Reinforcement Learning for Dynamic Optimization Problems

Abdennour Boulesnane
Faculty of Medicine, Salah Boubnider University
Constantine, Algeria
aboulesnane@univ-constantine3.dz

Souham Meshoul
Princess Nourah Bint Abdulrahman University RC-CCIS
Riyadh, Saudi Arabia
sbmeshoul@pnu.edu.sa

ABSTRACT

Many real-world applications require optimization in dynamic environments where the challenge is to find optima of a time-dependent objective function while tracking them over time. Many evolutionary approaches have been developed to solve Dynamic Optimization Problems (DOPs). However, there is still a need for more efficient methods. Recently, a new interesting trend in dealing with optimization in dynamic environments has emerged toward developing new Reinforcement Learning (RL) algorithms that are expected to give a new breath in DOPs community. In this paper, a new Q-learning RL algorithm is developed to deal with DOPs based on new defined states and actions that are mainly inspired from Evolutionary Dynamic Optimization (EDO) aiming appropriate exploitation of the strengths of both RL and EDO techniques to handle DOPs. The proposed RL model has been assessed using modified Moving Peaks Benchmark (mMPB) problem. Very competitive results have been obtained and good performance has been achieved compared with other dynamic optimization algorithm.

CCS CONCEPTS

• **Theory of computation → Reinforcement learning**; • **Computing methodologies → Continuous space search**.

KEYWORDS

Dynamic optimization problems, Reinforcement learning, Q-Learning, Moving peaks benchmark

ACM Reference Format:
Abdennour Boulesnane and Souham Meshoul. 2021. Reinforcement Learning for Dynamic Optimization Problems. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3449726.3459543

1 INTRODUCTION

Many real world problems require optimization over time because of the dynamic nature of the environments. Typical fields where such problems need to be solved include economics, engineering, communication systems, bioinformatics and machine learning [1], to name just a few. Time dependent optimization problems are most commonly known as Dynamic Optimization Problems (DOPs). Solving DOPs is not only a matter of locating global optima as in static optimization but of being able to track such optima in changing objective landscapes as well. DOPs have been defined in different ways. A unified description can be found in [3].

Recently, a new interesting trend in dealing with dynamic environments has emerged toward developing new algorithms that are able to effectively handle DOPs using Reinforcement Learning (RL) system [3]. Their proposal is mainly based on drawing a connection between Evolutionary Dynamic Optimization (EDO) and RL while trying to solve DOPs. In this paper and based on the new insight provided in [3], we propose a new Q-learning RL algorithm to solve dynamic optimization. The agent in our proposed model is trained to be able to deal with future dynamic environments by taking the appropriate action according to the optimal learning policy. Inspired from EDO experience of handling DOPs, each state represents a potential solution from the search space. Furthermore, several actions have been introduced in order to adapt the learner agent to the different states of dynamic environments. Besides, it is worth to mention that our proposed Q-learning model does not require any change detector to know the occurrence of future changes in the search space. Rather, it is the task of the learner agent to learn how to deal with different complex situations. The conducted experimental study is based on the modified moving peaks benchmark (mMPB) problem [2], where only single peak scenarios are considered, to evaluate the performance of the developed algorithm.

2 MOTIVATION AND PROPOSED APPROACH

The authors in [3], present a new interesting definition framework of DOPs and hence draw a connection between EDO and RL, allowing exploitation of their strengths to better solve DOPs. In this context and inspired by the new vision proposed in [3], we propose a new dynamic optimization algorithm based on RL to cope with DOPs. The proposed approach use the famous Q-learning algorithm to handle with dynamic environments. Unlike the work presented in [3], we have introduced new definitions of the elements that make up our proposed Q-learning algorithm including: states, actions, policy and reward.

-State: The state space is one of the major challenges in RL [5]. State spaces are usually very large when we deal with real-world problems or continuous search spaces. In our case, the state space S will represent the candidate solutions where the objective is to find the optimal state at each time a change occurs in the environment. Therefore, the size of the state space S ($|S|$) will be a parameter to study in the experimental settings.

-Action: In this part, we will describe the set of actions A used to train the agent to learn how to cope with DOPs. These actions are inspired from EA's handling of dynamic problems as follow:

GECCO '21 Companion, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8351-6/21/07…$15.00
https://doi.org/10.1145/3449726.3459543

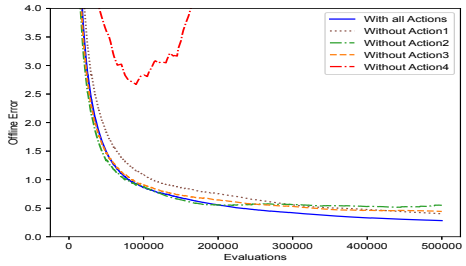

Figure 1: The averaged offline error over 30 runs obtained by our proposed model under different run conditions: with all actions/without some actions.

- *Action 1*: all candidate solutions (states) move towards the best solution found so far ($best_{state}$).
- *Action 2*: apply a local search near to the best solution found so far.
- *Action 3*: apply a local search near to the current state.
- *Action 4*: randomize all candidate solutions and reevaluate the best solution found so far.

-Policy: Given state s, an action a will be selected according to the famous ϵ-greedy policy function [5] used for exploration.

-Reward: The agent in our proposed algorithm is rewarded according to the spatial difference and the objective function values difference between the new state s' and the current state s.

The proposed algorithm works as follows, in the initialization phase, a Q-Table matrix of $|S|$x$|A|$ is initialized with generated zeros. After that, $|S|$ states in the search space are assigned values randomly for each dimension within the corresponding bounds, then the initial positions are evaluated based on the objective function. The algorithm proceeds iteratively as follow: firstly, the best state in the search space is updated. Then, the Q-learning is started for each state s. Based on the ϵ-greedy policy function, the agent selects an action a that will be used later to calculate the new state s' and the corresponding reward R. The number of states in the search space is constant where the new state s' position replaces the current state s. While the new next action a' is the best action index corresponding to the max Q value in the state s, which will be incorporated in the TD-target (Temporal Difference target). Through Bellman's equations, the agent try to minimize the gap between the expected future return of the TD-target and the $Q(s, a)$ value and hence building an action policy iteratively.

3 EXPERIMENTAL RESULTS

Next to our RL algorithm, it is worth to point out that in order to fairly compare the results, we have also created a new simple EDO algorithm based on Quantum Particle Swarm Optimization (QPSO) [4]. We adopt the acronym Dy-QPSO (Dynamic QPSO) to refer to this proposed algorithm.

From the experimental results and under different environmental conditions (Table 1), our proposed RL model shows a very competitive performance compared to EDO algorithms in handling DOPs. The proposed RL algorithm, in its course of learning, exploits actions with the high Q-values to receive the best reward, at the same time, it explores other actions in order to select the better ones in the future.

Table 1: Comparison of offline error (standard deviation) between Q-learning and Dy-QPSO algorithms on the mMPB problem with different change frequencies and different shift length.

Frequency (frq)			Severity (st)		
Values	Q-learning	Dy-QPSO	Values	Q-learning	Dy-QPSO
500	4.57(1.82)	**3.96(1.38)**	1	**0.46(0.18)**	0.84(0.15)
1000	2.31(0.97)	**2.02(0.61)**	2	**0.72(0.24)**	0.93(0.10)
2500	**0.99(0.39)**	1.38(0.24)	3	**0.88(0.29)**	1.08(0.17)
5000	**0.46(0.18)**	0.84(0.14)	5	**1.39(0.43)**	1.50(0.36)
10000	**0.25(0.10)**	0.44(0.05)	6	1.77(0.44)	**1.65(0.46)**

In order to show the importance of the proposed actions (*Action 1*,..., *Action 4*) used during the learning process, Figure 1 shows the effects of not using each of these actions on the model performance. Compared to the blue solid curve of the offline error labeled "With all Actions" in the same figure, the absence of *Action 1* prevents the algorithm to totally converge towards the optimal solution in each new environment. Whilst, the absence of *Action 2* or *Action 3* means the absence of local search which impacts negatively the quality of the global best solution and this appears clearly from the 30^{th} environmental change (i.e. after 200000 evaluations). However, the *Action 4* shows significant importance in the learning process and its absence means the efficiency degradation of the proposed model. This can be explained by the effectiveness of *Action 4* in dealing with the outdated memory problem resulting from the environmental changes.

4 CONCLUSIONS AND FUTURE WORK

In this paper, a new Q-learning RL algorithm is proposed to deal with optimization in dynamic environments. The elements that make up our proposed Q-learning algorithm including states, actions and reward function were defined in a new way inspired from EDO. For future work, it would be interesting to design more dynamic optimization algorithms based on RL models through a thorough exploration of the synergy between RL and EDO in handling DOPs.

REFERENCES

[1] Abdennour Boulesnane and Souham Meshoul. 2018. Effective Streaming Evolutionary Feature Selection Using Dynamic Optimization. In *Computational Intelligence and Its Applications*. Springer International Publishing, 329–340. https://doi.org/10.1007/978-3-319-89743-1_29

[2] Juergen Branke. 1999. Memory enhanced evolutionary algorithms for changing optimization problems. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*. IEEE, Washington, DC, 1882. https://doi.org/10.1109/cec.1999.785502

[3] Haobo Fu, Peter R. Lewis, Bernhard Sendhoff, Ke Tang, and Xin Yao. 2014. What are dynamic optimization problems?. In *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, Beijing, China, 1550–1557. https://doi.org/10.1109/cec.2014.6900316

[4] Jun Sun, Bin Feng, and Wenbo Xu. 2004. Particle swarm optimization with particles having quantum behavior. In *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*. IEEE, Portland, OR, USA, USA, 325–331. https://doi.org/10.1109/cec.2004.1330875

[5] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA.