been obtained by always casting the optimal number of "Yes" votes.*

For these experiment configurations, an ensemble of 1000 independent replications with different random number streams was performed to minimize the variance of the reported results. In order to investigate the performance of the schemes under a broad spectrum of environments, we test the schemes using three different representative $G(\lambda)$ functions — one sloped, with optimum close to $\lambda = 0.5$, $G \sim N(0.35, 0.2)$, another one also sloped, but with optimum farther from $\lambda = 0.5$, $G \sim N(0.125, 0.2)$, and finally, one peaked reward function, also with optimum far from $\lambda = 0.5$ (thus, being the most challenging one). In Table 1, Regret is reported after 10, 100, 1000, and 10 000 iterations for both the new *accelerating* scheme and the traditional *static* scheme.

As seen from the table, for all reported configurations, our ADL-TAB scheme not only learns faster initially, but also attains the best regret in the long run. Note that for the two bottom configurations, we use an augmented σ_F, $\widehat{\sigma_F} = c \cdot \sigma_F$, with $c = 1.5$, when the final observation noise σ_{ob} is calculated. Indeed, the constant c can be used to handle the non-stationarity arising as the number of voters grows, as demonstrated in Table 2.

Since ADL-TAB applies the standard deviation σ_G of the reward function $G(\lambda)$ to find overall observation variance, it is interesting to see how robust the scheme is to distortion of σ_G. As summarized in Table 3, setting σ_G too low is better than setting it too high in the present setting. Indeed, performance improves slightly with a lower σ_G.

Note that the above reported performance gap is reduced with the level of white noise added to G, as shown in Table 4. As the variance of the white noise raises to extreme values, the white noise dominates the overall observation noise, rendering the variance introduced by the voters insignificant. However, for realistic degrees of white noise, as also seen from the table, ADL-TAB clearly outperforms the static BS-TANB scheme.

Thus, based on our empirical results, we conclude that ADL-TAB is the superior choice for the Goore Game, both when σ_G is known or slightly distorted, providing significantly better performance in all experiment configurations.

Table 1. Regret after 10, 100, 1000, and 10 000 iterations for 10 players

Scheme	**#Players**	**Function**	**10**	**100**	**1000**	**10 000**
Accelerating	3	$G \sim N(0.125, 0.1)$	11.56	26.72	30.96	33.17
Static	3	$G \sim N(0.125, 0.1)$	11.63	27.27	34.88	47.20
Accelerating	3	$G \sim N(0.125, 0.2)$	5.26	8.47	10.35	11.09
Static	3	$G \sim N(0.125, 0.2)$	5.28	9.53	15.15	25.15
Accelerating	3	$G \sim N(0.375, 0.2)$	6.62	10.86	11.99	12.63
Static	3	$G \sim N(0.375, 0.2)$	6.73	12.15	14.36	17.72
Accelerating	5	$G \sim N(0.125, 0.1)$	18.37	41.60	51.78	61.65
Static	5	$G \sim N(0.125, 0.1)$	18.28	44.94	58.52	99.49
Accelerating	5	$G \sim N(0.125, 0.2)$	6.92	12.94	22.99	60.80
Static	5	$G \sim N(0.125, 0.2)$	7.01	15.39	32.94	69.86
Accelerating	5	$G \sim N(0.375, 0.2)$	6.12	20.47	22.70	25.75
Static	5	$G \sim N(0.375, 0.2)$	6.16	24.24	30.11	35.74
Accelerating	10	$G \sim N(0.125, 0.1)$	32.81	93.82	133.65	443.7
Static	10	$G \sim N(0.125, 0.1)$	32.84	99.27	143.67	549.8
Accelerating	10	$G \sim N(0.125, 0.2)$	10.19	19.57	39.58	110.53
Static	10	$G \sim N(0.125, 0.2)$	10.21	22.57	56.91	167.09
Accelerating	10	$G \sim N(0.375, 0.2)$	4.40	31.20	113.42	116.65
Static	10	$G \sim N(0.375, 0.2)$	4.41	32.03	163.40	197.31

Table 2. Performance with σ_F augmented, $\widehat{\sigma_F} = c \cdot \sigma_F$ (10 players, $G \sim N(0.1, 0.1)$, $\sigma_W = 0.1$)

Scheme / c	1.0	1.25	1.5	1.75
Accelerating	1030.0	684.7	444.7	408.7
Static	965.6	624.3	550.4	414.2

Table 3. Performance with distorted $\widehat{\sigma}_G$ given to ADL-TAB (10 players, $G \sim N(0.125, 0.2)$, $\sigma_W = 0.1$)

$\widehat{\sigma}_G$	$0.85 \cdot \sigma_G$	$0.90 \cdot \sigma_G$	$0.95 \cdot \sigma_G$	$1.0 \cdot \sigma_G$	$1.05 \cdot \sigma_G$	$1.10 \cdot \sigma_G$	$1.15 \cdot \sigma_G$
Regret	74.4	75.7	90.9	123.5	162.9	194.4	237.5

Table 4. Performance with varying degrees of white noise $N(0, \sigma_W)$ (10 players, $G \sim N(0.375, 0.2)$)

Scheme / σ_W	0.01	0.05	0.1	0.5	1.0	5.0
Accelerating	56.6	54.8	61.1	123.4	315.8	2012.0
Static	120.5	121.4	121.6	184.4	371.7	2013.3

5 Conclusion and Further Work

In this paper we proposed a novel scheme, ADL-TAB, for decentralized decision making based on the Goore Game. Theoretical results concerning the variance of the observations made by each individual decision maker, enabled us to accelerate learning as exploration turns into exploitation. Indeed, our empirical results demonstrated that the accelerated learning improves both learning accuracy and speed, outperforming state-of-the-art Goore Game solution schemes. As further work, we intend to study how the Kalman filter can be incorporated into ADL-TAB, so that non-stationary behavior can be modeled and addressed in a principled manner. We are also currently investigating how the present result can be extended to other classes of decentralized decision making problems.

References

1. Iyer, R., Kleinrock, L.: QoS Control For Sensor Networks. In: IEEE International Conference on Communications, vol. 1, pp. 517–521 (2003)
2. Granmo, O.C.: Solving Two-Armed Bernoulli Bandit Problems Using a Bayesian Learning Automaton. International Journal of Intelligent Computing and Cybernetics 3(2), 207–234 (2010)
3. Thompson, W.R.: On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. Biometrika 25, 285–294 (1933)
4. Norheim, T., Brådland, T., Granmo, O.C., Oommen, B.J.: A Generic Solution to Multi-Armed Bernoulli Bandit Problems Based on Random Sampling from Sibling Conjugate Priors. In: Proceedings of the Second International Conference on Agents and Artificial Intelligence (ICAART 2010), INSTICC, pp. 36–44 (2010)
5. Granmo, O.C., Berg, S.: Solving Non-Stationary Bandit Problems by Random Sampling from Sibling Kalman Filters. In: Proceedings of the Twenty Third International Conference on Industrial, Engineering, and Other Applications of Applied Intelligent Systems (IEA-AIE 2010), pp. 199–208. Springer, Heidelberg (2010)
6. Scott, S.L.: A modern Bayesian look at the multi-armed bandit. Applied Stochastic Models in Business and Industry (26), 639–658 (2010)
7. May, B.C., Korda, N., Lee, A., Leslie, D.S.: Optimistic Bayesian sampling in contextual-bandit problems. Submitted to the Annals of Applied Probability (2011)
8. Tsetlin, M.L.: Automaton Theory and Modeling of Biological Systems. Academic Press, London (1973)
9. Narendra, K.S., Thathachar, M.A.L.: Learning Automata: An Introduction. Prentice-Hall, Englewood Cliffs (1989)
10. Chen, D., Varshney, P.K.: QoS Support in Wireless Sensor Networks: A Survey. In: The 2004 International Conference on Wireless Networks, ICWN 2004 (2004)
11. Cao, Y.U., Fukunaga, A.S., Kahng, A.: Cooperative Mobile Robotics: Antecedents and Directions. Autonomous Robots 4(1), 7–27 (1997)
12. Granmo, O.C., Glimsdal, S.: Accelerated Bayesian Learning for Decentralized Two-Armed Bandit Solutions to the Goore Game (2010), Unabridged version of this paper