Applying Particle Swarm Optimization to Improve Average Reward in non-Stationary Multi-Armed Bandit

Elnaz Manifar, Behrooz Masoumi, and Mohammadreza Meybodi

***Abstract*—** Multi-Armed Bandit (MAB) is one of the reinforcement learning models, which is basically known to state the challenge of balancing exploration and exploitation. Many real-world learning and optimization problems can be modeled in this way. In this study, conventional techniques such as Pursuit, Softmax, and Reinforcement-Comparison, are implemented individually. Later, Particle Swarm Optimization (PSO) is integrated with these techniques to improve average reward in non-stationary and episodic multi-armed bandit. The comparative results present that the hybrid techniques improve average reward.

***Keywords*—** multi-armed bandit, particle swarm optimization, average reward, exploration, and exploitation.

I. Introduction

IN many real-world situations, we need to make decisions in order to maximize some numerical goals such as reward or profit. Sometimes, these decisions or the actions might not be fully observable by the time. Thus, we could utilize the known ones as well as, keep on exploring for new knowledge hidden in further found actions [1]. Since both approaches are not possible at the same time, the mentioned issue is addressed as the challenge of balancing exploration and exploitation. Undoubtedly, MAB problems are one of the best known methods to picture this challenge.

The multi-armed bandit problem is a mathematical model in reinforcement learning, which studies on the efficiency of information collection in stochastic and uncertain environment [2]. A decision-maker in each episode, is frequently asked to choose among n different actions and based on this choice, a numerical stochastic reward is returned. The final goal is to maximize the total sum of rewards over for example, 2000 times of action selection [3].

In the current studied bandit problem, each action is assigned a quantity called action value which is, derived by a standard normal distribution [4]. It is assumed that, there is no certain prior knowledge about this value by the time of choosing each action as well as, the bandit problem is non-stationary. The objective is to increase the returned average reward, over 2000 episodes of action selection, using a hybrid technique with PSO.

Heretofore, similar hybrid by Genetic algorithm has been done and produced approximately similar results to the conventional multi-armed bandit methods [3], [5].

To evaluate the proposed PSO hybrid technique, a number of simulations have been done. Based on the numerical results, the average reward is significantly improved after the hybrid.

This paper is organized as follow:
In section II, the non-stationary Multi-armed Bandit problems are defined. Sections III and IV, present the conventional techniques and the Particle Swarm Optimization approach, respectively and part V, belongs to the results and discussion.

II. Non-stationary Multi-armed Bandit Problem

Non-stationary MAB is one of the most challenging types of Bandit problems which the agents face increasing complexity of the environment. In such situations, there is no exact knowledge about action values thus, it will be more difficult to pick an action. The non-stationary multi-armed bandit problem can be formulated as follows:

A decision-making agent is facing n different actions *(a_i)*, *$i=1, 2, 3, \ldots, n$*, and is asked to choose only one among all, in each episode of play, t_j whereas, *$j=1, 2, 3, \ldots, K$.*

Clearly, after each action selection, based on the quality of decision, the agent is rewarded numerically $r_{tj}(a_i)$. The actual value of $r_{tj}(a_i)$ is called $Q^*_s(a_i)$ and is derived from a Normal probabilistic distribution with the mean 0 and variance 1. This value is initially unknown to the agents and the final goal is to maximize the received reward *J*, from the following equation:

$$\max J = \sum_{J=1}^{K} r_{tj} \tag{1}$$

III. Conventional methods

In this section, three suitable methods for non-stationary MAB are discussed. These methods called, Softmax, Pursuit, and Reinforcement comparison, are separately hybrided by PSO and have increased the average reward over the 2000

Elnaz Manifar, Faculty of Computer and Information Technology Engineering, Islamic Azad University, Qazvin branch, Iran, e.manifar@qiau.ac.ir.

Behrooz Masoumi, Faculty of Computer and Information Technology Engineering,, masoumi@qiau.ac.ir.

Mohammadreza Meybodi, Department of Computer Engineering
Amir kabir University of Technology, Tehran, Iran, mmeybodi@aut.ac.ir.

http://dx.doi.org/10.15242/IIE.E0114570

plays. The similar mechanism in all techniques is that, they keep an estimation of the action actual reward $Q^*_s\ (a_i)$, and update it whenever the action a_i is tried.

Let's have a look at these techniques:

A. Pursuit Method

Pursuit is one of the efficient learning methods for Multi-Armed Bandit which maintains both action value and the probability of choice. The dynamic type of this technique called Adaptive Pursuit is suitable for non-stationary MAB. In Adaptive Pursuit, There are boundaries for actions so that, neither of them could be less than $p_{\min}$ or proceed $p_{\max} = 1-(n-1)p_{\min}$ whereas n, is the number of possible actions. The probabilities of actions are updated before choosing each ones. Let a_g be the action with the highest value (greedy action) in the jth episode of play thus, the probability of choosing a is increased toward $p_{\max}$ and the rest are decreased toward $p_{\min}$. Therefore, $\sum_{i=1}^{n} p_{tj}(a_i) = 1$ is guaranteed. The updating equations are as follows:

$$p_{tj}(a_g) = p_{tj-1}(a_g) + \alpha(p_{\max} - p_{tj-1}(a_g)) \qquad (2)$$

$$p_{tj}(a_i) = p_{tj-1}(a_i) + \alpha(p_{\max} - p_{tj-1}(a_i)) \forall i \neq g \qquad (3)$$

Where α, 0<α<1, is a small positive parameter and a_i is the ith action. The action value estimation $Q_{tj}(a_i)$ in the jth episode of play, is updated after each time of trying a_i.

B. Softmax

Although choosing an action with the highest value (greedy action), is one of the most well-known way, but it has a significant weak point. The exploration process is done equally among all actions, either optimal or sub-optimal and considers the same probabilities for all actions to be chosen.

One way to come over this problem is to weigh the value of estimating functions for each action according to its optimality. In this case, the greedy action will have the highest probability to be chosen and the rest will receive different probabilities based on their values.

This approach is called Softmax which often uses Boltzman action selection rule. The most common Boltzman method uses the following distribution:

$$\frac{e^{Q_{t-1}(a)/T}}{\sum_b e^{Q_{t-1}(b)/T}} \qquad (4)$$

Where a is the chosen action in the tth play, b is any available action, and T is a positive parameter called temperature. By applying appropriate settings for T, Softmax will significantly work better on non-stationary MAB.

Higher values for T bring almost equal probabilities for all actions. Unlike, lower values make differences in the probabilities of actions. Some researchers might find Softmax a bit difficult since, it might require prior knowledge about T, action values, and e.

C. Reinforcement Comparison

The fundamental feature of reinforcement learning is that the more profitable actions, should own higher probabilities to be taken. In order to achieve that, Reinforcement-Comparison algorithms, not only maintain each action value estimations, but also an overall reward level and separate quantity for the action preferences.

Let $p_t(a)$, the priority of action a in the tth play then, according to Softmax equation, the probability of choosing a is as follows:

$$\pi_t(a) = P_r\{a_t = a\} = \frac{e^{p_{t-1}(a)}}{\sum_b e^{p_{t-1}(b)}} \qquad (5)$$

$$p_t(a_t) = p_t(a_t) + \beta(r_t - \bar{r}_t) \qquad (6)$$

$$\bar{r}_{t+1} = \bar{r}_t + \alpha(r_t - \bar{r}_t) \qquad (7)$$

Where $\pi_t(a)$ is the probability of choosing a in the tth episode of play. After each play, the action priority a_t, is incremented based on the difference between the reward r_t and the reference reward $\bar{r}_t$. α and β are the step-size.

IV. Particle Swarm Optimization

Particle swarm optimization is one of the well-known swarm intelligence approaches which was first introduced by Kennedy, Erbert and Shi to model social behavior [6], [7]. Afterwards, PSO was utilized in optimization problems as a global search method in the search space. This iterative algorithm, considers moving particles in the search space that each is, one of the possible answers. Each particle has two main traits; velocity and position.

In each iteration, all particles update their personal record according to their best experienced position and accelerate toward the particle with the best record. A simple PSO formulation and flowchart are as follows:

$$v_{ij}(t+1) = wv_{ij}(t) + r_1c_1(p_{ij}(t) - x_{ij}(t)) + r_2c_2(g\ (t) - x_{ij}(t)) \qquad (8)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \qquad (9)$$

Where v_{ij}, x_{ij} are respectively the velocity and position of the particle i^{th} toward the dimension of *j* in the search space. p_{ij} is the best experienced position by particle *i* and *g* is the best discovered position among all particles.

w, 0.4<w<0.9, is a small quantity called, inertia Coefficient. It enables PSO to explore for better records in the search space or exploit on the best found. The higher value of w, makes PSO to explore more and causes slower convergence, whereas the lower amount for w, makes the algorithm exploit on the best found records.

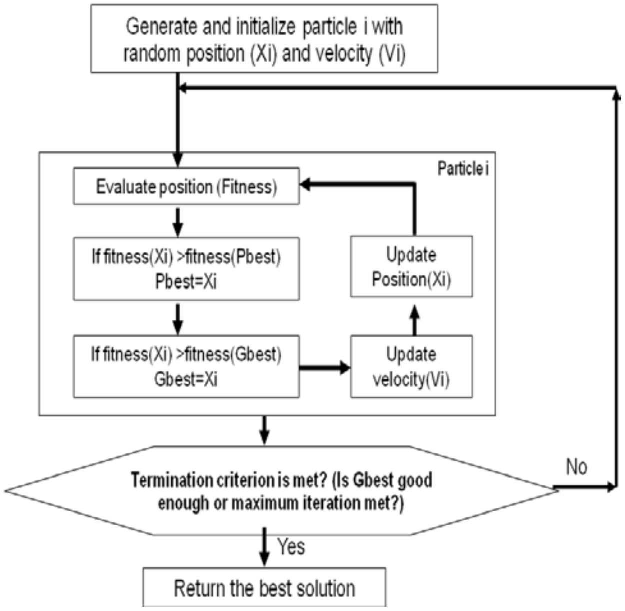
Generate and initialize particle i with random position (Xi) and velocity (Vi)
Particle i
Evaluate position (Fitness)
If fitness(Xi) >fitness(Pbest) Pbest=Xi
Update Position(Xi)
If fitness(Xi) >fitness(Gbest) Gbest=Xi
Update velocity(Vi)
Termination criterion is met? (Is Gbest good enough or maximum iteration met?)
No
Yes
Return the best solution

Fig. 1 Particle Swarm Optimization flowchart

V. Results and discussion

In this section, the results of conventional methods, both before and after hybrid with PSO, are presented.

The number of available actions for each player (agent) is set to 10 (*n=10*). The number of episodes (plays) is set to 2000 (*k=2000*), as well. As mentioned earlier, the actual value $Q^*_s (a_i)$ for each action is derived from a Normal distribution with the mean 0 and variance 1. Then, the average reward in each episode, is simply the mean of $Q^*_s(a_i)$ for all actions.

A. Conventional Methods

In this study, three important conventional methods for non-stationary Multi-Armed Bandit problem are implemented. These techniques are Pursuit, Softmax, and Reinforcement Comparison. Each method is implemented individually and compares to the hybrided one with PSO. For Pursuit method, α is set to 0.1 (α=0.1) and $p_{\max} = 1 - 9p_{\min}$ whereas $p_{\min}$ = 0.001. In Reinforcement Comparison method, both α and β are set to 0.1. Softmax technique is tested on two different temperatures, 0.1 and 0.01 which 0.1 significantly works better on our testbed.

B. PSO Hybrid Technique

In the proposed hybrid algorithm, 10 particles are corresponded to 10 available actions. The initial position of each is derived from the Uniform distribution $x_{ij}(t) \sim U(0,1)$. This value is the initial probability of choosing each action in Multi-Armed Bandit. The initial velocity of each particle is considered 0. The average reward in MAB is considered as the fitness function of PSO that after each action selection one value for action is noticed.

In every episode, each particle (action) updates its personal record -if notices any improvement- and accelerates toward the one with the best known fitness (average reward). Note that, the best known particle, is the action with the highest actual value, Q^*_s (a_i). The comparative results of conventional techniques, before and after hybrid with PSO are as follows:

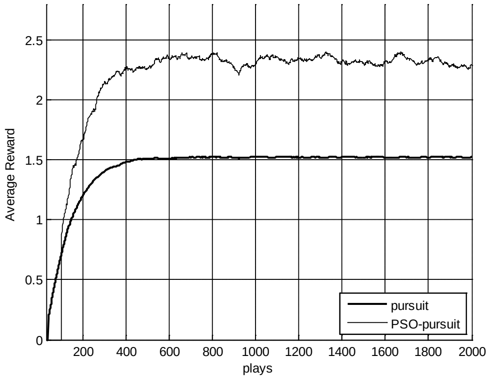

Fig. 2 Pursuit method before and after hybrid with PSO

TABLE I
Pursuit method Results before and After Hybrid with PSO

Method Average Reward	Episode									
	200	400	600	800	1000	1200	1400	1600	1800	2000
Pursuit	1.20	1.47	1.51	1.52	1.47	1.52	1.52	1.52	1.52	1.52
PSO.Pursuit	1.67	2.26	2.35	2.38	2.33	2.34	2.32	2.32	2.33	2.28

As the figure 2 and table I show, PSO-Pursuit (Hybrid) technique works significantly better for 0.76 units in average reward. While the classic Pursuit converges to 1.52, the hybrid technique exceeds 2.3.

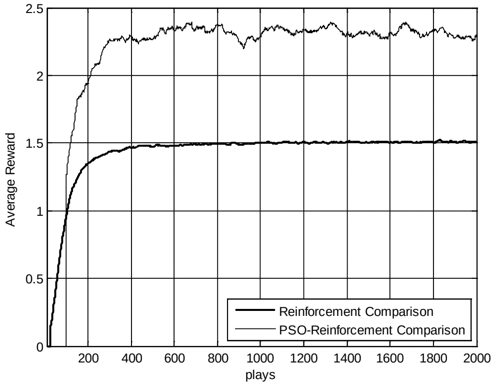

Fig. 3 Reinforcement-Comparison method before and after hybrid with PSO

http://dx.doi.org/10.15242/IIE.E0114570

TABLE II
REINFORCEMENT COMPARISON RESULTS BEFORE AND AFTER HYBRID WITH PSO

Method / Average Reward	Episode									
	200	400	600	800	1000	1200	1400	1600	1800	2000
Reinforcement Comparison	1.35	1.46	1.47	1.49	1.50	1.51	1.50	1.52	1.49	1.51
PSO_Reinforcement Comparison	1.67	2.29	3.34	3.37	2.29	2.33	2.30	2.32	2.33	2.28

In Reinforcement Comparison, the conventional technique works pretty similar to Pursuit and converges to 1.51 whereas the PSO_Reinforcement Comparison exceeds 3.3 in episodes 600 and 800.

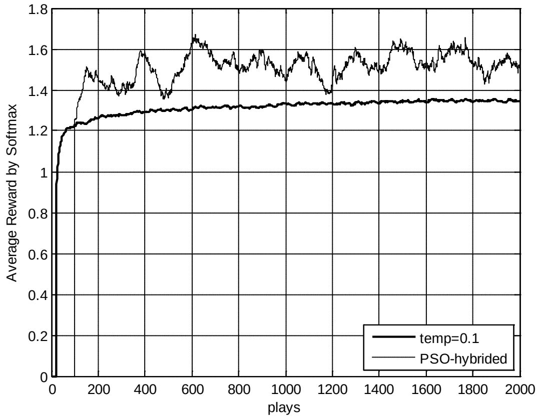

Fig. 4 Softmax method with the temp=0.1, before and after hybrid

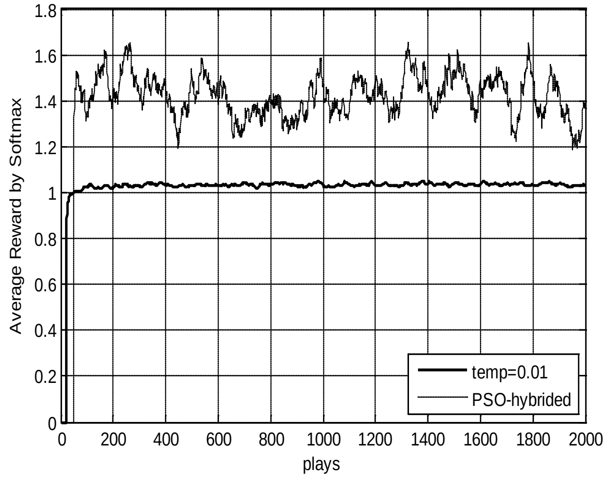

Fig. 5 Softmax method with the temp=0.01, before and after hybrid

TABLE III
SOFTMAX RESULTS (TEMP=0.1, 0.01) BEFORE AND AFTER HYBRID WITH PSO

Method / Average Reward	Episode									
	200	400	600	800	1000	1200	1400	1600	1800	2000
Softmax (Temp=0.01)	1.02	1.03	1.03	1.03	1.03	1.03	1.03	1.03	1.03	1.03
PSO-Hybrided (Temp=0.01)	1.43	1.45	1.34	1.34	1.4	1.44	1.46	1.48	1.43	1.35
Softmax (Temp=0.1)	1.24	1.28	1.3	1.31	1.32	1.33	1.33	1.34	1.34	1.34
PSO-Hybrided (Temp=0.1)	1.45	1.49	1.56	1.52	1.5	1.43	1.53	1.55	1.57	1.54

In Softmax methods, two temperatures 0.1 and 0.01 are tested. As figures 4, 5, and Table III show, the results of temperature 0.1 is superior to 0.01. Although, The PSO hybrid techniques improve the conventional Softmax for 0.21 to 0.41 units, it works worse than the hybrid on Pursuit and Reinforcement Comparison methods.

Total comparison results of the hybrid techniques, can be noticed in the following table.

TABLE IV
TOTAL COMPARATIVE RESULTS OF INTEGRATING CONVENTIONAL TECHNIQUES WITH PSO

Method / Average Reward	Episode									
	200	400	600	800	1000	1200	1400	1600	1800	2000
PSO_Pursuit	1.67	2.26	2.35	2.38	2.33	2.34	2.32	2.32	2.33	2.28
PSO_Reinforcement Comparison	1.67	2.29	3.34	3.37	2.29	2.33	2.30	2.32	2.33	2.28
PSO-Hybrided (Temp=0.01)	1.43	1.45	1.34	1.34	1.4	1.44	1.46	1.48	1.43	1.35
PSO-Hybrided (Temp=0.1)	1.45	1.49	1.56	1.52	1.5	1.43	1.53	1.55	1.57	1.54

As the table IV shows, the PSO hybrid technique works better on Pursuit and Reinforcement Comparison than Softmax. One of the reasons might be that setting the temperature in Boltzman distribution needs prior knowledge about the action values.

REFERENCES

[1] Joann`es Vermorel, and Mehryar Mohri, “Multi-armed Bandit Algorithms
and Empirical Evaluation, ” ECML, 2005 pp. 437–448.

[2] Ilya O. Ryzhov, and Warren B. Powell, “The value of information in multi-armed bandits with exponentially distributed rewards,” in *Rec. 2011 ICCS Int. Conf. Computational Science*, pp. 1363–1372.

[3] D. E. Koulouriotis, and A. Xanthopoulos, “Reinforcement learning and evolutionary algorithms for non-stationary multi-armed bandit problems, ” *Elsevier, Applied Mathematics and Computation* vol. 196, pp. 913–922, 2008.
http://dx.doi.org/10.1016/j.amc.2007.07.043

[4] Richard S. Sutton, and Andrew C. Barto, *Reinforcement Learning.* Evaluative Feedback, pp. 31–52.

[5] D. E. Koulouriotis, and A. Xanthopoulos, “A comparative study of ad hoc techniques and evolutionary methods for multi armed bandit problems,” *Operation Resource Intelligence*, vol. 8, pp. 105–122, Feb. 2008.

[6] J. Kennedy, and R. Eberhart, “Particle Swarm Optimization, ” *in Rec. 1995 IEEE Int. Conf. Neural Networks* , pp. 1942–1948.

[7] Y. Shi, and R.C. Eberhart, “A modified particle swarm optimizer,” *in Rec. 1998 IEEE Int. Conf. Evolutionary Computation*, pp. 69–73.

[8] Volodymyr Kuleshov, and *Doina* Precup, *“Algorithms for the multi armed bandit problem,” Machine Learning Research* vol. 1, pp. 1–48, 2000.

[9] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz, *“Pure exploration in finitely armed and continuous-armed bandits,” Theoretical Computer Science*, vol. 412, pp. 1832–1852, 2011.
http://dx.doi.org/10.1016/j.tcs.2010.12.059

[10] S´ebastien Bubeck, R´emi Munos, Gilles Stoltz, and Csaba Szepesv´ari, *“X-Armed Bandits,” Machine Learning Research*, vol. 12, pp. 1655-1695, France, May. 2011.

[11] R. Garbe, and K. D. Glazebrook, *“On a new approach to the analysis of* complex, Multi armed bandits,*” Mathematical Methods of Operations Resource*, vol. 48, pp. 419-442, 1998.
http://dx.doi.org/10.1007/s001860050036

[12] Pilar Ibarrol, and Ricardo Velez, *“Multi armed bandit processes with optimal selection of the operation times,” Sociedad de Estadistica e Investigacion Operativa Test,* vol. 14, No. 1, pp. 239-255, 2005.

[13] Peter Auer, Paul Fischer, and Nicol`O Cesa Bianchi, *“Finite time analysis of the multi armed bandit problem,” Machine Learning,* vol. 47, pp. 235–256, Netherlands, 2002.

[14] Robert Kleinberg, Alexandru Niculescu Mizil, and Yogeshwer Sharma, *“Regret bounds for sleeping experts and bandits,” Machine Learning*, vol. 80, pp. 245–272, 2010.
http://dx.doi.org/10.1007/s10994-010-5178-7