International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-12, October 2019

Towards an Improved Strategy for Solving Multi-Armed Bandit Problem

Semiu A. Akanmu, Rakhen Garg, Abdul Rehman Gilal

***Abstract*: Multi-Armed Bandit (MAB) problem is one of the classical reinforcements learning problems that describe the friction between the agent's exploration and exploitation. This study explores metaheuristics as optimization strategies to support Epsilon greedy in achieving an improved reward maximization strategy in MAB. In view of this, Annealing Epsilon greedy is adapted and PSO Epsilon greedy strategy is newly introduced. These two metaheuristics-based MAB strategies are implemented with input parameters, such as number of slot machines, number of iterations, and epsilon values, to investigate the maximized rewards under different conditions. This study found that rewards maximized increase as the number of iterations increase, except in PSO Epsilon Greedy where there is a non-linear behavior. Our Annealing-Epsilon greedy strategy performed better than Epsilon Greedy when the number of slot machines is 10, but Epsilon greedy did better when the number of slot machines is 5. At the optimal value of Epsilon, which we found at 0.06, Annealing Epsilon greedy performed better than Epsilon greedy when the number of iterations is 1000. But at number of iterations ≥ 1000, Epsilon greedy performed better than Annealing Epsilon greedy. A stable reward maximization values are observed for Epsilon greedy strategy within Epsilon values 0.02 and 0.1, and a drastic decline at epsilon > 0.1.**

Keywords—Multi-armed bandit strategy, reinforcement learning, metaheuristics, epsilon greedy, annealing, particle swarm optimization

I. INTRODUCTION

Multi-Armed Bandit (MAB) problem is one of the classical reinforcements learning problems that describe the friction between the agent's exploration and exploitation [1]. Epsilon greedy, Thompson sampling, POKER strategy, and William Press's Clinical Trial are classes of MAB strategies that have been applied to varieties of optimization challenges in diverse domains [2], [3]. Epsilon greedy strategy defines the agent's exploration, ϵ, but selects the optimal arm with the probability of $1 - \epsilon$, which defines its exploitation [4]. Thompson sampling, introduced by Thompson [5], is a Bayesian approach of implementing MAB strategy to account for uncertain reward distributions. Thompson draws a random sample from an expected distribution parameter of θ and the probability of getting the optimal arm is on model parameters, ω.

Revised Manuscript Received on October 30, 2019.
***** Correspondence Author
Semiu A. Akanmu*, Department of Computer Science North Dakota State University Fargo, USA.E-mail: semiu.akanmu@ndsu.edu
Rakhen Garg Department of Computer Science,North Dakota State University, Fargo, USA. E-mail: rakhen.garg@ndsu.edu
Abdul Rehman Gilal Department of Computer Science Sukkur IBA University, Sindh, Pakistan.E-mail:a-rehman@iba-suk.edu.pk

POKER (Price of Knowledge and Estimated Reward) strategy, on the other hand, is adapted into MAB strategy from a popular card game that combines chance and strategy. It is an adversarial strategy like game theory [6]. The William Press's Clinical Trial is an application of MAB to healthcare with general emphasis on heuristics. It guides when health specialists could decide to forgo testing new drug and "exploit" the best treatment, in a way that mimics the explore-exploit friction of a reinforcement learning space [7].This study, towards an improved strategy in MAB problem for reward maximization, adapted Annealing algorithm, and introduced Particle Swarm Optimization (PSO) to form a hybrid strategy with Epsilon greedy. Experiments and comparative analysis based on the number of slot machines, the number of iterations, epsilon values and rewards maximized at different instances are conducted, and the findings discussed. In the next section, Annealing algorithm and PSO, as metaheuristics strategies, are described. The third section discusses past related works, and the fourth section discusses our work. The experiments and results, findings and discussion are followed. Lastly, the limitation of this study, suggestions for future works, and conclusion are presented.

II. METAHEURISTICS: ANNEALING ALGORITHM AND PARTICLE SWARM OPTIMIZATION

Simulated annealing and Particle Swarm Optimization (PSO) are metaheuristics strategies with high performance rate in solving optimization problems [8]. Simulated Annealing is a probabilistic technique for global optimum approximation of a function, especially in a large search space. It is inspired by annealing in metallurgy, a technique that involves heating and cooling of a material for size increase and reduction of defects [3], [9]. PSO, on the other hand, is a computational method that solves optimization problem by iterative improvement of the candidate solution using cognitive and social terms as measures of quality [10]. The candidate solution, called particles, is influenced by its local best position and updated based on the better positions found by other particles. A simple PSO formulation is given as,

$$V_g(t+1) = wv_g(t) + r_1\, c_1\big(P_g(t) - x_g(t)\big) + r_2\, c_2\left(g(t) - x_g(t)\right) \ldots (i)$$

$$x_g(t+1) = x_g(t) + v_g(t+1) \ldots (ii)$$

v and x are, respectively, the velocity and position of the particle in ith position toward the dimension of j in the search space at time t. p is the best experienced position by particle i and g is the best discovered position among all particles. w is the inertia coefficient, while r and c and coefficients of the particle social, $P_g(t) - x_g(t)$, and

cognitive, $\left(g(t) - x_g(t)\right)$, fitness respectively. For problems where finding an approximate global optimum is more important than finding a precise local optimum in a fixed amount of time, simulated annealing and PSO may be preferable to alternatives such as gradient descent. In general, both Annealing and PSO are optimization techniques which have been extended into solving MAB problem and its general applicability to solve industrial problems.

III. RELATED WORKS

Studies such as Liu, Downo, & Reid [2], Vigne [11], Manifar et al., [12] are some of the recent studies that investigated how MAB strategies can be improved. There are different conditions and constraints set in simulating the problem space and comparing the existing and new strategies. The general conclusion, however, is that, each strategy has its best, average and worst-case scenarios. This study used Vigne ([11]), Manifar et al., [12] as its foundational footing.

Vigne [11] highlighted the agent behavior of an explore-exploit scenario in a multi-armed bandit problem space using greedy epsilon strategy. The study emphasized the advantage of prioritization of the agent's greed for reward maximization in a reinforcement learning space. Manifar et al., on the other hand, worked on the improvement of average reward in a non-stationary multi-armed bandit using PSO. Even though Manifar et al.'s result only showed comparative advantage with *Softmax* and Epsilon greedy in certain instances of their experiments, it showed that PSO is promising in solving multi-armed bandit problem. Lastly, Manifar's(2018) work of Annealing Epsilon greedy algorithm for web optimization, evaluated by A/B testing, demonstrated how metaheuristics can be hybridized with Epsilon greedy strategy.

IV. OUR WORK

In view of improving the multi-armed bandit problem solving strategy, epsilon greedy and William Press's clinical trial are suggested as potentially best choices for this study based on the literature reviewed. The justifications are two: First, Epsilon greedy, being an early developed strategy, has enormous implementation resources and is argued to be one of the best reward maximization strategies [7]. Second, William Press's clinical trial strategy, which evolved from heuristics, is sparingly used in solving reinforcement learning problem. This gives an opportunity to explore metaheuristics as an optimization strategy in this study for novel insights. Our work, therefore, aimed at achieving an improved strategy through a hybridization of the two classes. Annealing and PSO are explored to develop a hybrid of each of the metaheuristic strategies with Epsilon greedy. Vigne [11], Manifare et al., [12], as earlier discussed, are the core scholarly works in this regard. It is against this backdrop that our work (a) adapted Annealing Greedy Epsilon strategy to fit into simulation of multi-armed bandit problem by defining epsilon, ϵ, using decaying epsilon time of $\epsilon = 1/log(time + 0.0000001)$, and (b) hybridize PSO and Greedy Epsilon by parameterizing PSO's number of particles and target error as number of slot machines and the maximum number of the Jackpot probability (JP) respectively. We chose *gbest-fitness-value*, $g(t)$, instead of best position, $P_g(t)$, of the PSO's agent as a return variable value of our PSO's implementation. We used this in the calculation of the epsilon value in the multi-armed bandit problem using $1/log(gbest\_fitness\_value)$. This is because multi-armed bandit's agent is better described by its fitness, measured by how much it has learnt about its space than its position. The next section describes how the experiment is conducted, with a stage-by-stage description of the respective findings.

V. EXPERIMENTS AND RESULTS

The performances of the multi-armed bandit strategies are evaluated by the values of rewards maximized in the explore-exploit scenarios. These are ascertained by running the respective implementation codes for each of the strategies, namely, Greedy Epsilon, Annealing-Greedy Epsilon, and PSO-Greedy Epsilon strategies. First, for the Greedy Epsilon, the input parameters are number of slot machines, epsilon values, and number of iterations. The epsilon is assigned a value of 0.1 being the mostly used value in previous related studies [2], [11]. The experiment varied the number of iterations and different instances of 5 and 10 slot machines. Table 1 presents the maximized rewards of the Greedy Epsilon strategy for 5 and 10 slot machines. Figure 1 presents a line chart depicting a comparison of the Epsilon Greedy strategy for the 5 and 10 slot machines instances.

TABLE I. Maximized rewards of the Greedy Epsilon strategy (5 and 10 slot machines)

Epsilon-Greedy (Epsilon value: 0.1)

		Iteration	200	400	600	800	1000	1200
No of slot machine	5	Rewards	1604	3175	4735	6174	7745	9360
	10		1032	2603	4152	5701	7261	8821

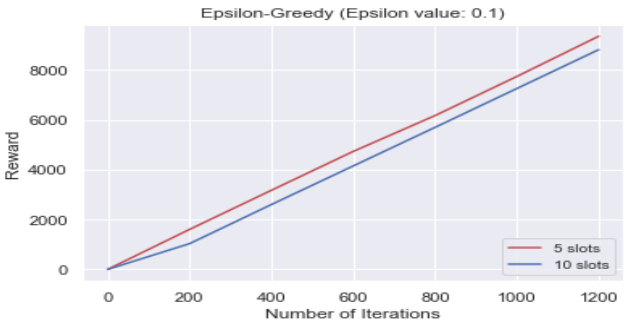

Fig.1. Epsilon Greedy strategy for the 5 and 10 slot machines instances

TABLE II. Maximized rewards of the Annealing Greedy-Epsilon Strategy (5 and 10 slot machines)

Annealing Epsilon-Greedy								
		Iteration	200	400	600	800	1000	1200
No of slot machine	5	Rewards	1494	2999	4493	5943	7503	9096
	10		1472	2955	4405	5877	7426	8975

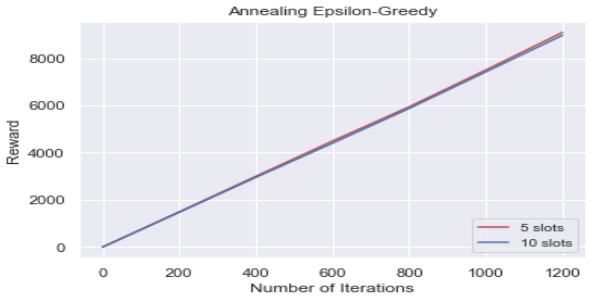

Fig.2. Annealing-Epsilon Greedy strategy for the 5 and 10 slot machines instances

Table III. Maximized rewards of the PSO Greedy-Epsilon Strategy (5 and 10 slot machines)

PSO Epsilon-Greedy								
		Iteration	200	400	600	800	1000	1200
No of slot machine	5	Rewards	1560	1349	2062	5767	7349	8766
	10		856	1613	2601	3336	4148	5851

Fig.3. PSO-Epsilon Greedy strategy for the 5 and 10 slot machines instances

Second, for the Annealing-Epsilon Greedy strategy, the number of slots and iterations are the only input parameters. The decaying time, which is used in calculating the epsilon value, is computed from the number of arms counts. Tables 2 presents the maximized rewards of the Annealing Epsilon-Greedy strategy for 5 and 10 slot machines. Figure 2 presents a line chart depicting a comparison of the Annealing Epsilon-Greedy strategy for the 5 and 10 slot machines instances.

Third, the PSO-Epsilon Greedy strategy, like Annealing-Epsilon greedy, has the number of slots and iterations has its input parameters. The PSO's *gbest* fitness value is used in computing its epsilon value. Table 3 presents the maximized rewards of the PSO Epsilon-Greedy strategy for 5 and 10 slot machines. Figure 3 presents a line chart depicting a comparison of the PSO Epsilon-Greedy strategy for the 5 and 10 slot machines instances.

Fourth, we investigated the optimal value of epsilon based on Liu, Downo, & Reid's [2] suggestion of 0.05 to 0.1 range. The maximized rewards for a range of epsilon values from 0.02 and 0.12 are investigated for 5 and 10 slot machines when the number of iterations is 1000. Table 4 presents the maximized rewards from the experiment. Figure 4 presents a line chart depicting the maximized rewards for the 5 and 10 slot machines instances across the range of the epsilon values.

TABLE IV. Maximized rewards of the Greedy-Epsilon Strategy (5 and 10 slot machines, Epsilon values 0.02 – 0.12)

Epsilon-Greedy (Number of iterations = 1000)								
		Epsilon values	0.02	0.04	0.06	0.08	0.10	0.12
No of slot machine	5	Rewards	8053	8031	8108	7921	7745	3499
	10		6986	7393	7525	7371	7261	4148

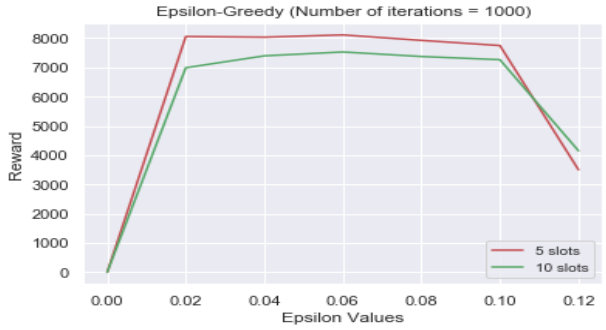

Fig.4. Maximized rewards of the Greedy-Epsilon Strategy (5 and 10 slot machines, Epsilon values 0.02 – 0.12)

TABLE V. Maximized rewards of the Annealing Epsilon greedy and Epsilon Greedy Strategies at Epsilon value = 0.06

Annealing Epsilon Greedy and Epsilon-Greedy (Number of Slot machine = 10)								
Number of iterations		200	400	600	800	1000	1200	1400
Rewards	Epsilon Greedy	1153	2724	4251	5844	7525	9118	10634
	Annealing Epsilon Greedy	1472	2955	4405	5877	7426	8975	10370

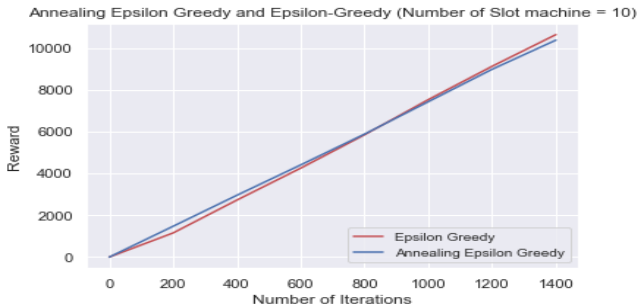

Fig.5. Maximized rewards of the Annealing Epsilon greedy and Epsilon Greedy Strategies at Epsilon Optimal Value

Fifth, for further insights, we run another experiment of Epsilon Greedy strategy at epsilon value 0.06, being its optimal value, and Annealing Epsilon greedy, with 10 as the number of slot machines over a series of iterations. Table 5 presents the maximized rewards for each of the iterations from the experiment.

Figure 5 presents a line chart depicting the maximized rewards for both Epsilon Greedy and Annealing Epsilon greedy when the epsilon value for Epsilon Greedy is placed at the optimal value.

The next section discusses the findings of these results and how they collectively contribute to the understanding of MAB for reinforcement learning space.

VI. FINDINGS AND DISCUSSION

The findings in this study are into three folds: (a) the relationship between increase in the number of iteration and the rewards maximized, (b) relationship between number of slot machines and the rewards maximized, and (c) the optimal value for Epsilon for reward maximization, and comparison between Epsilon Greedy and Annealing Epsilon-greedy strategies. This study found that rewards maximized by Epsilon Greedy and Annealing Epsilon-greedy strategies increase as the number of iterations increases. This is evident in the positive linear graphs depicted by Figures 1 and 2. The positive relationship between the increase in the rewards maximized and increase in the number of iterations is not significantly found in PSO Epsilon Greedy strategy because of the non-linear behavior of PSO, especially when the number of slot machines is 5.

It is also found that rewards are better maximized, that is, are of greater values, when the number of slot machine is 5 compared to when it is 10 for both Epsilon Greedy and Annealing Epsilon-greedy strategies. This is also partially true for the PSO Epsilon Greedy strategy. The difference is however not conspicuous in Annealing Epsilon greedy as it is in Epsilon Greedy. The PSO Greedy Epsilon revealed similar behavior except within iteration 200 to 600 where the strategy achieved an up-and-down reward output. It can, however, be asserted that the MAB strategies perform better with smaller number of slot machines which can be described as adversaries, which the agent is attempting to outsmart, within the reinforcement learning space. As shown in the results presented in Tables 1 and 2, Epsilon Greedy performed better than Annealing-Epsilon when no of slot machines is 5 while Annealing-Epsilon performed better than Epsilon Greedy when no of slot machines is 10. This supports the assertion that Annealing works better in a larger search space.

The optimal value of Epsilon, where the maximum rewards are achieved, is 0.06. A stable reward maximization values are observed for Epsilon greedy strategy within Epsilon values 0.02 and 0.1. The reward however nosedived at Epsilon value > 0.10, as shown in Figure 4. This supports the rationale for the choice of 0.1 as Epsilon value in all the past related studies. As an addendum, this study suggests that the optimal value is 0.06 based on the experimental results presented in Table 4 and Figure 4. The identification of the Epsilon optimal value prompted the comparison of Epsilon Greedy and Annealing Epsilon-greedy strategies. The number of slot machine is fixed at 10, with varying number of iterations. Expectedly, as presented in Table 5 and Figure 5, both strategies showed an increase in the maximized rewards as the number of iterations increases. It is however noteworthy that Annealing Epsilon-greedy performs better than Epsilon greedy when the number of iterations < 1000, but Epsilon-greedy performs better than Annealing Epsilon-greedy when the number of iterations ≥ 1000.

VII. LIMITATIONS FUTURE WORK AND CONCLUSION

This study exclusively assessed the MAB strategies based on reward maximization, without considering regret or penalty minimization. It is on this basis that only Epsilon-based MAB strategies are investigated. Epsilon favored exploitation over exploration for reward maximization, as against exploration over exploitation for regret minimization. Also, our proposed PSO Epsilon Greedy strategy, though showed similar behavior to other MAB strategies, did not perform better in terms of reward maximization.

Our suggested future works are: extension of MAB strategies assessment to include others like Thompson sampling and POKER which are regret minimization-focused and a comparative study that include both reward maximization and regret minimization assessment. Two, additional parameter and constraints specification for the proposed PSO Epsilon Greedy strategy for a better performance.

In conclusion, our adapted Annealing Epsilon-greedy strategy behaves consistently and showed better performance than Epsilon-greedy specifically when the arms (i.e. adversary) is not less than 10 and the exploitation bound (denoted by iteration) is less than 1000 for $\epsilon = 0.06$. This study identified $\epsilon = 0.06$ as the optimal value for Epsilon for reward maximization for MAB strategies.

REFERENCES

1. J. Langford and T. Zhang, "The Epoch-Greedy algorithm for contextual multi-armed bandits," in *Advances in Neural Information Processing Systems 20 - Proceedings of the 2007 Conference*, 2009.
2. L. Liu, R. Downe, and J. Reid, "Multi-Armed Bandit Strategies for Non-Stationary Reward Distributions and Delayed Feedback Processes," *arXiv Prepr. arXiv1902.08593*, 2019.
3. D. Henderson, S. H. Jacobson, and A. W. Johnson, "The Theory and Practice of Simulated Annealing," in *Handbook of Metaheuristics*, 2006.
4. M. Tokic and G. Palm, "Value-difference based exploration: Adaptive control between epsilon-greedy and softmax," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2011.
5. W. R. Thompson, "On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples," *Biometrika*, 1933.
6. J. Vermorel and M. Mohri, "Multi-armed bandit algorithms and empirical evaluation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2005.
7. R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
8. A. M. S. Asih, B. M. Sopha, and G. Kriptaniadewa, "Comparison study of metaheuristics: Empirical application of delivery problems," *Int. J. Eng. Bus. Manag.*, 2017.
9. A. R. Gilal, J. Jaafar, L. F. Capretz, M. Omar, S. Basri, and I. A. Aziz, "Finding an effective classification technique to develop a software team composition model," *J. Softw. Evol. Process*, vol. 30, no. 1, 2018.
10. S. Sengupta, S. Basak, and R. Peters, "Particle Swarm Optimization: A Survey of Historical and Recent Developments with Hybridization Perspectives," *Mach. Learn. Knowl. Extr.*, 2018.

11. F. Vigne, "Artificially Intelligent - A Closer Look at Reinforcement Learning," 2018. [Online]. Available: https://social.msdn.microsoft.com/Forums/en-US/4c76fe68-1078-422a-a5c2-5a0d26f1a5b0/artificially-intelligent-a-closer-look-at-reinforcement-learning?forum=msdnmagazine. [Accessed: 25-Apr-2019].
12. and M. M. Elnaz Manifar, Behrooz Masoumi, "Applying Particle Swarm Optimization to Improve Average Reward in non-Stationary Multi-Armed Bandit," 2018.

AUTHORS PROFILE

Semiu A. Akanmu is currently a software research engineer with Dickinson Research Extension center, North Dakota State University, US. He is also working on a doctoral research with interest in ontology modelling for software security. He had his Bachelor of Science (BSc) in computer science from Olabisi Onabanjo University, Ago-Iwoye, Ogun State, Nigeria, in 2008, Master of Science (MSc) and a Doctor of Philosophy (Ph.D.) in Information Technology (IT) from Universiti Utara Malaysia, Sintok, Malaysia, in 2013 and 2016, respectively.

Rakhen Garg is currently a graduate research assistant with High Performance Computing Center for Computationally Assisted Science and Technology, North Dakota State University, USA. He is also working on a master's research with interest in Data Science and Artificial Intelligence. He had his Bachelor of Engineering (BE) in Electronics & Communication from Chitkara University, Punjab, India, in 2015.

Abdul Rehman Gilal is a faculty member of Computer Science department at Sukkur IBA University, Pakistan. He has earned Doctor of Philosophy (Ph.D.) in Information Technology from Universiti Teknologi Petronas (UTP), Malaysia. He has been mainly researching in the field of software project management for finding the effective methods of composing software development teams. Based on his research publication track record, he has contributed in the areas of human factor in software development, complex networks, databases and data mining, programming and cloud computing.