



UNIVERSITÉ DE FRIBOURG  
UNIVERSITÄT FREIBURG



eXascale Infolab

UNIVERSITY OF FRIBOURG

MASTER THESIS

---

**Thesis Title**

---

*Author:*  
Corina Masanti

*Supervisor:*  
Prof. Dr. Philippe  
Cudré-Mauroux

*Co-Supervisor:*  
Dr. Giuseppe Cuccu

January 01, 1970

eXascale Infolab  
Department of Informatics



# Abstract

Corina Masanti

*Thesis Title*

Black-Box Optimization (BBO) has been used in many areas, including Reinforcement Learning (RL). Unlike Deep Learning approaches, BBO techniques do not require additional information about the problem structure. Calculating an accurate gradient in RL problems is usually non-trivial but required for the usage of Deep Learning. In BBO this step is obsolete, which gives us more flexibility overall and more efficiency in the evaluation process. In addition, in BBO it is easy to distribute the computation for parallel computing. With the flexibility BBO offers us, we can choose an arbitrary model. In this thesis, I looked into the Fourier series, Bézier curve, and Taylor approximations.

Problem -> Solution -> Results

**Keywords:** Black-Box Optimization, Reinforcement Learning, Fourier



# Contents

<b>Abstract</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Black-Box Optimization . . . . .	1
1.1.1 Evolution Strategies . . . . .	1
1.1.2 Covariance Matrix Adaptation Evolution Strategy . . . . .	1
1.2 Fourier . . . . .	1
1.3 Bézier . . . . .	1
1.4 Reinforcement Learning . . . . .	1
1.4.1 Benchmarks in Reinforcement Learning . . . . .	1
1.5 Previous Work . . . . .	2
<b>2 Experiments</b>	<b>5</b>
2.1 Experiments . . . . .	5
2.2 Results . . . . .	5
<b>3 Conclusion</b>	<b>7</b>
3.1 Conclusion . . . . .	7
3.2 Future Work . . . . .	7
<b>Bibliography</b>	<b>9</b>



# List of Figures

1.1	Reproduced Plots . . . . .	3
-----	----------------------------	---





# Chapter 1

## Introduction

Your introduction chapter here.

### 1.1 Black-Box Optimization

#### 1.1.1 Evolution Strategies

Evolution Strategies (ES) ...

#### 1.1.2 Covariance Matrix Adaptation Evolution Strategy

Covariance Matrix Adaptation Evolution Strategy (CMA-ES) ...

### 1.2 Fourier

### 1.3 Bézier

### 1.4 Reinforcement Learning

Reinforcement Learning (RL) ...

#### 1.4.1 Benchmarks in Reinforcement Learning

When developing a novel algorithm, we have to discover how it performs compared to existing methods. For this evaluation, we need standard benchmark problems. OpenAI Gym<sup>1</sup> is a toolkit created for exactly this scenario. It contains a collection of benchmark problems with various levels of difficulty. However, not all benchmark problems are meaningful for the evaluation of an algorithm. If a problem is trivial to solve, the results do not reflect the quality of the model adequately.

In the paper *Analyzing Reinforcement Learning Benchmarks with Random Weight Guessing* (Oller, Glasmachers, and Cuccu (2020)), the authors analyze and visualize the complexity of standard RL benchmarks based on score distribution. They tested their approach on the five Classic Control benchmarks from the OpenAI Gym interface: CartPole, Acrobot, Pendulum, MountainCar, and MountainCarContinuous. Given an RL environment, the authors conducted a fixed series of experiments. For these experiments, they used three NN architectures ( $N_{architectures} = 3$ ): a network without hidden layers (0 HL), a network with a single hidden layer of 4 units (1 HL, 4 HU), and a network with two hidden layers of 4 units each (2 HL, 4 HU). To avoid bias in the data, they did not include any learning. Instead, they chose the

---

<sup>1</sup>[gym.openai.com](http://gym.openai.com)

network weights i.i.d. from the standard normal distribution  $\mathcal{N}(0, 1)$  with Random Weight Guessing (RWG). With this, they initialized  $10^4$  samples ( $N_{samples} = 10^4$ ) with different random weights. Each of these samples represents a controller that maps observations to actions in the environment. The controllers were tested on an environment for 20 independent episodes ( $N_{episodes} = 20$ ). For each episode, they saved the score in the score tensor  $S$ . Algorithm 1 illustrates the procedure with pseudocode.

---

**Algorithm 1** Evaluation process taken from Oller, Glasmachers, and Cuccu (2020)

---

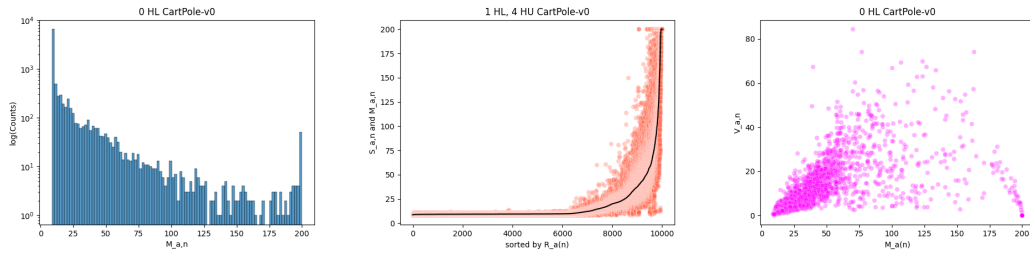
- 1: Initialize environment
  - 2: Create array  $S$  of size  $N_{architectures} \times N_{samples} \times N_{episodes}$
  - 3: **for**  $n = 1, 2, \dots, N_{samples}$  **do**
  - 4:     Sample NN weights randomly from  $\mathcal{N}(0, 1)$
  - 5:     **for**  $e = 1, 2, \dots, N_{episodes}$  **do**
  - 6:         Reset the environment
  - 7:         Run episode with NN
  - 8:         Store accrued episode reward in  $S_{a,n,e}$
- 

After the authors obtained the scores, they calculated the mean performance over all episodes from a sample and its variance. The samples were ranked according to their mean score. They then visualized their results with three plots: a log-scale histogram of the mean scores, a scatter plot of the sample scores over their rank, a scatter plot of score variance over the mean score. I reproduced their results following the mentioned methodology. My findings for the environment CartPole are displayed in Figure 1.1.

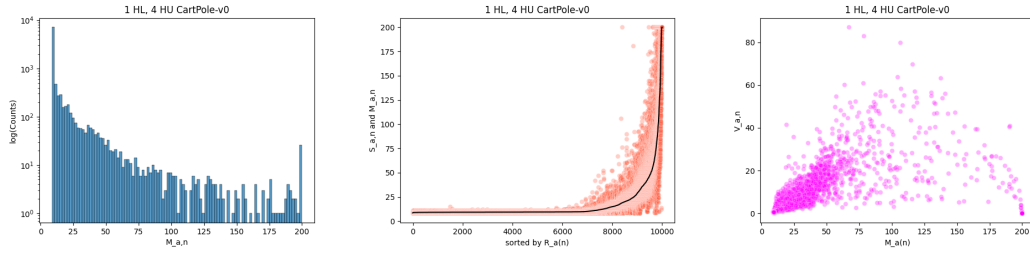
Describe method -> results -> conclusion

Describe plots, maybe add plots from other environments? now not really meaningful to show all architectures (not enough difference)

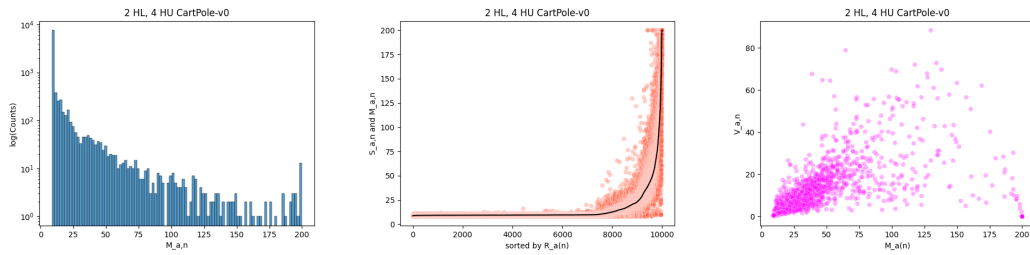
## 1.5 Previous Work



(A) Results of network architecture without hidden layers



(B) Results of network architecture with one hidden layer



(C) Results of network architecture with two hidden layers

FIGURE 1.1: **Plots.** Histogram plot, Scatter plots.



## Chapter 2

# Experiments

### 2.1 Experiments

### 2.2 Results



## Chapter 3

# Conclusion

### 3.1 Conclusion

In this work we...

### 3.2 Future Work

The continuation of this work includes...





# Bibliography

Oller, Declan, Tobias Glasmachers, and Giuseppe Cuccu (Apr. 16, 2020). “Analyzing Reinforcement Learning Benchmarks with Random Weight Guessing”. In: *arXiv:2004.07707 [cs, stat]*. arXiv: 2004.07707. URL: <http://arxiv.org/abs/2004.07707> (visited on 12/07/2021).