

# Parallel Genetic Classifier

Paulo Santos<sup>1</sup> and Maria Fidalgo<sup>1</sup>

Departamento de Informática da Faculdade de Ciências da Universidade de Lisboa  
{fcXXXX,fc49034}@alunos.fc.ul.pt

**Abstract.** The first sentence of an abstract should clearly introduce the topic of the paper so that readers can relate it to other work they are familiar with. However, an analysis of abstracts across a range of fields show that few follow this advice, nor do they take the opportunity to summarize previous work in their second sentence. A central issue is the lack of structure in standard advice on abstract writing, so most authors don't realize the third sentence should point out the deficiencies of this existing research. To solve this problem, we describe a technique that structures the entire abstract around a set of six sentences, each of which has a specific role, so that by the end of the first four sentences you have introduced the idea fully. This structure then allows you to use the fifth sentence to elaborate a little on the research, explain how it works, and talk about the various ways that you have applied it, for example to teach generations of new graduate students how to write clearly. This technique is helpful because it clarifies your thinking and leads to a final sentence that summarizes why your research matters.

**Keywords:** First keyword · Second keyword · Another keyword.

## 1 Introduction

The introduction should set the context for your project. Why is this topic relevant?

You should also define the scope of your project. You could design a software artifact that would end poverty and famine, but that is not realistic.

For example, this document describes the structure your paper should have. Despite using the LNCS LaTeX template <sup>1</sup>, the formatting template is not relevant, only the content structure is relevant.

Finally, you should define the goals of your project. For instance,

- To propose a method for the parallelization of Genetic Algorithms
- An implementation of such algorithm
- The experimental evaluation of such method, with comparison with a sequential alternative.

---

<sup>1</sup> LNCS is the official template for Europar 2019, in case you are interested.

## 2 Background

This is an optional section, as it depends on your project. In projects where a given specific knowledge is required to understand the article, you should give a brief introduction of the required concepts. In the case of genetic algorithms, you should present the basic algorithm in this section.

## 3 Approach

In this section, you should present your approach. Notice that an approach may be different than an implementation. An approach should be generic and ideally applied for different machines, virtual machines or languages. You should present algorithms or generic diagrams explaining the approach.

## 4 Implementation Details

In this section you can talk about details of how you implemented your approach. Depending on the detail of your approach, this might be an optional section.

As an example, I would detail how I implemented the phaser in the genetic algorithm, or how I implemented parallel mergesort on ForkJoin. Another aspect could be how to organize your arrays to minimize overhead in recursive calls.

## 5 Evaluation

### 5.1 Experimental Setup

In this section you should describe the machine(s) in which you are going to evaluate your system. Select the information that is relevant.

### 5.2 Results

In this section you should present the results. Do not forget to explain where the data came from.

You should include (ideally vectorial) plots, with a descriptive caption. Make sure all the plots (Like Figure ?? are well identified and axis and metrics are defined.

### 5.3 Discussion

Here you should discuss the results on a high level. For instance, based on our results, the parallelization of the merge-sort is relevant as no other parallel work occurs at the same time, and the complexity  $O(N\log(N))$  can have a large impact when the number of individuals is high.

## 6 Related Work

Several implementations of the genetic algorithm were made throughout the years. We will shortly talk about two implementations and how they fit in the scope of our work.

Dominic and Willis [1] developed a MATLAB toolbox, GPTIPS, which is able to perform regression through genetic programming. The main difference between their approach and ours is that they do not explore the parallelism of the algorithm, focusing on the usability of the toolbox. Moreover, they chose to include nonlinear operators, that we decided to leave out. Our work is, therefore, important to whom intends to develop a fast approach of the genetic classifier. Additionally, GPTIPS requires the purchase of a payed software (MATLAB), available to a less broader population.

Jenetics [2] is another genetic programming implementation. It is a Java library designed to abstract different concepts within the genetic programming panorama, such as Gene, Genotype and Chromosome, allowing it to serve a vast spectrum of domains. This library implements the Java Stream Interface and provides ForkJoin Parallelization. This is, therefore, a generic purpose implementation for genetic algorithm. On the other hand, in our work we provided a study of genetic programming specific for regression, where other parallelization techniques were able to achieve better results than ForkJoin, like the Island Models.

## 7 Conclusions

Here you should resume the major conclusions taken from discussion. Ideally, these should align with the objectives introduced in the introduction.

You should also list the future work, i. e., tasks and challenges that were outside your scope, but are relevant.

## Acknowledgements

First Author wrote the part of the program implemented the phasers. Second Author implemented the MergeSort in parallel.

Both authors wrote this paper, with First Author focusing on the introduction, related work and conclusions while the Second Author focused on approach and evaluation.

Each author spent around 30 hours on this project.

## References

1. Dominic, P., Leahy, D., Willis, M.: Gptips:an open source genetic programming toolbox for multigene symbolic regression. *Lecture Notes in Engineering and Computer Science* **2180** (12 2010)
2. Wilhelmstötter, F.: Jenetics: Library user's manual 4.3 (2018), <https://books.google.pt/books?id=6Y-5BQAAQBAJ>