

# Gradient Bias to solve the Generalization Limit of Genetic Algorithms through hybridization with Reinforcement Learning

Federico Esposito and Andrea Bonarini

AI and Robotics Lab  
Dipartimento di Elettronica, Informazione e Bioingegneria  
Politecnico di Milano  
Via Ponzio 34/5 - 20133 Milano MI - Italy  
<http://airlab.deib.polimi.it/>  
{federico.esposito, andrea.bonarini}@mail.polimi.it

**Abstract.** Genetic Algorithms have recently been successfully applied to the Machine Learning framework, being able to train autonomous agents and proving to be valid alternatives to state-of-the-art Reinforcement Learning techniques. Their attractiveness relies on the simplicity of their formulation and stability of their procedure, making them an appealing choice for Machine Learning applications where the complexity and instability of Deep Reinforcement Learning techniques is still an issue. However, despite their apparent potential, the classic formulation of Genetic Algorithms is unable to solve Machine Learning problems in the presence of high variance of the fitness function, which is common in realistic applications.

To the best of our knowledge, the presented research is the first study about this limit, introduced as *the Generalization Limit of Genetic Algorithms*, which causes the solutions that Genetic Algorithms return to be not robust and in general low-performing. A solution is proposed based on the *Gradient Bias* effect, which is obtained by artificially injecting more robust individuals into the genetic population, therefore biasing the evolutionary process towards this type of solutions. This Gradient Bias effect is obtained by hybridising the Generic Algorithm with the Deep Reinforcement Learning technique DDPG, resulting in the Explorative-DDPG algorithm (*X-DDPG*).

*X-DDPG* will be shown to solve the Generalization Limit of its genetic component via Gradient Bias, while outperforming its DDPG baseline in terms of agent return and speed of learning.

**Keywords:** Machine Learning · Genetic Algorithms · Deep Reinforcement Learning · DDPG · Gradient Bias · Distributed Exploration.

## 1 Introduction

In recent years, researchers have shown the ability of Genetic Algorithms (GA) to extend beyond rudimentary Machine Learning applications to the training of

autonomous agents, for a number of different tasks. Their general formulation makes it possible to address very complex problems [9, 4], even relating to the building blocks of intelligence [19, 2], with minimal intervention of the designers, leaving the process free to extract from raw data as much information as required. In the last years, GAs have been used to evolve agents that obtained performance comparable or even superior to those trained with Deep Reinforcement Learning algorithms in a number of benchmark environments [20, 17]. Indeed, the modern trend seems to be leaning to GAs to solve more and more complex tasks.

However, there is a structural limitation. We show that, if for fixed policies the tasks present stochasticity (in the policy and/or in the environment) which leads to trajectories that are subject to high variance, GAs are unable to evolve robust and highly-performing controllers. We call this problem the *Generalization Limit of Genetic Algorithms*. To the best of our knowledge, no previous study highlights these issues as a structural limitation of GAs that makes them incompatible with most realistic problems concerning autonomous agents.

The main issue resides in the fact that initialized controllers are strongly biased towards poorly performing and high variance individuals, and that such solutions are a basin of attraction for GAs, which prefer such type of controllers and evolve solutions with these undesired characteristics.

To solve this problem, we propose to hybridize the GA with a popular Reinforcement Learning algorithm, Deep Deterministic Policy Gradient (DDPG) [13]. Controllers obtained with Reinforcement Learning techniques are intrinsically more robust, since the learning objective itself is to increase the theoretically expected score over the environment stochasticity. By periodically injecting DDPG-trained agents into the genetic population, evolution is biased towards more robust solutions. We name this phenomenon the *Gradient Bias* effect. The resulting hybrid algorithm is called *Explorative DDPG (X-DDPG)*, combining a GA with a distributed variant of DDPG called AE-DDPG [23]. It will be shown that X-DDPG is able to successfully face the *Generalization Limit* of its genetic component, while also outperforming its RL baseline AE-DDPG.

Experiments are performed on RL benchmarks offered by the OpenAI Gym platform [5].

This paper is organized as follows. In *Section 2*, the work related to this research is illustrated. *Section 3* is dedicated to the *Generalization Limit of Genetic Algorithms*, introduced with experimental results and then discussed to provide a better understanding of the underlying issue. In *Section 4*, GAs are hybridized with AE-DDPG to create the X-DDPG algorithm, and the mutual benefits that each component receives from the other are discussed, focusing on the *Gradient Bias*.

## 2 Related Work

The central topic of this research are Genetic Algorithms, applied to the end-to-end training of autonomous agents, a field where they have gained increasing success in the last decades [1, 4]. In the field of Evolutionary Robotics [9],

Artificial Neural Networks are used as robot controllers and trained by Evolutionary Strategies. Notable are the works on *Minimal Cognition* [19], robot locomotion [3], and symbol grounding [2]. They have also proved to be valid alternatives to Reinforcement Learning techniques for some tasks [20, 17]. However, no previous study addresses the core topic presented in this paper, the *Generalization Limit of Genetic Algorithms*, the low performance of GAs in the presence of high variance of the fitness functions. Stochasticity in GAs has been widely addressed in literature [15, 22, 11], either by stating the general need for a reliable fitness estimation or the desire for a robust solution. Our contribution moves forward by stating that the selection mechanism of GAs is incompatible with high-variance fitness distributions, and that this limit is thus a structural problem of the algorithm.

To address the *Generalization Limit*, we propose to support the evolutionary process by *hybridization* with a RL algorithm: DDPG. The idea of hybridizing RL algorithms with GAs is not new, though in most cases these algorithms are built to have GAs support the RL process, either by influencing the agent network weights [18, 6] or the training data [16]. In [7, 12], the genetic component is used as a *distributed exploration mechanism*, where the transitions gathered when evaluating the population’s new individuals are used to train the Reinforcement Learning agent using DDPG. The present research builds on similar ideas. In particular, [12] also includes the technique of injecting the DDPG-trained agent into the genetic population. This is the core idea behind the *Gradient Bias* effect introduced in this paper.

We present a hybrid algorithm, *X-DDPG*, which uses as RL component a *distributed exploration* algorithm, *AE-DDPG* [23]. While in regular DDPG training experiences are obtained only from the learning agent, in *distributed exploration* algorithms many copies of the centralized agent contribute to these experiences. This idea gave birth to techniques such as A3C [14], Apex [10], Impala [8] and *AE-DDPG* [23] itself. Each method differs from the others in the way the information gathered by the distributed explorers is used in the training process.

### 3 The Generalization Limit of Genetic Algorithms

The perspective of being able to train neural policies with GAs, without the need of computationally expensive and unstable backpropagation, seems ideal. However, no previous study explicitly addresses the structural issue which we introduce. The aim of this Section is to support the validity of the following statement, which we propose as the core of our contribution:

*“Classical Genetic Algorithms cannot be employed effectively for agent training in the presence of high variance of the fitness function.”*

#### 3.1 Experiments

Individuals of the population are agents controlled by neural policies, whose architectures follow that of the original DDPG algorithm [13]. Experiments

are performed in two benchmark environments from the OpenAI-Gym library, *LunarLanderContinuous-v2* and *Swimmer-v2*<sup>1</sup>. The *scalar fitness* of each individual is the average score along 5 different episodes. Figure 1 shows the mean and best *scalar fitness* of the population during the evolutionary process.

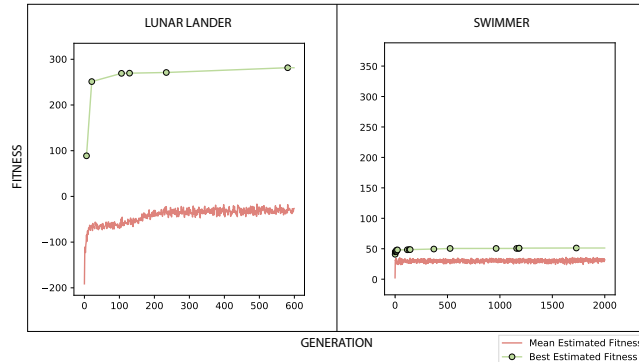


Fig. 1: Mean and best assigned scalar fitness during evolution

A complete evolutionary process was repeated in each environment 5 times, leading to similar early saturation of the fitness. The reason for this behavior may be caused by the very issue this research addresses.

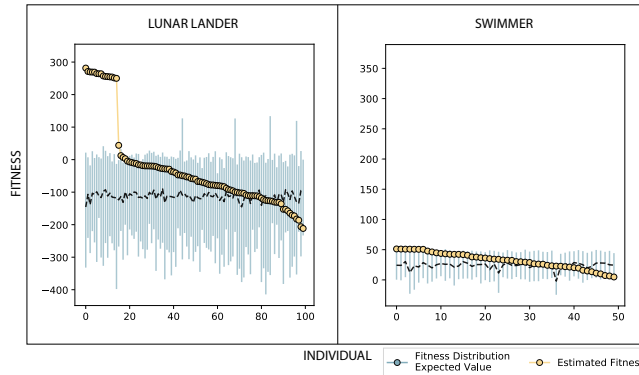
**Fitness Distribution** To monitor the expected value and variance of the true *fitness distributions* of individuals during evolution, they are estimated from data coming from 50 episode scores per-individual (Figure 2). Note that the *scalar fitness* used in the evolutionary process is the average of 5 episode-wise scores instead.

We estimate the fitness distribution mean and spread of two groups of individuals, respectively:

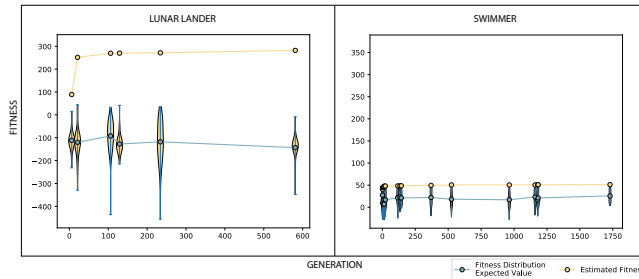
- *Last generation population*: each individual of the last generation population. Figure 2a.
- *Best individuals in evolution history*: whenever an individual emerges with *scalar fitness* above the current best population fitness, we compute its *true fitness distribution*. Figure 2b.

It is clear that the *scalar fitness* is a high overestimation of the true expected value of the fitness distribution. Moreover, the variance is actually never reduced as learning proceeds. These results suggest that the algorithm is not actually finding solutions with highest performance.

<sup>1</sup> The structure and parameters of the algorithm can be found at <https://github.com/AIRLab-POLIMI/GA-DRL>, in the GA section.



(a) Fitness distribution and scalar fitness of each individual of the last generation population



(b) Fitness distribution and scalar fitness of individuals with highest scalar fitness in history

Fig. 2: Estimated fitness distribution expected value and spread compared to assigned scalar fitness

### 3.2 Discussion

Multiple features of the classical Genetic Algorithm concur to generate the *Generalization Limit* in stochastic environments, and are illustrated in the following.

#### Evaluation

- Due to stochasticity, individual per-episode outcomes are not deterministic. Instead, each individual  $i$  has an associated *fitness distribution*  $f_i$ .
- The *scalar fitness*  $\hat{f}_i$  of each individual, required for the evolutionary process, is only computed once as the algebraic mean over a limited number  $n$  of episodes.
- Computing an individual’s *scalar fitness* from the outcome of  $n$  episodes is equivalent to sampling from the distribution of the algebraic mean, which is related to individual’s own *fitness distribution*  $f_i$  (same expected value, variance rescaled by  $\frac{1}{\sqrt{n}}$ ).

- As a result, *scalar fitnesses*  $\hat{f}_i$  of high-variance individuals may be *overestimated*, meaning that the sampled algebraic mean of the scores over the  $n$  episodes would be much higher than the true distribution’s expected value  $\mathbb{E}_{f_i}$ , due to the high variance of the distribution itself.
- The *scalar fitness* of robust/low-variance individuals may be lower than that of the overestimated individuals with high-variance.

### Selection

- Individuals are chosen for Crossover and Survival with a probability that increases with their *scalar fitness*  $\hat{f}_i$ .
- Selected individuals will bias the propagation of genetic information, and therefore solution type, throughout evolution.
- The classical Genetic Algorithm uses only the individuals’ *scalar fitnesses* for selection. There is no way to account for the true *fitness distribution*.
- Overestimated individuals present high *scalar fitness* but also high-variance, low-robustness *fitness distributions*. These overestimated individuals are therefore selected with high probability, but will transmit to future generations the actual low-performance genetic information of their distributions.
- *overestimated* individuals with high-variance distributions may be selected more frequently than *robust* individuals.

### Initialized Population

- Only a portion of high-variance individuals will be *overestimated*.
- Initialized populations are usually mostly composed of high-variance individuals, providing the necessary initial bias for *overestimated* individuals to overrun the robust ones.

The problem is that Genetic Algorithms try to find solutions with the highest possible *scalar fitness*. When individuals are instead characterized by *fitness distributions*, individuals with high-variance are more likely to get higher *scalar fitness* than robust individuals do, because the latter, characterized by low variance, would have to also present high expected values to sample high values. Moreover, the initial population is mostly composed of individuals prone to *overestimation*, since random solutions are more likely to have high variance than high performance. As a result, high-variance solutions are a basin of attraction for classical GAs: the genetic search is biased towards these higher variance solutions, where the sampled fitness highly overestimates the true expected value.

## 4 X-DDPG

To confront the *Generalization Limit*, we propose a hybrid solution that combines the GA with an RL-based training process, with the aim of biasing the genetic search toward more robust solutions. We call this algorithm *Explorative-DDPG*.

## 4.1 Motivations

**Addressing the Limits of Genetic Algorithms** When the amount of robust individuals is very low, the proportion of high-variance solutions will increase and eventually overrun the entire population. The problem could be solved by artificially injecting robust individuals in it. The problem is how to find a *source of robust individuals* without having to estimate their fitness distribution, which may in general be a very computationally expensive procedure.

Reinforcement Learning techniques find policies with the highest possible Q-value, which is the theoretical expected return, and, as a consequence, RL solutions are intrinsically robust.

We propose to support the genetic evolution with a parallel Reinforcement Learning process, used as a continuous source of more robust policies. Since RL agents are trained with backpropagation, we called this the *Gradient Bias* effect.

**Addressing the Limits of Reinforcement Learning** GAs could also in turn support the RL process.

In most DRL algorithms, the Experience Replay is filled by experiences gathered only by the learning agent itself. This poor sample diversity may lead to early convergence, a problem addressed in literature with *distributed exploration* strategies. GAs could contribute to the distributed exploration, by inserting in the *Experience Replay* trajectories sampled by the genetic individuals when evaluated. This would provide a more diversified set of experiences available to the RL training.

These are the motivations that lead to the proposal of the novel algorithm *X-DDPG*.

## 4.2 The Algorithm

In *X-DDPG*, a distributed DDPG training and a Genetic evolution are executed in parallel. The two processes are mutually interacting.

- **Improve DDPG – Maximal Exploration** : The exploratory processes are now two. The DDPG agent learns from transitions stored in its Experience Replay, which is composed of two separate memories: one for the distributed AE-DDPG exploration, and the other for the Genetic Experiences.
- **Improve the GA – Gradient Bias** : At the beginning of each generation, a copy of the most recent DDPG agent is inserted in the population, and is subject to Evaluation and Selection. This mechanism *biases* the genetic search towards regions of more robust solutions, which the GA can traverse more efficiently than DDPG.

### 4.3 Experiments and Results

Experiments are performed on the two benchmark environments mentioned in Section 3. The structure of the neural policies and the DDPG training are the ones prescribed by the original DDPG paper [13]<sup>2</sup>.

**X-DDPG And The Generalization Limit Of Genetic Algorithms** We compare the *fitness distributions* of the best individuals in the evolution history (Figure 3b) with those of the individuals from the last generation (Figure 3a), for the two cases of regular GA and X-DDPG. Fitness distributions are estimated from 50 episode scores per-individual, with respect to the *scalar fitnesses*, which are only composed of 5.

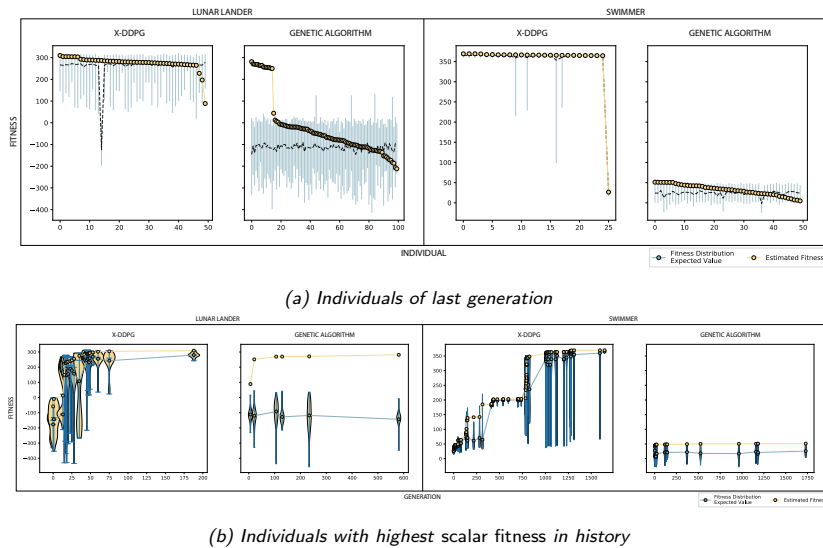


Fig. 3: Estimated fitness distributions of individuals of last generation (3a) and individuals with highest scalar fitness (3b), computed over 50 episodes per-individual, with and without Gradient Bias

The best individuals throughout the evolution still show variability in their fitness distributions. However, their expected values are close to the maximum, around which most of the spread is concentrated, and thus, solutions are quite robust. Moreover, the *scalar fitness* is in general *not an overestimation* of the true expected value of the distribution. This is a very important property for a GA since, in general, fitness distributions are not computed during training,

<sup>2</sup> Further details on the algorithm configuration and parameters can be found at <https://github.com/AIRLab-POLIMI/GA-DRL>, in the X-DDPG section

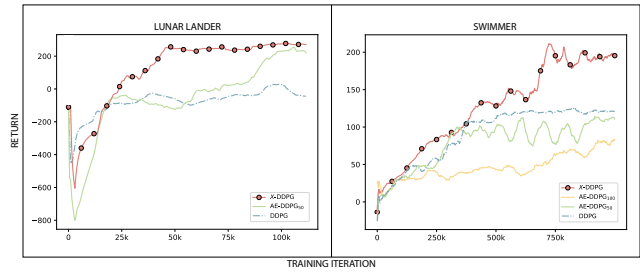


and designers must rely solely on the *scalar fitness* to monitor the evolutionary process and quality of the solutions.

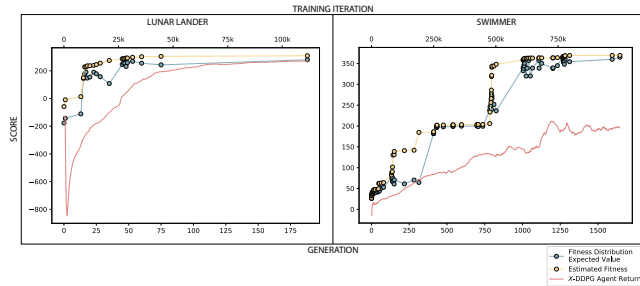
The sole *Gradient Bias* is able to make the evolutionary process not only feasible and reliable, but also capable of returning solutions with a quality comparable with the state of the art.

**X-DDPG Outperforms RL Baselines** The algorithm uses as RL component the distributed AE-DDPG algorithm [23].

The performance of the X-DDPG gradient training is now compared to that of the pure AE-DDPG and regular DDPG. Results are reported in Figure 4a, showing that the RL agent in X-DDPG outperforms agents trained by simple DDPG and AE-DDPG.



(a) Comparing performance of agents trained by X-DDPG and its baselines AE-DDPG and DDPG



(b) Comparison of the fitness of the best individuals in the history of the genetic search, showing both the scalar fitness and the estimated distribution, with the performance of the DDPG agent trained with X-DDPG

Fig. 4: Monitoring training agent performance

The final results are a comparison of the performance of the solutions generated by the two components of X-DDPG, independently from each other. Figure 4b shows the performance of both its RL agent and the best individuals generated by its genetic evolution. The best individuals of the genetic search actually outperform the RL agents from the very first generations, reaching optimal regions. As a consequence, the overall algorithm obtains very high-performance

solutions in a time even shorter than what required by the RL process, thanks to the GA, which is supported by the *Gradient Bias* provided by DDPG.

This is a very interesting result, as it shows that indeed GAs are very powerful tools to train autonomous agents, and that the *Generalization Limit* can be overcome to unleash their full potential. It is clear that genetic individuals do not only exploit the *Gradient Bias* in narrow regions surrounding the RL agent. Instead, this bias is just an initial tool, used to overcome the first barriers of high-variance solutions.

## 5 Conclusions

In this paper, a structural issue was presented, which makes Genetic Algorithms ineffective when the problem induces high variance of the fitness distributions. We called this the *Generalization Limit of Genetic Algorithms*, as the obtained controllers are unable to generalize their working behavior, resulting, in general, in this situation, in poor performance and unreliable learning.

It was shown that the problem arises from the interaction of GAs with the stochasticity of the trajectories. The latter leads to high variance *Fitness Distributions* of the controllers, while the first is prone to overestimation of the expected performance. When the variance is high, the overestimated fitness of higher variance individuals surpass that of the more robust, lower variance, individuals, and the overall population is biased towards the preservation of these high-variance, unreliable controllers.

In Section 4, a solution to this problem is presented, called the *Gradient Bias*. Indeed, by periodically injecting in the population more robust individuals, a threshold is reached where overestimated controllers cannot overrun the robust ones anymore, and are instead progressively discarded.

A reliable source of robust individuals comes from a parallel Reinforcement Learning algorithm, *AE-DDPG*. The two processes are mutually interacting, resulting in the hybrid algorithm *Explorative-DDPG*. *X-DDPG* was shown to be able to solve the *Generalization Limit* of the Genetic search, while also outperforming agents trained with pure *AE-DDPG*.

Genetic Algorithms are actually able to exploit regions with higher fitness more efficiently than DDPG, but only once such regions are approached by the population. This need of the genetic search to be biased outside lower performance regions may be the reason that led in the past to underestimate the potential of Genetic Algorithms.

### 5.1 Limits and Future Directions

**Generality of the statements** The presented results have been shown for a limited number of tasks. Our current research aim is to widen the range of environments to show the generality of both the issue and the solution that we presented. This will also include different incarnations of the Genetic Algorithm formulation.

**A pure Genetic Solution.** Though promising, the *Gradient Bias* effect can only support the Genetic Algorithm when the GA is used to solve a problem formulated in an RL framework. Though it is argued by Sutton and Barto [21] that any learning problem can be formulated as such, it may still be an issue to always model a problem in this fashion. The ideal condition would be to solve the *Generalization Limit* with a variation of the pure GA. It has been shown in this paper that, once appropriately supported, GAs are more than capable to solve even complex problems, outperforming the more commonly used RL algorithms.

**The Power of Hybridization**  $X$ -DDPG has been introduced in this paper mainly as a solution to the *Generalization Limit* of its genetic component, with the introduction of the *Gradient Bias* effect. However, in the environments under study,  $X$ -DDPG obtained state of the art results in terms of training iterations and final performance, by only affecting the training experiences. Future studies will focus on different types of DRL algorithms, applied to a larger and more diversified set of environments.

## References

- [1] J. Bongard. “Morphological Change in Machines Accelerates the Evolution of Robust Behavior”. In: *Proceedings of the National Academy of Sciences* 108.4 (2011), pp. 1234–1239. ISSN: 0027-8424. DOI: 10.1073/pnas.1015390108. URL: <https://www.pnas.org/content/108/4/1234>.
- [2] J. Bongard and J. Anetsberger. “Robots Can Ground Crowd-Proposed Symbols By Forming Theories Of Group Mind”. In: *The 2019 Conference on Artificial Life* 28 (2016), pp. 684–691. DOI: 10.1162/978-0-262-33936-0-ch109.
- [3] J. Bongard and R. Pfeifer. “A Method for Isolating Morphological Effects on Evolved Behaviour”. In: (July 2003).
- [4] J. Bongard, V. Zykov, and H. Lipson. “Resilient Machines Through Continuous Self-Modeling”. In: *Science* 314.5802 (2006), pp. 1118–1121. ISSN: 0036-8075. DOI: 10.1126/science.1133687. URL: <https://science.sciencemag.org/content/314/5802/1118>.
- [5] G. Brockman et al. “OpenAI Gym”. In: *CoRR* abs/1606.01540 (2016). arXiv: 1606.01540. URL: <http://arxiv.org/abs/1606.01540>.
- [6] S. Chang et al. “Genetic-Gated Networks for Deep Reinforcement Learning”. In: Dec. 2018.
- [7] C. Colas, O. Sigaud, and P.-Y. Oudeyer. “GEP-PG: Decoupling Exploration and Exploitation in Deep Reinforcement Learning Algorithms”. In: *CoRR* abs/1802.05054 (2018). arXiv: 1802.05054. URL: <http://arxiv.org/abs/1802.05054>.
- [8] L. Espeholt et al. “IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures”. In: *CoRR* abs/1802.01561 (2018). arXiv: 1802.01561. URL: <http://arxiv.org/abs/1802.01561>.

- [9] I. Harvey, P. Husbands, and D. Cliff. *Issues in Evolutionary Robotics*. 1992.
- [10] D. Horgan et al. “Distributed Prioritized Experience Replay”. In: *CoRR* abs/1803.00933 (2018). arXiv: 1803.00933. URL: <http://arxiv.org/abs/1803.00933>.
- [11] Y. Jin and J. Branke. “Evolutionary optimization in uncertain environments—a survey”. In: *IEEE Transactions on Evolutionary Computation* 9.3 (2005), pp. 303–317.
- [12] S. Khadka and K. Tumer. “Evolution-Guided Policy Gradient in Reinforcement Learning”. In: (Nov. 2019).
- [13] T. P. Lillicrap et al. *Continuous Control With Deep Reinforcement Learning*. 2015. arXiv: 1509.02971 [cs.LG].
- [14] V. Mnih et al. “Asynchronous Methods for Deep Reinforcement Learning”. In: *CoRR* abs/1602.01783 (2016). arXiv: 1602.01783. URL: <http://arxiv.org/abs/1602.01783>.
- [15] P. Rakshit, A. Konar, and S. Das. “Noisy Evolutionary Optimization Algorithms—A Comprehensive Survey”. In: *Swarm and Evolutionary Computation* (Nov. 2016). DOI: 10.1016/j.swevo.2016.09.002.
- [16] M. Ramicic and A. Bonarini. “Selective Perception As a Mechanism To Adapt Agents To The Environment: An Evolutionary Approach”. In: *IEEE Transactions on Cognitive and Developmental Systems* (2019), pp. 1–1. ISSN: 2379-8939. DOI: 10.1109/TCDS.2019.2896306.
- [17] T. Salimans et al. *Evolution Strategies as a Scalable Alternative to Reinforcement Learning*. 2017. arXiv: 1703.03864 [stat.ML].
- [18] A. Sehgal et al. “Deep Reinforcement Learning Using Genetic Algorithm for Parameter Optimization”. In: *2019 Third IEEE International Conference on Robotic Computing (IRC)*. Feb. 2019, pp. 596–601.
- [19] A. C. Slocum et al. “Further Experiments in the Evolution of Minimally Cognitive Behavior: From Perceiving Affordances to Selective Attention”. In: *In*. MIT Press, 2000, pp. 430–439.
- [20] F. P. Such et al. “Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning”. In: *CoRR* abs/1712.06567 (2017). arXiv: 1712.06567. URL: <http://arxiv.org/abs/1712.06567>.
- [21] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. Second. The MIT Press, 2018. URL: <http://incompleteideas.net/book/the-book-2nd.html>.
- [22] S. Yang, Y. Ong, and Y. Jin. *Evolutionary Computation in Dynamic and Uncertain Environments*. Vol. 51. Mar. 2007. ISBN: 978-3-540-49772-1. DOI: 10.1007/978-3-540-49774-5.
- [23] Z. Zhang et al. “Asynchronous Episodic Deep Deterministic Policy Gradient: Towards Continuous Control in Computationally Complex Environments”. In: *CoRR* abs/1903.00827 (2019). arXiv: 1903.00827. URL: <http://arxiv.org/abs/1903.00827>.