Master's Degree in Informatics Engineering

Dissertation

Genetic Programming Algorithms for Dynamic Environments

July, 2015

João Pedro Gonçalves Teixeira de Macedo
jmacedo@student.dei.uc.pt

Advisor

Prof. Dr. Ernesto Costa

FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

University of Coimbra
Department of Informatic Engineering

Master's Degree in Informatics Engineering
Dissertation
July, 2015

Genetic Programming Algorithms for Dynamic Environments

Author
João Pedro Gonçalves Teixeira de Macedo
University of Coimbra
Department of Informatic Engineering
jmacedo@student.dei.uc.pt

Supervisor
Prof. Dr. Ernesto Jorge Fernandes Costa
University of Coimbra
Department of Informatic Engineering
ernesto@dei.uc.pt

Jury
Prof. Dr. Luís Paquete
University of Coimbra
Department of Informatic Engineering
paquete@dei.uc.pt

Prof. Dr. Vasco Pereira
University of Coimbra
Department of Informatic Engineering
vasco@dei.uc.pt

Abstract

Evolutionary Algorithms (EA) are a family of search heuristics from the area of Artificial Intelligence. They have been successfully applied in problems of learning, optimization and design, from many application domains. Currently, they are divided into two families, Genetic Algorithms (GA) and Genetic Programming (GP). Genetic Algorithms evolve solutions for a specific problem. On the other hand, Genetic Programming evolves programs that, when executed, produce the solutions for specific problems.

Many of the successful applications of EAs have been on static environments, i.e., environments whose conditions remain constant throughout time. However, many real world applications involve dynamic environments, meaning that the problems themselves change over time.

The difficulty of evolving solutions in dynamic environments emerges from a common problem of EAs known as premature convergence. This phenomenon happens when the population converges to a good quality area of the search space, being the individuals very similar to each other. In static environments, this may cause the algorithm to only find local optima instead of the global optimum solution. On the other hand, in dynamic environments, this phenomenon may cause a greater difficulty and delay in finding good solutions when the environment changes, specially if the new environment is very different from the previous one.

There is already some work on adapting GAs for evolving solutions in dynamic environments. However, the same can not be said for Genetic Programming. The goal of this thesis is to fill that gap. We will do so by transposing some of the existing mechanisms for GAs to GPs. Moreover, we will propose novel approaches, that have not yet been employed in GPs. We will test the developed algorithms in three well known benchmark problems, with different types of dynamic environments, and proceed to do a statistical analysis of the collected data.

Keywords: Evolutionary Algorithms, Genetic Programming, Dynamic Environments

Resumo

Os Algoritmos Evolucionários (AE) são uma família de heurísticas da área da Inteligência Artificial que tem sido aplicada com sucesso em problemas de aprendizagem, optimização e design de vários domínios de aplicação. Actualmente, os AE são divididos em Algoritmos Genéticos (AG) e Programação Genética (PG). Os Algoritmos Genéticos evoluem soluções para um dado problema. Por outro lado, a Programação Genética evolui programas que, quando executados, produzem as soluções para os problemas alvo.

Grande parte das experiências bem sucedidas com Algoritmos Evolucionários foram realizadas em ambientes estáticos, isto é, ambientes cujas condições se mantêm constantes ao longo do tempo. No entanto, existem muitas aplicações do mundo real que envolvem ambientes dinâmicos, sendo que os próprios problemas veriam ao longo do tempo.

A dificuldade em evoluir soluções em ambientes dinâmicos surge de um problema comum nos Algoritmos Evolucionários. Esse fenómeno, conhecido como convergência prematura, mostra-se quando a população converge para uma área de boa qualidade do espaço de procura, sendo que os indivíduos se tornam muito semelhantes entre si. Geralmente, em ambientes estáticos, isto leva à descoberta de soluções sub óptimas. Por outro lado, em ambientes dinâmicos, este fenómeno pode não só atrasar, como dificultar a recuperação da qualidade da população após uma alteração ambiental. Tal é especialmente notório em casos de alterações com grande severidade.

Actualmente, existem alguns trabalhos focados na adaptação de Algoritmos Genéticos para Ambientes Dinâmicos. No entanto, tal não é verdade para Programação Genética. O objectivo deste trabalho é colmatar essa lacuna, partindo de mecanismos propostos para Algoritmos Genéticos e adaptando-os para Programação Genética. Para além disso, serão propostos novas abordagens para lidar com este tipo de ambientes. Todos os algoritmos propostos serão testados em três problemas com diferentes tipos de dinâmicas. Posteriormente, os dados recolhidos serão alvo de análise estatística.

Palavras-chave: Algoritmos Evolucionários, Programação Genética, Ambientes Dinâmicos

Acknowledgements

I would like to thank my advisor Prof. Dr. Ernesto Costa, for giving me the opportunity of doing research, for his support and guidance.

I thank my family, girlfriend, friends and coleagues, for all the support given, not only while making this thesis but throughout all my time at University of Coimbra.

Finally, I would also like to thank all members of the Ecos research group, for proportioning a good and stimulating working environment.

This thesis was supported by the project MassGP (PTDC/EEI-CTP/2975/2012), FCT, Portugal.

Contents

List of Figures

Abbreviations

EA	**E**volutionary **A**lgorithm
GA	**G**enetic **A**lgorithm
GP	**G**enetic **P**rogramming
ES	**E**volutionary **S**trategies
EP	**E**volutionary **P**rogramming
DE	**D**ynamic **E**nvironment
SGA	**S**tandard **G**enetic **A**lgorithm
TGA	**T**ransformation **B**ased **G**enetic **A**lgorithm
ETGA	**E**nhanced **T**ransformation **B**ased **G**enetic **A**lgorithm
HMGA	Triggered **H**yper**M**utation **G**enetic **A**lgorithm
MEGA	**M**emory-**E**nhanced **G**enetic **A**lgorithm
MIGA	**M**emory-**I**mmigrants **G**enetic **A**lgorithm
AMGA	**A**ssociative-**M**emory **G**enetic **A**lgorithm
VMEA	**V**ariable-size **M**emory **E**volutionary **A**lgorithm
SGP	**S**tandard **G**enetic **P**rogramming algorithm
TH	**T**riggered **H**ypermutation
FM	**F**ixed **M**emory
HM	**H**ypermutation **M**emory
TM	**T**ransformation **M**emory
IM	**I**mmigrants **M**emory
RIM	**R**andom **I**mmigrants **M**emory
HMTS	**H**ypermutation **M**emory, **T**ransformation and **S**tandard GP algorithm
TMHS	**T**ransformation **M**emory, Triggered **H**ypermutation and **S**tandard GP algorithm
FMTH	**F**ixed **M**emory, **T**ransformation and Triggered **H**ypermutation

Chapter 1

Introduction

1.1 Problems

Throughout the years, computers have been used to solve real world problems. Some of those problems can be said to be easy, as they have an analytical solution and do not take much time to be solved. Unfortunately, there are many important problems that either do not have an analytical solution or are computationally intractable. For solving these problems, computer scientist have been using heuristics, which yield faster, yet approximate, solutions.

1.2 Evolutionary Algorithms

In this thesis we are interested in problem solvers known as Evolutionary Algorithms (EA), a family of heuristics inspired by natural processes, Mendel's genetics and Darwin's Theory of Evolution. EAs can be further categorized in Genetic Algorithms (GA) and Genetic Programming (GP). The main difference between these two types of heuristics is that while Genetic Algorithms evolve solutions to the target problem, Genetic Programming evolve programs that produce the solutions for the target problems.

The evolution of solutions are meant for specific problems, which can be considered as being either static or dynamic. As an example, the evolution of a classifier is usually a static problem. However, if the conditions defining the classes change over time, it becomes dynamic.

1.3 Motivation

Evolutionary algorithms usually suffer from a problem known as premature convergence. Over time, the entire population has a tendency to converge to a good quality area of the search space, thus considerably losing diversity. While having good quality, it is not guaranteed that it is the best area of the search space. This leads to the algorithm getting trapped in sub-optimal solutions, not being able to find the global optimum.

In dynamic environments, the premature convergence is also troublesome. If the environment changes significantly, it is expected that the optimum solution is also quite different from the previous one. Considering that the population has low diversity, it will be difficult to explore distant areas of the search space, thus making it difficult to find good solutions in the new environment.

The lack of population diversity causes the standard algorithm to perform poorly in dynamic environments. Thus, it is necessary to employ techniques aimed at enabling the evolutionary algorithms to deal with such environments.

1.4 Objectives

To deal with dynamic environments, many researchers proposed several modifications to the standard Genetic Algorithm. These modifications turned out to be very different, relying on using mechanisms like memory, immigrants, multi populations or adaptive parameters. Unfortunately, little work has been done on adapting Genetic Programming for dynamic environments.

The goal of this work is to fill that gap, i.e., to transpose some of the techniques devised for GAs to GPs and devise novel GP algorithms adapted for dynamic environments, and experimentally assess their usefulness.

Our work will be based on a set of benchmark problems representative of different classes, such as symbolic regression, binary classification and the evolution of a controller for an ant in the Santa Fe trail.

1.5 Working Hypothesis

In this work we pursue the hypothesis that the ability of evolutionary algorithms to cope with dynamic environments can be increased by maintaining a reasonable amount of population diversity over the generations, keeping knowledge from the past, or both,

1.6 Organization

The rest of this document is organized as follows: in Chapter 2, we give the fundamental background for understanding this thesis, in Chapter 3, we describe the state of the art on approaches for dynamic environments, in Chapter 4, we describe the benchmark problems, their dynamics, the chosen fitness metrics and the function and terminal sets that the algorithms shall use. We also describe the statistical tests that we will use to analyse the results of our experiments. In Chapter 5, we describe the simple techniques and discuss their results. In Chapter 6, we describe novel hybrid techniques which combine some of the simple techniques, and describe their results, with the ultimate goal of finding the best algorithms for each case, i.e, benchmark problems or scenarios. Finally, in Chapter 7, we summarize the results and give suggestions for future work. The results of the statistical tests are present in Appendix A.

Chapter 2

Background

In this chapter we briefly discuss some concepts that serve as background for this thesis.

2.1 Evolutionary Algorithms

For some problems, an exact solution is too hard to find. For others the search for that solution would take so long that would make the it useless. In those situations we are usually interested in the next best thing, finding a good enough solution in a reasonable amount of time. Evolutionary Algorithms (EAs) are search heuristics that can be employed in these latter cases. They are inspired by Darwin's Theory of Evolution by Natural Selection and Mendel's genetics.

An evolutionary algorithm starts by randomly creating a population of possible solutions. Each candidate solution is called an individual. It then proceeds to evolve them as if they were living beings. There are some variation operators, typically crossover (that resembles sexual reproduction) and mutation. From the application of those operators to some parents it produces new individuals, the offspring, that join the existing population. Finally, it has to select the survivors, that will constitute the population to be evolved in the following generation.

Selection is usually performed based on the quality of the individuals, the most fit having a higher change to survive. The described process is illustrated in algorithm 1.

Algorithm 1 Generic EA

Require: $popSize, nGens, problem$
Ensure: $bestIndividual$
  $pop \leftarrow createInitialPop(popSize, problem)$
  $popFitness \leftarrow evaluate(pop, problem)$
  while $currGen \leqslant nGens$ **do**
    $parents \leftarrow selectParents(pop, popFitness)$
    $offspring \leftarrow variation(parents)$
    $offspringFitness \leftarrow evaluate(offspring, problem)$
    $pop \leftarrow survivorsSelection(pop, offspring, offspringFitness)$
    $bestIndividual \leftarrow selectMostFit(pop)$
  end while
  return $bestIndividual$

Where the variable *popSize* holds the size of the population, the *nGens* holds the total number of generations, the *pop* is the population, the *popFitness* holds the fitness of each individual and the *currGen* holds the index of the current generation.

Evolution moves an initial random population towards a good quality area of the search space, often causing low population diversity, i.e, all individuals are very similar to each other. In Static Environments this may lead to the discovery of local optima and not the global optimum, an event known as premature convergence. However, in Dynamic Environments, this is specially problematic. Depending on the severity of the environmental modification, i.e, if the new environment is similar or completely different from the previous one, it may be difficult for the EA to find new good quality solutions before the next change. Thus, new EAs, capable of coping with Dynamic Environments, have to be developed. Some of the used techniques focus on population diversity, either maintaining it at an acceptable level, or increasing it when environmental changes are detected. Others use knowledge from past environments in order to steer the population towards better areas of the search space. These techniques will be described in Chapter 3.

2.2 Tree-based Genetic Programming

Genetic Programming is a type of EA which evolves individuals that have no predefined form or length and that have to be executed for solving the problem. Examples of such individuals are computer programs, mathematical functions and rule based classifiers. In order to achieve this, some of its details are different from the other Evolutionary

Algorithms. However, GP follows the same basic general procedure: iteratively evolve a population of individuals by reproduction with mutation.

In the following Subsections we will describe with detail the major elements of GP that are specific to this family of EA. The description follows the standard GP as defined by Koza in [9].

2.2.1 Representation

In this work we will focus on Tree-based Genetic Programming. Unlike other EAs whose individuals' representations are vectors of numbers or characters, in the standard GP each individual is represented by an expression tree. Those trees are composed by nodes and branches. Each node is matched either to the Function Set or to the Terminal Set. The Function Set is composed by all possible functions that we can compute in order to perform actions, and return values. The Terminal Set is composed by all the numerical constants, input variables or functions that require no arguments. For example, in a simple Symbolic Regression problem, a common Function Set is composed by the four arithmetic operations, Function Set = {+, -, *, /}, where / represents the protected division. The protected division is similar to the normal division operation, except in case the denominator is 0, where it returns a predefined value. In our work we chose to return 1 in that situation. The Terminal Set for this problem is composed by all the input variables, and a random constant generator.

2.2.2 Population Initialization

In order to increase initial diversity, it is wise to create trees of many different shapes and sizes. However, like in the other EAs, we should not make any restrictions on where the new individuals will be located in the search space, unless those restrictions are meant to spread them as most as possible. The simplest way of not making such restrictions is creating the individuals randomly, using an uniform distribution. Two common methods for creating GP individuals are the *Grow* and the *Full* methods. The *Grow* method creates trees whose branches can have a depth up to the specified maximum depth. The decision on whether to make a branch longer or shorter is random. On the other hand, the *Full* method creates trees in which all branches have the maximum specified depth. In order to increase the initial diversity, these methods are usually combined in the *Ramped Half and Half*. In this method half of the trees are created using the *Grow* method, and the other half are created using the *Full* method. Furthermore, greater diversity is achieved by having trees of different depths. For that reason trees are created with depths ranging from 2 to the maximum specified depth, in a way that all depth levels have the same number of trees. This is represented in algorithm 2.

Algorithm 2 Ramped Half-and-Half

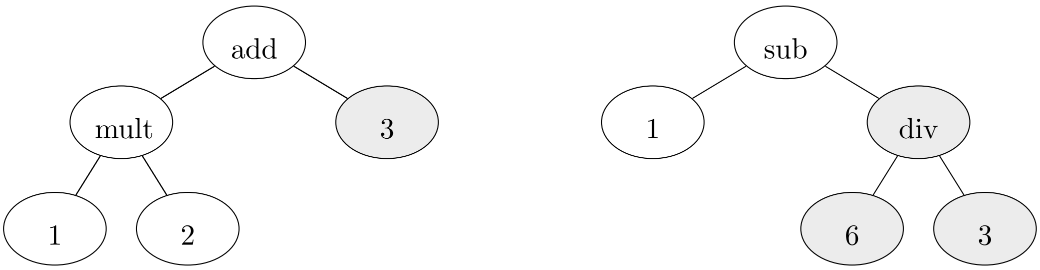
Require: *popSize, maxDepth, terminalSet, functionSet*
Ensure: *pop*
  $indivsPerDepth \leftarrow popSize/(maxDepth - 1)$
  $currDepth \leftarrow 2$
  while $currDepth \leqslant maxDepth$ **do**
    $nIndivs \leftarrow 0$
    while $nIndivs \leqslant indivsPerDepth$ **do**
      if $mod(nIndivs, 2) = 0$ **then**
        $pop \leftarrow pop \cup createGrowIndividual(currDepth, terminalSet, functionSet)$
      else
        $pop \leftarrow pop \cup createFullIndividual(currDepth, terminalSet, functionSet)$
      end if
      $nIndivs \leftarrow nIndivs + 1$
    end while
    $currDepth \leftarrow currDepth + 1$
  end while
  return *pop*

2.2.3 Variation Operators

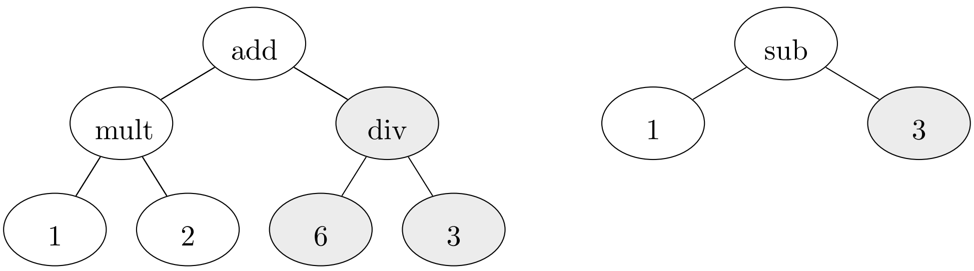
Evolution is only possible by varying the existing individuals. That can be made either by crossover or mutation, being common to use both. Crossover is normally used to introduce bigger modifications into the individuals. On the other hand, mutation is usually, but not always a less disruptive process. The other main difference between both mechanisms is their inspiration. Crossover resembles sexual reproduction, being necessary at least two parents to produce the offspring. On the other hand, mutation tries to replicate the natural counterpart of genetic mutations, being only necessary one individual to produce another, somewhat similar to the former. This last mechanism can be seen as a form of asexual reproduction.

2.2.3.1 Crossover

In tree-based Genetic Programming, crossover made by exchanging subtrees of two parents, thus the name subtree crossover. This is achieved by randomly choosing one node of each individual and exchanging the subtrees rooted at those nodes. The choice of these nodes should not be completely random. As there are many more leaf nodes than internal nodes, there would be a great probability that only small portions of the parents would be exchanged. To counteract that effect a simple technique is usually employed. By using a 90% probability of choosing an internal node and 10% of choosing a leaf node, we can increase the odds of making bigger modifications to the individuals, without significantly reducing the number of possible combinations of individuals. This mechanism is depicted in figure 3.1



(A) Parent trees. The nodes in gray are the selected subtrees.



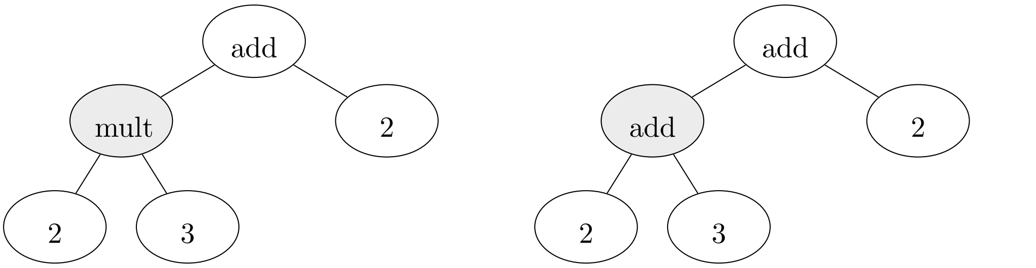
(B) Offspring trees. Created by switching the selected subtrees from each parent.

FIGURE 2.1: Subtree Crossover

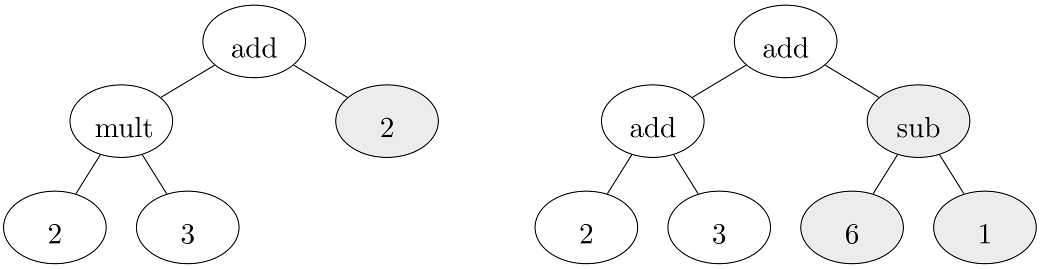
2.2.3.2 Mutation

Mutation is usually done in one of two configurations: node mutation or subtree mutation. Node mutation starts by randomly selecting one node from the tree. Then, this node is replaced by another of the same type, i.e, leaf nodes are only replaced by nodes from the terminal set and internal nodes are only exchanged with nodes from the function set. Subtree mutation consists in randomly selecting one subtree from the individual and replacing it by another, randomly generated, subtree. Both methods are exemplified in figure 2.2 .

In both cases there is the need to check if the replacing node is compatible with the one being replaced, that is, has the same arity, expects parameters of the same types and returns a value of the same type as the one being replaced.



(A) Node Mutation. The tree on the left is the individual prior to mutation, with the selected node in gray. The tree on the right is the result of the mutation.



(B) Subtree Mutation. The tree on the left is the individual prior to mutation, with the selected node in gray. The tree on the right is the individual after the mutation.

FIGURE 2.2: GP Mutation

2.2.4 Selection

In order to evolve a population it is necessary to select individuals, whether it is for reproduction or for surviving in the next generations. In the next subsections we describe the most common methods for selecting individuals in each of these situations.

2.2.4.1 Selection of the parents

For creating new individuals by means of sexual reproduction it is necessary to select the mates. This selection can be made randomly, however, it is most commonly guided by the quality of the individuals. For this latter situation there are a number of available methods:

- *Roulette Wheel.* This method selects only one individual, with a probability proportional to its relative fitness. A common implementation consists in computing the relative fitnesses of the individuals and ordering them from the worst to the best. Then, a random number is generated. The individual selected is the first which has a relative fitness larger or equal to that random number.

- *Stochastic Universal Sampling.* This method is similar to the Roulette Wheel, with the difference that a set of individuals are selected. These individuals are equally spaced in terms of relative fitness. Usually, it is implemented in the following manner: as in the Roulette Wheel, the relative fitnesses of the individuals are computed, and ordered from the worst to the best performing. A random number is generated in the interval $[0, \frac{1}{n}]$, where n is the number of individuals to select. Then, another n-1 pointers are created by adding $\frac{1}{n}$ to the original random number. These pointers are used as in the Roulette wheel, yielding a set of n individuals.

- *Tournament Selection.* This method consists in randomly selecting a number of individuals from the population. The best individual from that set is returned as the selected individual.

- *Rank Selection.* This selection method starts by ordering the individuals according to their fitness, from the worst to the best. The probability of a certain individual being selected is proportionate not to its fitness, but to its ranking. This reduces the selective pressure caused by methods that are proportionate to the fitness of the individuals, enabling even the worst possible individual to be selected.

2.2.4.2 Selection of the survivors

The methods described for selecting the parents can also be used for selecting the survivors. However a number of methods have been proposed for this task. Three of the most common methods are described below:

- *Generational.* On each generation, create a set of individuals with the same size as the current population, containing only the offspring. This set will be the population of the next generation.
- *Elitism.* This mechanism is similar to the Generational. However, a number of the best individuals from the current population, known as the elite, survive into the next generation. The remaining individuals to survive into the next generation are drawn from the best of the offspring.
- *Steady State.* This method, generally known as $\mu + \lambda$, consists in using a number, μ, of parents to create a number, λ, of offspring. Afterwords, a set of the same size of the current population is chosen, containing the best individuals from both the parents and the offspring. This set will be the population for the next generation. Common instances of this selection mechanism are $1 + \lambda$ and $1 + 1$.

2.2.5 Quality Evaluation in GP

The fitness of an individual is a measure of how well the solution represented by it solves the target problem in a specific environment.

Taking as an example the evolution of predictors by means of Genetic Programming, the quality of each individual can be measured as the difference between the output and the expected values. On the next subsections we describe some mechanisms for measuring the quality of the individuals.

2.2.5.1 Root Mean Square Error

The Root Mean Square Error (RMSE) is a measure of the difference between the expected and the predicted values. It is usually employed in Symbolic Regression problems, and is formally defined as

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(y_t^* - y_t)^2}{n}},$$

where n is the total number of inputs, y_t^* is the value predicted for input t and y_t is the expected value for that same input. We shall use this fitness function to measure the quality of the GP individuals in the Symbolic Regression benchmark problem, described in Section 4.1.

2.2.5.2 F1 score

The F1 score is a performance measure for classifiers defined by two other measures, Precision and Recall. Both Precision and Recall measure the performance of a classifier by comparing its outputs with the training data.

When training a classifier, the classification of an example may fall in one of the following four categories:

- *True Positive*. In this case the example was correctly classified as belonging to the positive class.
- *True Negative*. In this case the example was correctly classified as belonging to the negative class.
- *False Positive*. In this case the example was classified as being positive when in fact it belongs to the negative class.
- *False Negative*. In this case the example was classified as belonging to the negative class when in fact it belongs to the positive class.

These categories were initially proposed for binary classifiers. However, a multiclass classifier can easily be seen as multiple binaries. In this case each positive and negative classes would correspond to belonging or not to a certain class of the multiclass classifier.

Precision is then defined as the proportion of true positive examples, in the universe of all examples classified as positive. More formally,

$$Precision = \frac{TP}{TP+FP},$$

where TP is the number of true positives and FP is the number of false positives.

Recall, on the other hand, is defined as the proportion of examples correctly classified as being positive in the universe of all positive examples. More formally,

$$Recall = \frac{TP}{TP+FN},$$

where TP is the number of true positives and FN is the number of false negatives.

Finally, the F1 score can then be defined as a combination of Precision and Recall:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}.$$

This fitness function shall be used in the Classification benchmark problem, described in Section 4.2, to measure the quality of the GP individuals.

2.2.5.3 Linear Scaling

Linear Scaling is a method for reducing the error of the solutions for approximating mathematical functions. The basic idea behind it is that an individual may produce a curve with the same shape as the target, but with different scale. By scaling that output it is possible to obtain better approximations from an otherwise worse individual. The scaled output of an individual is computed by:

$$y_s = a + by \ ,$$

being

$$a = \bar{t} - b\bar{y}$$

and

$$b = \frac{\sum_{i=1}^{N}[(t_i - \bar{t})(y_i - \bar{y})]}{\sum_{i=1}^{N}(y_i - \bar{y})^2},$$

where $\bar{t}$ and $\bar{y}$ are the average target and average output, t_i is the i^{th} target, y_i is the individual's i^{th} output and N is the number of examples.

It can then be used with any fitness measure. As an example, computing the MSE of scaled outputs would be done using the following formula:

$$MSE(t, a + by) = \frac{\sum_{i=1}^{N}(a + by_i - t_i)^2}{N}$$

Linear Scaling is further discussed in [7] and [8].

The Fitness Functions are problem dependent. For that reason, it is only in Chapter 4 that we will choose which ones to use.

2.2.6 Bloat and Overfitting

Genetic Programming poses difficulties different from the other Evolutionary Algorithms. In this Section we discuss bloat and overfitting. While the latter exists in many AI families, in GPs it takes another relevance.

Bloat is the uncontrolled growth of individuals without a corresponding fitness increase. This creates some problems, because the programs become more computationally costly to run and harder to interpret. Furthermore, it is commonly accepted that it often may lead to poor generalisation capabilities, as larger individuals are more prone to overspecialise.

Many theories have been proposed for the causes of bloat. In [11, 17, 18], the authors present some of those theories. Luke and Panait [11] devise one theory of their own, and present and test several methods to counteract bloat in GPs. They propose a variant of a pareto-based multi-objective parsimony pressure method to prevent bloat, and compare it with the existing ones. Also, modifications to the Lexicographoc parsimone pressure method are made, in order to tackle problem domains where individuals with equal fitnesses are rare.

In [17], Silva and Costa introduce Dynamic Limits, a set of novel approaches to controlling bloat. They proceed to compare their performance, using as baseline the static depth limit, proposed by Koza in [9].

Silva et. al. [18] make a review of existing theories for bloat and methods for counteracting it. They take special interest in the Crossover Bias theory, and the Operator Equalisation method inspired by it, presenting and comparing two variants of it with Cartesian and Standard GP. They do this comparison for both bloat and overfitting.

Many authors refer to bloat and overfitting as either existing or non existing, without a measure for quantifying them. Motivated by that, in [24] Vanneschi et. al. proposed metrics capable of quantifying this phenomenons. However, that work is intended to be a initial effort, rather than a finished product.

Overfitting is present in every form of EA. In GPs, it gains another expression, as it is commonly accepted that it can be very related to the size of the individuals. However, recently this idea has been contested, as studies indicate that bloat free solutions may overfit, as well as heavily bloated individuals may generalise well. The authors of [24] and [18] further discuss this phenomenon.

While for the first theories bloat control methods may be sufficient to prevent overfitting, other methods are needed for the latter chain of thought. Gonçalves and Silva [5] focus on the prevention of overfitting without regarding bloat. They propose two methods based on interleaving the use of one or more instances of the training data on each generation.

2.3 Measuring the Quality of the Algorithms

While tackling problems with static environments, one can use simple fitness measures. Most commonly, the reported results are the fitness of the best individual at the end of the run, or along the generations. However, dynamic environments pose new difficulties. In these environments it is not adequate to simply report the quality of an individual at the end of the run. If we were to do it, the fitness value obtained would only represent the quality of the individual in the final environment, and give no indication of what happened in the previous ones.

There are many different ways of measuring the quality of the algorithms. We now present some of the available in the literature.

2.3.1 Accuracy of the algorithm

This measure has more than one definition in the literature. In [21] it is defined as the average of the difference between the fitness value of the best individual before an environmental change, when compared to the optimum value. More formally, it is defined as

$$accuracy = \frac{1}{N} \sum_{i=1}^{N} (Err_{i,r-1}),$$

being N the number of changes in a run and r the period (number of generations) between environmental changes. Consequently, $Err_{i,r-1}$ is the difference between the optimum value and the fitness of the best individual just before the change.

On the other hand, Weicker in [27] defined accuracy as being measured at each generation. In this case the accuracy values are normalized, and for that reason there is not only the need to know the optimum fitness value, but the worst possible value as well. This as the advantage of reducing the biases introduced by environmental modifications, as the value of the optimum value is susceptible to change when the environment changes. More formally, it is defined as

$$accuracy(t) = \frac{Fitness(best,t) - Min_{fitness}(t)}{Max_{fitness}(t) - Min_{fitness}(t)} \ .$$

2.3.2 On-line Performance

The On-line performance measure was initially proposed by De Jong in [6] and is intended for adaptive systems that are optimizing the performance of online systems. It measures the fitness of the algorithm to the specified problem and environments, as it evaluates the quality of the entire population throughout a period of time. It can be defined as the weighted average of all evaluations up to the current moment. More formally,

$$on\text{-}line = \frac{\sum_{t=1}^{NumberOfGenerations} (c_t * \frac{\sum_{i=1}^{PopulationSize} (F(indiv_i))}{PopulationSize})}{\sum_{t=1}^{NumberOfGenerations} (c_t)},$$

where NumberOfGenerations is the number of generations passed since the beginning of the run, $F(indiv_i)$ is the fitness of the individual i, c_t is the weight given to the evaluation at generation t, and PopulationSize is the number of individuals in the population.

The weights allow emphasizing the convergence or initial performance of the algorithm. However, De Jong opted for using all weights equal to 1, which leads to the simplification

$$on\text{-}line = \frac{\sum_{t=1}^{NumberOfGenerations} (\frac{\sum_{i=1}^{PopulationSize} (F(indiv_i))}{PopulationSize})}{NumberOfGenerations},$$

that is present in works such as [27] and [21].

2.3.3 Off-line Performance

The Off-line Performance was initially proposed as the weighted average of the fitness of the best individuals throughout the run. More formally, and assuming a minimization problem,

$$\textit{off-line} = \frac{\sum_{t=1}^{NumberOfGenerations}(c_t * u_e^*(a_t))}{\sum_{t=1}^{NumberOfGenerations}(c_t)},$$

where $u_e^*(a_t) = \min\{u_e(a_1), ..., u_e(a_t)\}$, $u_e(a_t)$ is the fitness of the best individual at generation t, NumberOfGenerations is the number of generations passed since the beginning of the run and c_t is the weight given to the fitness of that individual. If it is a maximization problem, this metric can be adapted simply by considering the individual with maximum fitness value instead of the minimum.

The previous method poses a big problem. While blindly comparing the fitness of every individual created since the beginning of the run, we may compare values from different environments, which makes them incomparable. In order to solve this, a modification to this mechanism was made, in which only individuals from the same environment are compared in $u_e^*(a_t)$. Furthermore, the formula was also simplified by assigning the value 1 to every weight. This leads to the following definition:

$$\textit{off-line} = \frac{\sum_{t=1}^{NumberOfGenerations}(u_e^*(a_t))}{NumberOfGenerations} .$$

2.3.4 Average Best Of Generation

This quality measure gives an indication of the average quality of the best individual of each generation, throughout multiple runs. It is formally defined as

$$AverageBOG = \frac{\sum_{r=1}^{NRuns}(\sum_{g=1}^{NGenerations}(f(BOG_{rg})))}{NRuns * NGenerations},$$

where NRuns is the number of runs made, NGenerations is the total number of generations and $f(BOG_{rg})$ is the fitness of the best individual at generation g of run r.

2.3.5 Stability

The stability of an algorithm measures its capabilities of not being affected by environmental changes. The less an algorithm is affected by an environmental change, the more stable it is. More formally, the stability of an algorithm is defined as

$$\text{stability(t)} = \max\{0, accuracy(t-1) - accuracy(t)\}$$

where accuracy is the normalized accuracy described above. The closest the value is of 0, the more stable is the algorithm.

2.3.6 Adaptability

Adaptability is a measure of how close to the optimum solution could the algorithm get in between environmental changes. It is formally defined as

$$adaptability = \frac{\sum_{i=1}^{K}[\frac{\sum_{j=0}^{r-1} Err_{i,j}}{r}]}{K} \ ,$$

where K is the number of changes during the run, r is the duration of the period between environmental changes, measured in generations and $Err_{i,j}$ is the difference between the fitness of the individual at generation j after the environmental modification i, to the optimum value of environment i.

In this work we will use the Off-line Performance and the Best of Generation, which consists of the metric described in Subsection 2.3.4 without the outer mean, as defined in the following function:

$$BOG = \frac{\sum_{g=1}^{NGenerations}(f(BOG_g)))}{NGenerations}$$

By doing this, we obtain a value of the average fitness of the best individual along the generations for each run, instead of obtaining a single value for all runs, so that we can perform statistical analysis.

Using the Off-line Performance, we can have a good idea of how well an algorithm is able to regain quality between environmental modifications. On the other hand, the Best of

Generation gives an idea of how good an individual would be, if it was drawn at any random generation. Thus, we consider that by using these two measures we will have a good view on how the algorithms do throughout the runs.

2.4 Dynamic Environments

In nature, an environment defines everything that is around an individual and constrains its behavior. Darwin's Theory of Evolution dictates that individuals that are most adapted to their environment have more chances of surviving longer and thus reproducing more often.

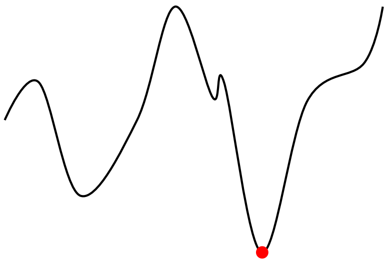
In Evolutionary Algorithms, an environment can be as simple as the function to be approximated or the world where a robot is immersed. In the latter case, the environment can be formed by obstacles, other agents and goals, such as objects that the robot wants to catch.

Usually, in machine learning problems, we are faced with static environments, that is, environments whose characteristics do not vary throughout the experiment. As an example, the evolution of GP individuals for approximating a mathematical function could lead to the discovery of a solution of minimum error, as depicted on figure 2.3A. However, a modification to the environment could change the target function significantly, thus making the previously best individual perform poorly in this new environment. This is depicted on figure 2.3B.

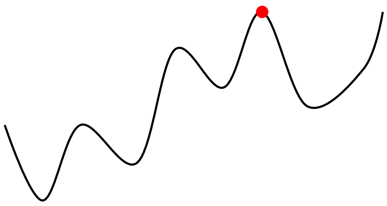
The GP algorithm can recover from the modification to the environment and find new solutions with good quality, specially if it was adapted for dealing with Dynamic Environments. However, this would greatly depend on **when** this change took place and **how** the new situation is. Theoretically, slowly and slightly changing environments are easier to deal with when compared with fast and severely changing ones. We will further discuss this in Subsections 2.4.1 and 2.4.2, following the structure presented in [21].

2.4.1 When does an environment change?

The success of a GP algorithm depends greatly on the moments when the environment changes. Not only it is important to know if the changes happen frequently or rarely,



(A) Before the Environmental change



(B) After the Environmental change

Figure 2.3: Fitness landscapes before and after an Environmental change

but also if they can be predicted or are completely unexpected. This time component of the Dynamic Environments can be characterized according to:

1. *Frequency of change.* This metric is the inverse of the period between environmental changes. This period is the number of generations during which the environment remains unchanged. Typically, longer periods between changes lead to an easier environment to achieve good results in. In practice, if the period is long enough, the environment can be seen as being static, rather than dynamic. On the other hand, if the frequency is at its maximum, changing the environment each generation, it can be rather difficult to achieve a reasonable performance.

2. *Type of change period.* It consists in the way the period between environmental changes varies over the time. It can be considered as being in one of these categories:

   (a) *Periodic or linear.* This is the simplest kind of change period. In this case, the period does not vary, having always the same length. Suppose the environment changes every 20 generations. In this case it will always change every 20 generations.

   (b) *Patterned.* Despite not being constant, the change periods vary accordingly to a well defined, cyclic pattern. As an example, consider that the first environmental change takes place after A generations and that the second takes place B generations after the first one. Further suppose that the environment changes continuously taking place after periods whose length changes from A to B cyclically. We can characterize that environment as having a patterned change period, following an A-B-A pattern.

(c) *Nonlinear.* In this case the duration of the period between environmental changes follow the behavior of a nonlinear function.

(d) *Random.* There is not any mathematical method or expression that can predict when the next environmental changes will take place. They do not follow any pattern of periodicity, being the hardest kind of change period to make predictions in, specially if the frequency of change is also high.

3. *Predictability of the change.* It defines a degree of how much it is possible to predict an environmental change. As we said before, the easiest environments to predict changes are those who have a linear change period, preferably with low frequencies of change. On the other hand, environments with random change periods and high frequencies of change are by far the most difficult ones to make predictions in. It is interesting to be able to predict when the environment will change, as we can use algorithms that are able to predict and prepare to this changes before they take place.

2.4.2 How does an environment change?

Predicting when a environment will change is interesting, but it is only part of the problem, as different types of changes may require different mechanisms to optimally adapt the current solutions to the new environment. As in characterizing when the changes take place, there are also three distinct aspects that characterize how they modify the environment:

1. *Severity of the change.* The first and most simple one is the degree to which the new environment differs from the previous. It is evident that when an environment suffers a great modification we need to employ techniques that enforce the exploration of the search space, whereas if it only slightly changes, the previous population will only have a minor change of quality, thus being better to keep enforcing the exploitation.

2. *Types of environmental changes.* This defines the way a environment changes, that is, if it always change to a new environment, to a previously seen or to one that is similar to another that has been previously seen.

   (a) *Cyclic.* In this case, there is a limited number of environments that appear one after the other in a cycle. As an example suppose that three environments, A, B and C, succeed each other cyclically, in a A-B-C-A pattern.

   (b) *Cyclic with noise.* Similarly to the cyclic environmental changes, here the environments also change in a cyclic manner. The difference is that those environments are not exactly the same every time they reappear. That is due to a noise factor that slightly alters the environments, keeping them very similar, yet different from past occurrences. As an example, if in a cyclic environmental change, the environments changes in a A-B-C-A pattern, adding noise, the pattern would become A-B-C-A'-B'-C'-A".

   (c) *Probabilistic.* Here, there is also a fixed number of environments. However, simply being in one environment at a given moment t does not guarantee what the next environment, at time t+1, will be. In fact, the concept of next environment does not apply in this case, as it is possible, with an associated probability, to change from an environment to any other, at any given time.

   (d) *Random.* There is the possibility that the environment changes to another that has neither any relation to the current or to the past ones.

3. *Predictability of the new environment.* As we stated earlier, the ability to predict the next environment or at least how different it will be from the current one gives the EA a better chance to adapt the current population for the future. That should enhance their quality in the next environment or, at least, reduce the time needed for readjusting the solutions to the new reality. This prediction can be done more easily if the environmental transitions are cyclic. However, they are impossible to predict in cases that exhibit random environmental transitions.

There is a great number of combinations of the above parameters, that translate into many different scenarios, each with its level of difficulty. For the sake of simplicity, we chose to define three scenarios, for each main level of difficulty, i.e., easy, medium and hard. These scenarios are defined as follows:

- *Easy*. This is the simplest of the three scenarios. Here, we are interested in having low frequencies of change, resulting in long periods between them. We also want them to take place periodically, so that there is exactly the same number of generations between changes and the environments succeed each other in order. Finally, they will have low severity, so that the next environment is only slightly different from the current one.

- *Medium*. This scenario has an average level of difficulty, between the easiest and the hardest. We defined two versions of this scenario.

  1. In the first version, the changes take place on patterned times, being the next environment chosen probabilistically, depending on the current one. The result of these changes is an environment that is somewhat different from the previous one.
  2. The second version has periodical changes with higher frequencies of change than the ones used in the Easy scenario. Furthermore, the change between environments is periodical with noise, with means that despite knowing the general form of the next environment we do not know exactly how it will be. The changes also have more severity than the ones from the easiest scenario.

- *Hard*. This is the hardest scenario of the three. The changes will be severe, each environment being quite different from the others. They will change more frequently and at random times.

In Chapter 4 we further describe these scenarios, with the problem dependent details.

Chapter 3

State of the Art

In this Chapter we will review some work that has already been undertaken on the application of Evolutionary Algorithms, more specifically Genetic Algorithms and Genetic Programming, to evolve solutions in Dynamic Environments. We will classify these works based on how they handle those environments. We propose three categories to classify them, depending on (1) their manipulation of the algorithm's Parameters, (2) use of Memory or (3) relying on Immigrants.

1. *Parametric methods*. These methods consist in changing the values of the parameters of an evolutionary algorithm, like the population size, the elite size, the crossover rate or the mutation rate. The parameters are generally altered as a response to an environmental change.

2. *Memory based approaches*. These mechanisms consist in remembering good solutions from past environments and using them in new, similar ones. For that reason, they tend to not be very helpful in scenarios where the environment changes too severely and has no similarity to past occurrences. The memory can be classified as implicit or explicit.

    (a) *Implicit*. Implicit memory schemes are usually implemented by means of Multiploidy. Multiploidy consists in each individual having more genes than the ones that are expressed in its phenotype. The genes that are not expressed can be use to remember good individuals from past environments. A dominance scheme is used to select which genes are represented. Smith and

Goldberg discuss diploidy and dominance schemes in [22]. Furthermore, we give examples of mechanisms with dominance schemes in Section 3.1.2.

(b) *Explicit.* Here, good individuals from past environments are stored in a special location, being also possible to store information from those environments. When an environmental change occurs, those individuals are injected into the evolving population, steering it towards good quality areas of the search space.

3. *Immigrant methods.* In these methods, a number of individuals are generated and injected into the population for further evolution. They can either be generated randomly or based on some individuals. When they are based on the best individual they are called Elitist Immigrants.

3.1 Genetic Algorithms for Dynamic Environments

In this Section we describe the work done on adapting Genetic Algorithms for coping with Dynamic Environments.

3.1.1 Parametric methods

One way to enable an Evolutionary Algorithm to thrive in dynamic environments is to create diversity when a change is detected. On the other hand, another approach would be to maintain a reasonable level of population diversity along the entire run.

For the former approach, one simple method would be to reinitialize the entire population when a change was detected. While this could have some appeal for its simplicity, it would not save any knowledge acquired during the previous generations. Thus, depending on the severity and periodicity of the changes, it would potentially take longer to regain the quality of the population than a method that kept past knowledge.

Another possibility would be to use Triggered Hypermutation as proposed by Cobb in [2] and further investigated in [3] . In this approach, mutation is applied normally with a low probability, called base mutation rate. When an environmental change is encountered, the base mutation rate is increased to a hypermutation rate. The number of performed mutations is probabilistically increased, thus increasing the population diversity along with the possibility of finding better solutions in other areas of the search space.

The severity of the change should be taken in consideration when choosing the amount of increase made to the base mutation rate. An example of that is Variable Local Search, an adaptive operator that only slightly increases the mutation rate at times of environmental change. A study made in [25] concluded that for low severity environmental changes this operator was superior to Triggered Hypermutation, as the optimal solution was not very far away from the existing best.

In [10] it is proposed a method that adapts both the crossover and mutation rates, regarding the fitness of the population and the specific individuals. More specifically, the crossover rate for a set of parents varies accordingly to the formulas:

$$Pc = \begin{cases} \frac{k_1(F_{bestparent} - F_{min})}{F_{max} - F_{average}}, & \text{if } (F_{bestparent} - F_{min}) \leqslant (F_{max} - F_{average}) \\ k_3, & \text{if } (F_{bestparent} - F_{min}) > (F_{max} - F_{average}) \end{cases}$$

On the other hand, the mutation rate for the i^{th} individual varies accordingly to the formulas:

$$Pm = \begin{cases} \frac{k_2(F_i - F_{min})}{F_{max} - F_{average}}, & \text{if } (F_i - F_{min}) \leqslant (F_{max} - F_{average}) \\ k_4, & \text{if } (F_i - F_{min}) > (F_{max} - F_{average}) \end{cases}$$

k_1 and k_2 are scale factors, meant to keep the rates in the interval [0,1], k_3 and k_4 are constants in the same interval. $F_{bestparent}$ is the fitness of the best parent, F_{min} is the worst possible fitness, F_{max} is the best possible fitness, $F_{average}$ is the average fitness of the population and F_i is the fitness of the i^{th} individual. Using these formulas, the mutation rate is lower for fitter individuals and higher for poorer performing ones. Similarly, the crossover rate is directly proportional to the relative quality of the best parent. This encourages the exploitation around good quality individuals and exploration when faced with lower quality ones.

Although this method was proposed directly for preventing premature convergence, it could prove helpful in situations of environmental change, as it promotes a certain level of population diversity.

3.1.2 Memory methods

Memory methods are specially good in cyclical or patterned environments, i.e., environments that have occurred in the past and that will appear again in the future. As we have stated, they can be subdivided into Implicit and Explicit memory methods.

The use of Implicit Memory is present in [4] where a redundant individual representation was used by means of diploidy. This means that each individual had two strings of genes, but only one is expressed at a particular time. The sole purpose of the other string was to keep genetic material that had good quality in past environments. The decision on which genes are expressed in the phenotype is made using a triallelic dominance scheme. In this scheme, each gene can have the symbol 0, 1 or 2, where the symbol 1 represents a recessive gene with value 1, and the symbol 2 represents a dominant gene of value 1. The the symbol 0 dominates the symbol 1 and the symbol 2 dominates the symbol 0, meaning that the expressed gene would only have the value 1 if at least one gene had the symbol 2 or both genes had the symbol 1. For the remaining situations the gene would be expressed as a 0. For the sake of clarity, the outcome of two competing genes is represented in Table 3.1, where the first line and column represent each gene segment and the inner cells represent the outcome.

	0	1	2
0	0	0	1
1	0	1	1
2	1	1	1

Table 3.1: Triallellic dominance scheme

In [14] a four alleles dominance scheme was proposed, were each gene could assume a value 0 or 1 that could be either recessive or dominant. This dominance scheme changed whenever the individuals suffered a fitness reduction greater than 20%. In such situations the recessive genes turned dominant and the dominant into recessive, so that every expressed gene that resulted from different alleles was inverted.

One way to use an Explicit Memory would be to store good quality individuals from past environments and once an environmental change is detected, inject them back to the existing population. If the characteristics of the new environment can be easily perceived, one could opt to select from memory only the individuals that performed well in similar environments.

In [21] four algorithms are compared, all of them using Explicit Memory for coping with Dynamic Environments. In these algorithms, the memory is also used to detect environmental changes. That is achieved by evaluating the entire memory in every generation and, if at least one of its individuals has its fitness altered, the algorithm detects a modification to the environment. The Memory-Enhanced Genetic Algorithm (MEGA) stores good individuals from past environments in memory, which is updated every time the environment changes as well as in random intervals ranging from 5 to 10 generations. When an environment change is detected, the memory is also used to steer the population towards better areas of the search space. This is obtained by creating a new population containing the most fit individuals from the last population and from the memory, in the context of the new environment. The Memory-Immigrants Genetic Algorithm (MIGA) is similar to the previous algorithm. However, the memory contributes to the evolution of the population in every generation. Each generation, the best individual from the memory is used to create, by means of mutation, a number of immigrants that are injected into the population. Unlike the previous algorithms, the Associative-Memory Genetic Algorithm (AMGA) stores in memory information about the environments alongside their best individuals. This is made by storing pairs, each of them containing the best individual in a given moment and a vector containing the allele distribution that characterizes that environment. When an environmental change is detected, the allele distribution vector associated with the best individual is used to create new individuals that are injected into the population. The Variable-Size Memory Evolutionary Algorithm (VMEA) is similar to MEGA, with some important differences. This algorithm adapts the sizes of both the memory and the population, while maintaining the total number of individuals constant. Furthermore, when an environmental change is detected, only the best individual from memory is injected into the population. After some experimentation, the author concluded that VMEA outperforms the other three algorithms.

3.1.3 Immigrant methods

Instead of only increasing the population diversity when the environment changes, it could be useful to maintain it at a reasonable level throughout the generations. Immigrant based approaches are very useful in this situation. There have been some works using different types of immigrants. The simplest one, the random immigrants, consists simply in introducing a number of randomly generated individuals at each generation. Those new individuals will replace part of the ones present in the population, thus increasing the exploration ability of the algorithm, without losing much of the exploitation ability. The individuals to be replaced are selected accordingly to a replacement strategy. A commonly used replacement strategy simply selects the worst individuals of the population to be replaced. A more sophisticated approach, proposed in [30], consists in using hybrid immigrants. This method could address environmental changes of low, average and high severity by creating immigrants both using mutation on existing individuals and generating others randomly. If the environment changes only slightly, the new optimal is expected to be close to the previous one. In that case, it makes sense to create immigrants by mutating the best solution of the last environment. For average severity changes the same can be applied for its dual individual. Finally, for the cases where the environment changes dramatically it is likely that the best solution is at a completely different area of the search space, and so the randomly generated individuals have a better chance of succeeding. The quantity of each kind of immigrants is adapted based on their performance. At a given generation, say **i**, 10 immigrants of each kind were created and evaluated. Suppose the elitist immigrants were the best. Then, on generation **i+1**, the number of elitist immigrants would be increased and the others decreased so that the total number of immigrants would remain constant.

3.1.4 Other methods

In [19], it is proposed an approach that does not use crossover. Instead, it uses Transformation, a variation operator inspired by a biological process that takes place in bacteria. When cells die, their remains, containing DNA, are still present in the environment for an amount of time. During that period, some cells, more specifically some bacteria, are able to incorporate segments of the dead cells' DNA, replacing parts of their own.

Taking inspiration from this process, a computational approach was proposed that substitutes crossover in Genetic Algorithms. On each generation, a number of individuals are selected for transformation, that is, the incorporation of gene segments present in a gene pool into their genetic material. In order to do that, a point in the individual's chromosome has to be chosen, and from that point onward, the genes are replaced by the ones from the foreign segment. Depending on the length of the gene segment, this mechanism allows for both exploration and exploitation of the search space. In [20] four different approaches to applying GA in a dynamic environment are compared, specifically the dynamic 0/1 knapsack problem. They use two variants of GA with transformation, one with Triggered Hypermutation (HMGA) and another one with random immigrants. The difference between the algorithms that use transformation are the parameters. In TGA - Transformation based Genetic Algorithm, the parameters are chosen without any experimentation. On the other hand, in ETGA - Enhanced Transformation based Genetic Algorithm, those parameters are chosen after extensive experimentation. The results of the experiments show that the choice of the approach is very dependent on the problem at hand. If the frequency of change is low, the ETGA is typically better, if it is high, the HGMA is a safer bet. The poor performance of HGMA on lower frequencies of change may be due to the loss of diversity towards the end of the cycles, whereas the ETGA maintains a constant high diversity in its population.

Regarding the selection methods, there are some that encourage both the exploration and exploitation. One of them is called sharing. In this method, individuals that are close in the search space share their fitness, having it reduced. Consequently this encourages the exploration as it lowers the fitness of similar individuals, benefiting the ones that are isolated in different areas of the search space. Another selection mechanism is called crowding. In this mechanism, a number of individuals is selected for reproduction at each generation. For each resulting offspring a number of individuals are selected from the population, being the most similar one replaced by the offspring. That quantity of individuals is called the crowding factor. This counteracts the tendency the individuals have to converge to the same region of the search space. This two selection mechanisms have been combined in the Worst Among Similar replacement scheme. It was able to adapt to environmental modifications by searching for new optimal solutions in different regions of the search space. That ability was due to not only allowing but also promoting competition between similar and different individuals.

3.2 Genetic Programming for Dynamic Environments

Most works on Genetic Programming have focused on static environments. However, some mechanisms that were proposed for avoiding premature convergence and local optimum can be reasonably successful in dynamic environments. That is because a common problem of a traditional Genetic Programming algorithm is the natural convergence of the population towards good solutions of a given environment. When the environment changes, the former good solution can now be quite bad. However, as all the population has converged to that area, it is rather difficult to search in distant areas of the search space. For that reason, solutions that aim at increasing or maintaining a reasonable degree of population diversity are expected to perform better than traditional GP.

In the following Subsections we review some of the proposed methods for enabling GPs to cope with Dynamic Environments.

3.2.1 Parametric methods

In [16] it is proposed a GP variant adapted for Dynamic Environments by modifying some of the parameters of the algorithm.

1. *Elitist proportion.* This consists in the proportion of the best individuals from the previous population that survive in the current one. If the fitness of the best individual has increased, it is likely that we are in a good quality area, thus it is interesting to increase the exploitation capacity of the GP. In order to do it, this ratio is increased by 0.1, with an upper bound of 1. On the other hand, if the fitness decreases, we should encourage the exploration of the search space, thus the elitist proportion is decreased by 0.1, with a lower bound of 0.1.

2. *Crossover probability.* Like in the elitist proportion, when the fitness increases, the Crossover probability is decreased by 0.1 down to a lower bound of 0.1. Similarly, when the fitness decreases this probability is linearly increased by 0.1, up to an upper bound of 0.9. This process creates more individuals in times of need, i.e, loss of quality, and saves resources when good individuals have already been found. However, it can be very disruptive.

3. *Mutation rate.* Here it is proposed to search for a good set of probabilities cyclically. In the mentioned work, three different mutation probabilities are used: the probability that an individual is mutated, that an operator is applied and that each node of the tree is mutated. As small probabilities are better for exploitation and bigger probabilities for exploration, at times of change, all three probabilities

are added a random number in the interval [0,1] and scaled back to that same interval. As soon as a good set of probabilities is found, they remain constant until the environment is modified again.

4. *Culling*. This mechanism aims at increasing the diversity of the population by replacing a portion of the worst individuals by dynamically generated ones. It is similar to the random immigrants, however, each new individual is created with only a terminal symbol as the root node. They are then subject to mutation, accordingly to the three adaptive rates that were described above. This mechanism is applied with a 50% probability.

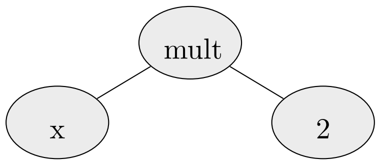
This adaptive approach was compared to a standard GP and Neural Networks trained with the Gradient Descent algorithm. The algorithms were tested on three dynamic classification problems. The authors concluded that the Adaptive Genetic Algorithm outperformed the others in all three problems, regardless of the severity of the environmental modifications.

In [31] the adaptation of the crossover and mutation rates was explored in a different way. In this work, the normal crossover rate is decreased and the mutation rate increased only if after a number of generations a new best individual has not been found. When a new best individual is found, the crossover and mutation rates go back to their original values. This helps the GP to explore different areas of the search space and escape local optimums.

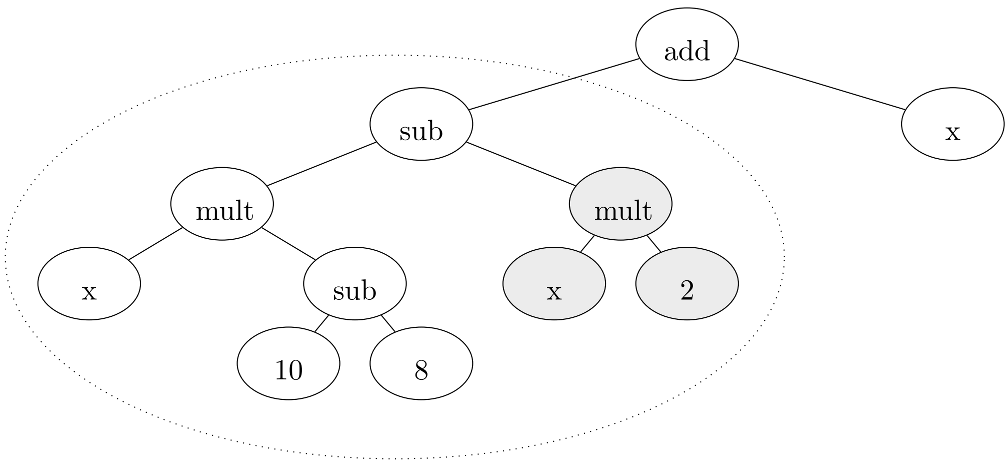
In [23] it was studied the effect of adapting the size of the population accordingly to its quality in the current environment. The main idea is quite simple. When the fitness is good, there is no need to keep a big population, so it can be reduced. The reduction of the population size allows saving resources while the environment does not change. On the other hand, when the quality starts to drop, a need for looking in other areas of the search space emerges, and so the population grows by the addition of new randomly generated individuals.

3.2.2 Memory

To some extent, genetic programming may automatically use implicit memory. That phenomenon exists in the form of introns. Introns are subtrees that have no expressiveness in the phenotype. As an example, take the problem of approximating a mathematical function. Suppose that at a given generation, the tree depicted in Figure 3.1A has a good quality. When the environment changes, it may start performing poorly and be excluded from the population. However, it may still survive if, by means of crossover, and before being excluded from the population, it had found its way into an individual that has good quality in the new environment. This tree is now part of an intron, as depicted in Figure 3.1B, with the complete intron being the selected subtree. As it has no expression in the phenotype, that is, it simplifies to 0, has no influence on the performance of the individual, and may endure for many generations. Later, by means of crossover, it is possible to reappear as an individual and perform well on another environment.



(A) Fit individual before the environmental change.



(B) Individual that endures in the new environment. The previous individual (in gray) survives as part of an intron, which is surrounded by the dotted ellipse.

FIGURE 3.1: Survival of an unfit individual as an intron.

In [26] was proposed an algorithm, DyFor, that aims at Time Series Forecasting in Dynamic Environments using GP. In order to succeed it uses a sliding time window, whose length adapts throughout the run. When the fitness is good, the window grows, so that future predictions are based on more data from the past. On the other hand, when the fitness drops it is a sign of an environmental change. In this situation the window starts to shrink in order to ignore data from a now old environment, enabling the evolution of solutions adequate to the new reality.

As described above, this method makes use of implicit memory through introns. Furthermore, it also uses explicit memory in the form of dormants. Dormants are solutions that have had good quality in the past and were stored in memory. When an environmental change is detected these dormant solutions are injected back into the general population in order to speed up the convergence to a good quality area of the search space. They are continuously injected into the population until the window stops shrinking, i.e. the data contained in the window was generated by only one environment.

3.2.3 Other methods

In [15] artificial dynamic environments were used with the purpose of speeding up the convergence of Standard GP algorithms for Symbolic Regression problems. In it, the authors aim at approximating the polynomial function: $x+x^2+x^3+x^4+x^5+x^6+x^7+x^8$. The authors used a quite simple idea to turn this static problem into dynamic. They defined two kinds of changes, modular and structural. For the modular changes only the operators were modified. They created a set of functions containing the target and others like $x - x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + x^8$ and $x + x^2 + x^3 + x^4 + x^5 - x^6 + x^7 + x^8$. The structural changes, on the other hand, consisted on omitting parts of the target polynomial. For this case a set of functions was also created, containing the target and sub functions like $x + x^2 + x^3$ and $x + x^2 + x^3 + x^6 + x^7 + x^8$. This sets of functions will be the targets for the dynamical environments. After a given number of generations a change will be made by randomly choosing a function from the set and using it as the target. If this choice is not random but in a way that the degree of the polynomial increases, then we will have a step evolution. The authors concluded that the use of Dynamic Environments not only leads to a faster convergence of the GP individuals into a good quality area of the search space, but also contributes to the discovery of higher quality solutions. This is a kind of step evolution, with a similar concept to the one proposed by Baptista and Costa in [1]. In that paper, the step evolution is used to reduce the time needed for simulations of open-ended evolution of artificial life.

3.3 Detecting Changes

Some algorithms for Dynamic Environments involve reacting to the changes, while others simply maintain a high level of population diversity. For those who do react to change, it is necessary to detect it. In this section we describe some of the most common methods in literature.

3.3.1 Sentinels

The first method we describe is the use of sentinels, which was proposed by Morrison in [13]. Sentinels are GA or GP individuals that are uniformly distributed in the search space. Although they are part of the population, they do not evolve, not being possible to replace them. For that reason, they can be used as indicators of the fitness landscape. Alterations to their fitness indicate modifications in the environment. By the number of affected sentinels and by the differences in their fitness it is possible to deduce the severity of the change and, sometimes, where are the new good quality areas in the search space.

3.3.2 Fitness of the Best Individual

A simple way of detecting an environmental change is by an alteration of the fitness of an individual. In elitist algorithms, a number of the best individuals found so far survive unaltered in the next generation. After a number of generations in the same environment, it becomes harder to find better solutions, which means that the best individual rarely changes. By evaluating the elite in all generations we have a good indicator of the environment. If the best individual has not been altered but its fitness has, it is an indication of a modification in the environment. This method was used by Riekert in [16].

3.3.3 Memory

The explicit memory used in some algorithms can be used not only for dealing with the environmental changes, but also for detecting them. Similarly to the previous methods, the evaluation of the individuals in the memory on each generation allows for detecting environmental changes. An environmental change is detected if a memory individual has not been updated between evaluations but its fitness has changed. This method has been used in [29] and [21].

3.3.4 Other Methods

Another method for detecting environmental changes was used by Cobb in [3]. In that work, the performance of a generation is computed as the running average of the fitness of the best individuals of the population from the past 5 generations. As in the other methods, when the performance of a generation is lower than the previous one, an environmental change is detected.

In our work we are interested in comparing different approaches to adapting GP to Dynamic Environments, not in comparing different methods of detecting changes. For that reason we chose to detect changes using the fitness of the best individual, as it is simple and possible to use in all algorithms.

Chapter 4

Experimental Study

In this Chapter we present the Benchmark Problems in which we will test our algorithms, their Dynamics and Fitness measures.

The Benchmark Problems were chosen from three classes that are representative of the most common applications of GPs. In fact, in [12], the authors concluded that the most common classes of benchmark problems in the surveyed papers were Symbolic Regression, Classification and Path Planning, in this order. In the Path Planning class, the most common problem is the evolution of an Artificial Ant, i.e, the Santa Fe Ant Trail. In a more recent study, [28], White et. al. concluded that despite some controversy, this three classes of problems are still relevant to the field, and should be part of a benchmark suite. Another conclusion from that study is that it is still a common practice for authors to develop their own problems.

We chose to develop the Symbolic Regression and Classification problems based on previous works in Dynamic Environments. On the other hand, to the best of our knowledge there still are no efforts in GPs for Path Planning in Dynamic Environments. For that reason, we opted to use the Santa Fe Ant Trail problem, as it is well known in the field of GP.

The Dynamics will be defined having Section 2.4 of Chapter 2 as reference. Furthermore, the Fitness measures used are described in Section 2.2.5 of Chapter 2. Finally, we also present the problem dependent Function and Terminal Sets.

4.1 Symbolic Regression

In this Section we describe the problem of Symbolic Regression. We start in Subsection 4.1.1 by defining it in its static form. In Subsection 4.1.2 we describe how we made this problem dynamic and, in Subsection 4.1.3, we describe how to evaluate the quality of the GP individuals evolved for this problem.

4.1.1 Definition

Regression and Symbolic Regression are methods for approximating the mathematical model that produces a set of data. On one hand, Regression only searches for the parameters that make a predefined, fixed model, fit the data. On the other hand, Symbolic Regression makes no assumptions on the form of this model, searching for both its structure and for its parameters. In our work we will pursue a Symbolic Regression approach to approximate polynomial functions.

One example of a Symbolic Regression problem is the approximation of a polynomial function such as $x + x^2$. A Symbolic Regression Regression approach would consist on combining mathematical operations, the input variable x and numerical constants in order to obtain the best possible approximation to the target function. One of the best individuals, represented in Figure 4.1 would simply be a combination of an addition and a multiplication operation, and the input variable x.

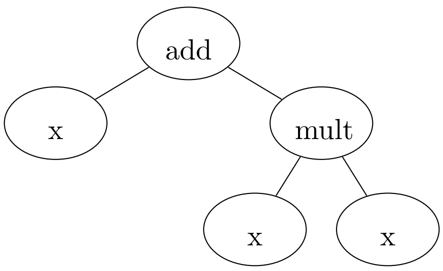


FIGURE 4.1: Example of a GP individual encoding the target function.

Using GP, each individual of the population would consist of a tree representation of mathematical functions. The internal nodes represent the mathematical operations, such as addition and multiplication, and are drawn from the function set. The leaf nodes are drawn from the terminal set, that contains the input variables and other elements, such as random constants. The fitness of each individual is a measure of how close it is to the target function. Given a set of input-output pairs, the fitness is computed by comparing the results yielded by the individual to the expected ones.

In this work we will approximate polynomial functions inspired by the ones present in [15], which are described on Section 3.2.3 of Chapter 3. However, the goal of that study was to prove that dynamic environments can contribute to speed up the convergence of Standard GP individuals to the optimal solution. In our work we are interested in taking mechanisms developed for adapting GAs to Dynamic Environments and using them with GPs to solve some Dynamic Problems. Because of that we do not have a target function that we force as the environment for the final generations. Instead, we have a set of environments that change among them. Further details are given in the following sections.

4.1.2 Dynamics

Usually a Symbolic Regression problem has a single target function that is approximated. However, this problem can be dynamic, which means that the target function changes from time to time.

In this thesis, we will only use modular modifications, which consist of changing the operators between operands. Thus, the severity of the changes depends both on the number of operators modified and on the degree of the operand whose operator is modified.

As it was stated on Section 4.1.1, we will adapt an already presented problem for the purpose of this thesis. As we do not intend to evolve a specific target function as the original authors do, we shall use a different nomenclature and will not force one specific function as the final target.

There will be three scenarios, each with an increasing difficulty level. The difficulty levels were chosen having as reference the characteristics of Dynamic Environments described in Section 2.4 of Chapter 2 i.e., **when** the environmental changes take place and **how** they modify the environments. As it was said, changes that take place with low frequency and periodically, that only modify the environment slightly, are easier than those that take place very often, at random moments and that have high severity. Furthermore, scenarios in which the modifications are cyclical are easier than those that have random changes, as there are more opportunities to take advantage of past knowledge. This scenarios are described in the following Subsections.

4.1.2.1 Easy

The first level is meant to be the simplest. For that reason there will be a long period in between environmental changes, with a fixed duration of 200 generations. The modifications made to the environment will consist in alternating between two target functions, that differ from each other in only one operator, thus being cyclical with low severity and taking place periodically. The set of possible target functions is composed by:

- A: $x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + x^8$
- B: $x - x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + x^8$

4.1.2.2 Medium

Here, the modifications will be more severe and frequent. The environmental changes take place at pre-determined moments, following a pattern of 100 - 120 - 80 - 100 generations. Furthermore, the number of possible environments is limited to 5. On each environmental change, the next environment shall be chosen probabilistically, depending on the current one. The available environments will consist in:

- A: $x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + x^8$,
- B: $x + x^2 - x^3 + x^4 + x^5 - x^6 + x^7 - x^8$,
- C: $x - x^2 + x^3 - x^4 + x^5 + x^6 - x^7 + x^8$,
- D: $x + x^2 - x^3 - x^4 + x^5 + x^6 + x^7 - x^8$,
- E: $x + x^2 + x^3 + x^4 - x^5 - x^6 - x^7 + x^8$.

The transitions between environments are made accordingly to the transition graph of Figure 4.2, whose connections have the probabilities defined in Table 4.1.

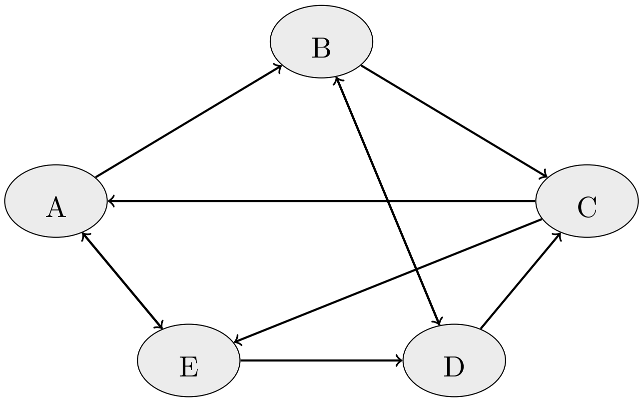


FIGURE 4.2: Transitions between the environments of the medium difficulty regression scenario

	A	**B**	**C**	**D**	**E**
A	0%	75%	0%	0%	25%
B	0%	0%	40%	60%	0%
C	5%	0%	0%	0%	95%
D	0%	35%	65%	0%	0%
E	75%	0%	0%	25%	0%

TABLE 4.1: Transition probabilities between the environments of the medium difficulty regression scenario

4.1.2.3 Hard

In this scenario the evolution of good solutions is expected to be more difficult than in the previous ones. Here the changes will not only be more frequent, as they will take place at any random moment, with a minimum change period of 50 generations. After that number of generations there will be a 50% probability of the environment being modified in the current moment. Furthermore, there will be no predefined number of environments. Instead, each operator will have a 50-50 chance of being inverted, i.e. positive members becoming negative and vice versa, thus causing random modifications to the environment. Initially, the environment shall be defined by the original function: $x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + x^8$, as in the other scenarios.

4.1.3 Fitness

In the Symbolic Regression problem the fitness of each individual will be computed using the Root Mean Square Error, as described on Section 2.2.5 of Chapter 2. As it is a measure of the error, the lower the value of RMSE, the fitter the individual.

4.2 Classification

In this Section we describe the problem of Classification. We start in Subsection 4.2.1 by defining it in its static form. In Subsection 4.2.2 we describe how we made this problem dynamic and, in Subsection 4.2.3, we describe how to evaluate the quality of the GP individuals evolved for this problem.

4.2.1 Definition

In this Subsection we will address a binary classification problem in a D-dimensional world as defined in [16] . The two classes are separated by a decision hyperplane defined by the equation: $H(x) = \sum_{i=1}^{D-1}(a_i * x_i) + c$.

The role of the GP algorithms is to evolve individuals that approximate the decision frontier and classify the examples depending on their location relative the hyperplane. For this study we set D to 10 dimensions. The bias, c, which is the value in which the hyperplane intersects the axis of the D^{th} dimension, is set to 0 and each coefficient of orientation, a_i, is initially set to 1. These parameters can be modified for obtaining new environments.

4.2.2 Dynamics

As discussed in Subsection 4.2.1, this problem was made dynamic by varying the decision hyperplane at each environmental change. The decision hyperplane is defined by its orientation and bias. Depending on the intended severity of change we could modify only one or both parameters in different degrees. However, modifying both parameters would make the process of defining the modifications more difficult and prone to error, possibly resulting in harder modifications than intended. For this reason, in this work we chose to only modify the orientation of the hyperplane.

4.2.2.1 Easy

In this scenario the modifications will have low severity. Like in the easiest scenario of the Symbolic Regression problem, here we shall only use two distinct environments.

The first being the original, and the second a slight modification of it. The existing environments are formally defined as:

- $H_1(x) = \sum_{i=1}^{9} x_i$,
- $H_2(x) = 2x_1 + \sum_{i=2}^{9} x_i$,

An environmental change simply consists in replacing the current target hyperplane by the unused decision hyperplane from the set, thus resulting in cyclical modifications to the environment. Furthermore as we intend to have long periods between environmental changes, they will only occur once every 200 generations.

4.2.2.2 Medium

Here the modifications will have a bigger impact than in the previous scenario. For that reason every environment differs on two orientation coefficients from the others. Furthermore, the modified coefficients are noisy, being added a value from a Gaussian distribution, with mean 0 and standard variarion 1, as defined by the formulas below. This means that although the changes still take place periodically, the environments will never be exactly the same.

The environmental changes will take place periodically, every 100 generations, being the modifications made to the environment cyclical, following the sequence:

- $H_1(x) = \sum_{i=1}^{9}(x_i)$,
- $H_2(x) = (1 + N(0,1))x_1 + \sum_{i=2}^{8}(x_i) + (1 + N(0,1))x_9$,
- $H_3(x) = \sum_{i=1}^{7}(x_i) + \sum_{i=8}^{9}((1 + N(0,1))x_i)$,
- $H_4(x) = x_1 + \sum_{i=2}^{3}((1 + N(0,1))x_i) + \sum_{i=4}^{9}(x_i)$,
- $H_5(x) = \sum_{i=1}^{4}(x_i) + (1 + N(0,1))x_5 + \sum_{i=6}^{7}(x_i) + (i + N(0,1))x_8 + x_9$,

where N(0,1) is a value drawn from a Gaussian distribution with mean 0 and standard deviation 1.

4.2.2.3 Hard

In the hardest scenario the severity of the modifications will be substantially higher, by changing the orientation of the decision hyperplane in 5 randomly selected coefficients at each environmental change. These coefficients, a_i, are modified by adding a value from a Gaussian distribution with mean 0 and standard deviation 10 to them. In this scenario the environmental changes will take place at any given generation, as long as there is a minimum of 50 generations between them. After that period, every generation will have a 50% probability of being the moment of environmental modification. Thus, in this scenario the modifications occur frequently, at random moments, causing severe and random modifications to the environment.

4.2.3 Fitness

The fitness of the individuals will be assessed using the F1 score, as defined on Section 2.2.5 of Chapter 2. Unlike the RMSE, the higher the value of F1 score the better the individual.

4.3 Santa Fe Ant Trail

In this Section we describe the problem of evolving the controllers of artificial ants, for navigating the Santa Fe Trail. We start in Subsection 4.3.1 by defining it in its static form. In Subsection 4.3.2 we describe how we made this problem dynamic and, in Subsection 4.3.3, we describe how to evaluate the quality of the GP individuals evolved for this problem.

4.3.1 Definition

The Santa Fe Ant Trail Problem consists in evolving the controller of a digital ant so that it is able to navigate in a two dimensional world, collecting a food trail within the limited number of moves. The world is represented by a 32x32 toroidal grid. Being a toroidal grid means that the world is not bounded. If an ant moves beyond an edge, it will just reappear in the opposite side of the grid, as if it was a spherical world.

In its original form, the ants only have 400 moves to collect the trail of 89 food pellets depicted on Figure 4.3 .

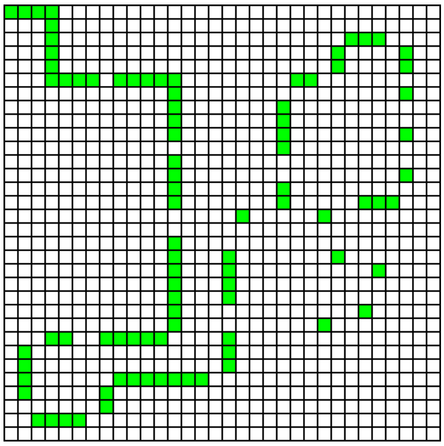


Figure 4.3: Original Santa Fe Trail

By thinking in a natural environment, it is not normal that the food pellets remain at the same positions throughout time. Their movement may be caused by many different factors such as the wind or other coexisting agents in the environment. Furthermore, it is also possible that new food pellets appear or that the existing ones disappear, as they are eaten or made available by other agents. The details of the implemented environmental modifications are described in the next Section.

4.3.2 Dynamics

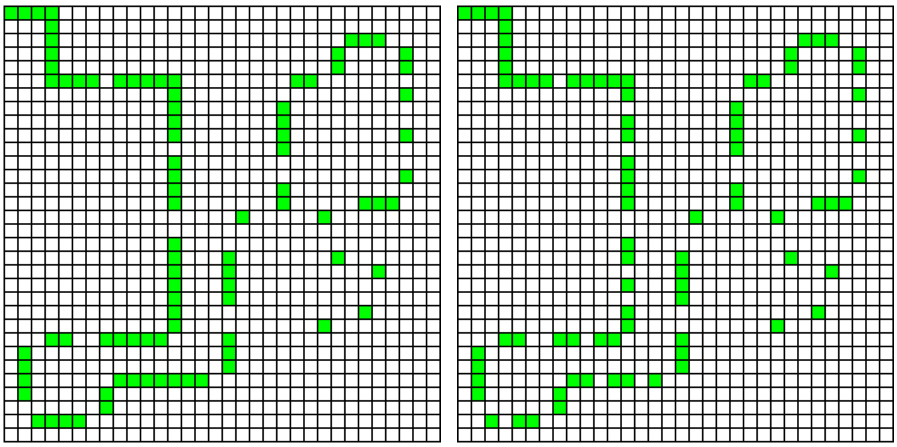
The static environment of this problem can be made dynamic by altering the position and number of the food pellets. For the sake of simplicity, the natural environment factors are not replicated nor are the other agents that can interact with the environment. However, we can simply simulate the effects of their actions by moving the position of the food pellets, omitting some or adding others.

The severity of the change depends both on the position of the omitted food pellets as well as on their number. For example it is much harder to make a turn in the trail if the food pellet immediately after the turn has been omitted. On the other hand, if a considerable amount of food pellets are omitted, even from straight segments, it may create a reasonable difficulty to the ant. In case the food pellets are not omitted but moved, the severity will again depend on both their past and new positions as well as on their number. Moving the pellets one cell to the left or right will result in a slight change of the trail. However, moving a considerable amount of food pellets to faraway locations of the world may result in a complete redesign of the trail.

In our work we opted by simply omitting a number of food pellets depending on the level of difficulty desired. As in the other problems, we specify the details of the three difficulty scenarios in the following Subsections.

4.3.2.1 Easy

In the easiest scenario we shall only make minor modifications to the original path. As in the previous problems, there will only be two environments, the original and one created by removing 7 food pellets from it in a way not to create holes bigger than one cell. These environments are depicted on Figure 4.4.



(A) Original trail (B) Trail 2

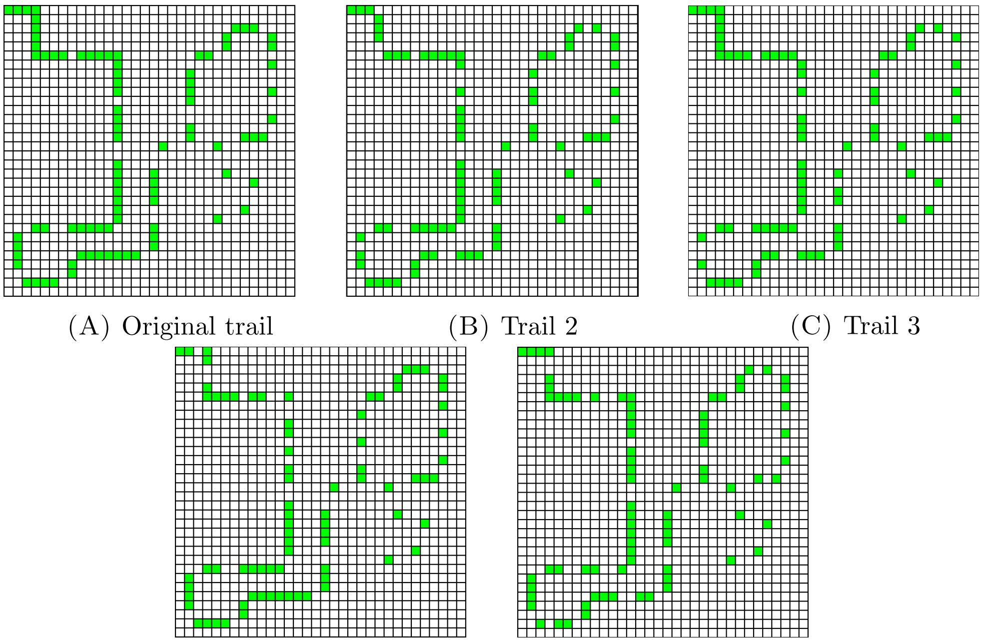
Figure 4.4: Easy Environments for the Santa Fe Ant Trail Problem

Each 200 generations the current environment is replaced by the other, thus having periodic changes with low frequency, that cause cyclical modifications with low severity.

4.3.2.2 Medium

In the medium difficulty scenario there will be five environments, the original and four created by removing 12 food pellets from it. The food pellets were removed from positions chosen specifically in order not to create holes bigger than two cells in the trail. As in the Symbolic Regression problem, here the environmental changes will also take place at non periodic, yet patterned moments, following the same pattern of 100 - 120 -

80 - 100 generations between modifications to the environment. At each environmental change, the next environment shall be selected accordingly to the exact same scheme of the medium scenario of the Symbolic Regression problem. This means that the choice of the next environment is probabilistic, depending on the current one, as can be seen in Figure 4.2 and in Table 4.1, where environment A translates to the original trail and environments B to E translate to trail 2 to 5 respectively. The existing environments are depicted on Figure 4.5.



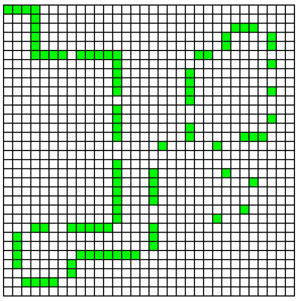
(A) Original trail (B) Trail 2 (C) Trail 3

(D) Trail 4 (E) Trail 5

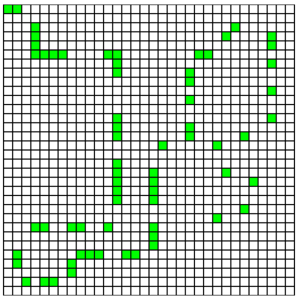
Figure 4.5: Medium difficulty Environments for the Santa Fe Ant Trail problem

4.3.2.3 Hard

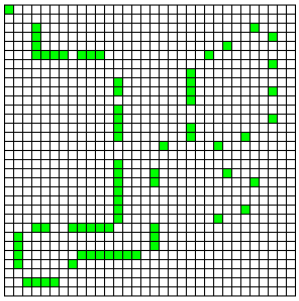
Finally, in the hardest scenario the modifications will be very significant. There will exist five environments, the original and four others created by removing 17 food pellets from it. As this is the hardest scenario, each environmental change will consist on randomly selecting one environment from the set. Furthermore, the environmental changes will take place at any generation, given that there is a minimum change period of 50 generations. That is achieved by using a 50% probability of selecting a new environment on each generation past the minimum change period. Thus, this scenario presents frequent changes that take place at random moments, causing severe and random modifications to the environment. The environments present in the set are depicted in Figure 4.6.



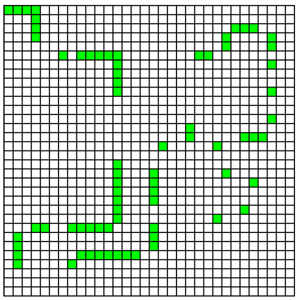
(A) Original trail



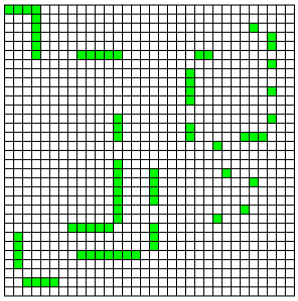
(B) Trail 2



(C) Trail 3



(D) Trail 4



(E) Trail 5

FIGURE 4.6: Hard Environments for the Santa Fe Ant Trail problem

4.3.3 Fitness

The Fitness of each individual in this problem is the simplest of the three, as it consists on simply counting the number of eaten food pellets by the ant within the specified time limit. As in the Classification problem, here we are interested in maximizing the fitness measure, as the more food pellets are eaten by an individual's ant, the more quality it has.

4.4 Function and Terminal sets

The Function and Terminal sets are not dependent on the algorithms described on Chapter 4, but on the problems on which they will be tested. For that reason, and as all the algorithms use the same Function and Terminal sets, we chose to only describe them in this Chapter.

For the classification and regression problems, the Function Set is composed by: F={+, -, *, /}, where / stands for protected division, that returns 1 if the denominator is 0. The Terminal Set is composed by: T = {x, *randomConstantGenerator* }, where x is the vector of the available input variables and *randomConstantGenerator* generates random numerical constants. While the random constants are helpful in the classification problem, they are not part of any of the target polynomials of our Symbolic Regression problem. However, we chose to still include the random constant generator in the Terminal Set of this latter problem to mimic the cases where we do not know the target function to be approximated.

The Terminal and Function sets for the Santa Fe Ant Trail problem are the same as the ones used by Koza in [9]. The Terminal Set is composed by three simple actions: T={Move, Right, Left}. Left and Right consist in turning the ant 90^{o} in those directions. Move consists in moving the ant 1 cell in its current orientation. The Function Set is composed by three functions, F={IfFoodAhead, Progn2, Progn3}. IfFoodAhead is a conditional function that tests if there is a food pellet in the cell directly ahead of the ant. If there is food ahead, it executes the first of its arguments. On the other hand, if there is not food ahead, it executes its second parameter. Progn2 and Progn3 take two and three parameters respectively, which is the sole difference between these functions. They process their parameters in sequence, regardless of the surrounding environment. As an example, the expression Progn2(Right, Move) would make the ant turn 90^{o} to its right and move 1 cell in that direction.

4.5 Statistics

In this Section we describe the statistical tests that we shall use in the analysis of the algorithms' results.

4.5.1 Normality tests

The first step in a statistical analysis is to discover if the sampled data follows a normal distribution. This can be made through descriptive statistics or by applying normality tests to the data. Depending on the results of these tests, we may need to assess the homogeneity of the variances, in order to choose between parametric or non parametric tests for comparing the data.

In the descriptive statistics we have the skewness and the kurtosis. The skewness is a measure of the distribution's asymmetry. If it is 0, the tails on both sides of the distribution are symmetric. Otherwise one is longer than the other. On the other hand, the kurtosis is a measure of how peaked a distribution is. Positive values indicate a peaked distribution, whereas negative values indicate a flat distribution. A normal distribution has values for both skewness and kurtosis of 0.

The Kolgomorov-Smirnov test compares two distributions based on two hypothesis. The null hypothesis states that the sampled data is drawn from a reference distribution, in our case, a normal distribution. The alternative hypothesis states that the sampled data is not drawn from the specified distribution. The test yields two results, test statistic, D, and a p-value. If the p-value is lower than our significance value, in this case 0.05, we can reject the null hypothesis that the data follows a normal distribution. Otherwise, we apply the Levene test to asses the homogeneity of the variance. This test compares if two samples are drawn from a distribution with equal variances. In our case, we compare the sampled data with the normal distribution. Thus, the null hypothesis states that all samples are drawn from distributions with equal variances. The alternative hypothesis states that the samples are drawn from distributions with different variances. The test yields two results, a test statistic W, and a p-value. If the p-value is superior to our significance value, 0.05, we can not reject the null hypothesis that the samples are drawn from distributions with equal variances, thus, in combination with the result of the Kolmogorov-Smirnov test, we can assume that the sample is from a normal distribution. Otherwise we consider that the sampled data is drawn from another distribution. In our work we were always able to reject the null hypothesis of the Kolmogorov-Smirnov test, thus, we did not have to apply the Levene test.

As we shall discuss in Chapter 5, all of our samples follow non normal distributions. Thus, we use non parametric tests, more specifically the Friedman's Anova and the Wilcoxon Signed Ranks.

4.5.2 Friedman's Anova

The Friedman's Anova is a non parametric test that compares many paired samples to test whether they are drawn from the same distribution. Its null hypothesis states that all samples have the same distribution. The alternative hypothesis states that there are samples drawn from different distributions. It yields two results, a Chi-Square statistic and a p-value. If the p-value is lower than our significance, i.e. 0.05, we reject the null hypothesis, thus considering that there are differences between the samples, in our case, between the tested algorithms. However, this test does not give any information on which sample are different or the direction of that difference. For that reason we have to apply a pairwise test, such as the Wilcoxon Signed Ranks test. To minimize error propagation, we have to apply a continuity correction. We chose the Bonferroni correction, a simple but conservative technique. By using this technique we can guarantee the 95% confidence interval, used at a global level.

4.5.3 Wilcoxon Signed Ranks

The Wilcoxon Signed Ranks test is a non parametric test that compares two paired samples. Its null hypothesis states that the median of differences between the two samples is 0. In our case, this means that the two algorithms perform equally according to that metric. On the other hand, the alternative hypothesis states that the median of differences between the two samples is not 0, thus existing differences between the algorithms. This test yields four important results. A z-score, a p-value, the sum of positive ranks and the sum of negative ranks. If the p-value is lower than the significance value, i.e. 0.05, we can reject the null hypothesis and consider that there are differences between the algorithms. Afterwords, comparing the sum of positive and negative ranks we can determine which algorithm is better, depending on if we are minimizing or maximizing the values, i.e minimizing error or maximizing accuracy.

Chapter 5

Results and Analysis

In this Chapter we describe and compare the performance of the Simple Techniques, i.e., the Standard GP algorithm (SGP), the Triggered Hypermutation (TH), the Immigrants (I) and the Fixed Memory (FM). These are the three techniques used in GAs that we proposed to transpose to GPs. This process consists in taking the standard GP algorithm and modifying it accordingly to the concepts of each technique. We shall statistically analyse their performance in the three Benchmark Problems and difficulty levels. The tables containing the results of the statistical tests can be seen in Appendix A.

5.1 Description

5.1.1 Standard GP

The Standard Genetic Programming algorithm (SGP) is a traditional Genetic Programming algorithm, as defined by Koza in [9]. The chosen parameters are based on his work, and subject to adaptations derived from empirical experiments and necessary to accommodate the environmental changes. A tournament scheme shall be used for selecting the parents. The selection of the survivors is generational, with an elite that holds 6.25% of the population, which translates to 25 individuals. Our parameters are represented in table 5.1.

The maximum tree depth is the only parameter that is problem dependent. That is due to the target polynomials to be evolved in Symbolic regression being too hard to achieve

Population size	400
Number of Generations	5000
Crossover Probability	80%
Mutation Probability	10%
Maximum tree depth	7 or 10
Tournament size	50
Elite size	25
Population Initialization	Ramped half and half

TABLE 5.1: Parameters for the Standard GP Algorithm

with trees of seven levels. For that reason, in that Benchmark Problem the maximum tree depth was increased to ten. In the remaining problems this was not made because it was not necessary to achieve the target solutions, and because doing it would allow for larger trees that would slow down the evolution.

5.1.2 Triggered Hypermutation

The parametric method we chose to employ is called Triggered Hypermutation (TH). It consists greatly of a standard GP, similar to the one described in Section 5.1.1, with the exception that it will use subtree mutation with a base mutation rate of 10%. When an environmental change is detected, the base mutation rate is replaced by an hypermutation rate of 50%. After a period of 25 generations the mutation rate goes back to the normal base rate. The remaining parameters are equal to those of table 5.1.

5.1.3 Immigrants

The Immigrant approach we shall use is called ERIGA, and was proposed by Yang and Tinós in [30] for Genetic Algorithms. It consists in the combination of Random and Elitist Immigrants. On each generation, a number $n_{ei}(t)$ of immigrants is created by subtree mutation from the best individual of the population. Also, a number $n_{ri}(t)$ of random immigrants are generated. The number of elitist immigrants $n_{ei}(t)$ and random immigrants $n_{ri}(t)$ are variable, but the total number of created immigrants, $n_i(t)$ remains constant over the generations. As an example, if in a generation the elitist immigrants are more successful than the random ones, the elitist number for the next generation will be increased by a factor α and the number of random immigrants is decreased by the same amount. In this algorithm, subtree mutation is employed both during evolution and in the creation of the elitist immigrants, with probabilities of 10% and 100% respectively. The total number of immigrants created on each generation is 30% of the population size and, in the beginning, half of them are elitist and the other half random. The minimum and maximum amount of individuals to create from any kind is 4% and 26% of the population size. The value of α is set to 2% of the population size. The rest of the parameters are equal to the ones of table 5.1 and, in the remaining, of this document we shall refer to this algorithm as Immigrants.

5.1.4 Fixed Memory

Our Memory enhanced GP will be inspired by MEGA, proposed in [29] by Yang. As in MEGA, only the information of the best individuals is stored in memory, and no information about the environments is kept. The memory will be updated periodically, every seven generations, as well as when an environmental change is detected. Updating the memory consists in replacing one of its individuals by another one selected from the population. If the update is scheduled, the individual to be inserted into the memory is the best from the current population. On the other hand, if the update is due to an environmental change, the individual to be inserted into the memory is the one which performed best in the previous environment.

Upon updating the memory, the algorithm first checks if its capacity has not yet been reached. If there is still room available, the individual from the population is inserted into the memory. On the other hand, if the memory is full, select one individual from it according to a Replacing Strategy and replace it by the individual from the population, provided that the former has less quality than the latter. The Replacing Strategy we will use consists in replacing the most similar individual. In our work, the similarity of two GP individuals is assessed by how close their performance is, in a specific environment. This is computed according to the formula:

$$similarity(individual_1, individual_2) = |fitness(individual_1) - fitness(individual_2)|,$$

where smaller values mean more similarity between two individuals. Thus, the individual from memory to be replaced is the one which has the lowest value of similarity with the individual selected from the population.

When an environmental change is detected, all individuals from both memory and population are evaluated and if the best individual from memory is better than the worst from the population, the former replaces the latter in the population.

There are not many studies supporting the relationship between the size of the memory and the type of problems to be solved. As we have a limited number of environments, it would be reasonable to set the memory size so that it had capacity for containing at least one individual of each environment. However, the GP algorithm is not supposed to know a priori the number of existing environments. For that reason we shall use a size for the memory that depends on the size of the population. A common capacity in the literature for GAs is 10% of the size of the population. We will use that size for the memory.

The remaining parameters are similar to the ones from table 5.1. In the remaining of this document we shall refer to this algorithm as Fixed Memory, due to the fixed length of its memory.

5.2 Variation Operators

In this Section we discuss the variation operators used in these algorithms. Although they are already well known in literature and have been described in Sections 2.2.3.1 and 2.2.3.2, it is worth revisiting them, as they have suffered some minor modifications.

The crossover operator employed is a traditional subtree crossover, with the difference that the offspring is created with a depth restriction. In order to accomplish this, a random subtree is selected from the first parent. Then, the maximum depth for the second subtree is computed in order to allow the offspring to grow up to the imposed limit. The second subtree is then selected from the second parent, with the restriction that it must not be deeper than the previously computed depth, and put into the place of the first subtree.

In the mutation, we are not interested in making very disruptive changes in the individuals. For that reason, we only allow the mutation of subtrees with depths up to half the depth of the individual. The new subtree is generated with the restriction that when in the individual, it does not make it deeper than the maximum allowed, thus allowing the individual to grow or shrink.

5.3 Normality Tests

The chosen normality test, Kolmogorov-Smirnov, determines that the data collected from all algorithms follows non normal distributions, hence the usage of the non parametric tests of Friedman's Anova and Wilcoxon Signed Ranks, depending on the number of algorithms to compare simultaneously. Further discussion on this tests is present in Section 4.5 of Chapter 4. The results of this tests for the Symbolic Regression, Classification and Santa Fe Ant Trail benchmark problems are available in tables A.7 to A.12, A.41 to A.46 and A.75 to A.80, respectively. Furthermore, the results of the descriptive statistics of the Symbolic Regression, Classification and Santa Fe Ant Trail benchmark problems are present in tables A.1 to A.6 ,A.35 to A.40 and A.69 to A.74, respectively.

The first step of the statistical analysis is determining if the sampled data follows a normal distribution. This is important for deciding whether to use parametric or non parametric tests. We started by describing the data, as depicted in tables A.1, to A.6, A.35 to A.40 and A.69 and A.74, paying special attention to the values of skewness and kurtosis. Furthermore, we applied the Kolgomorov-Smirnoff test to the data, as represented in tables A.7 to A.12, A.41 to A.46 and A.75 to A.80, respectively for the Symbolic Regression, Classification and Santa Fe and Trail benchmark problems. From the analysis of the results of this test, as well as the values of skewness and kurtosis, we can reject, with 95% confidence, the null hypothesis that the data follows a normal distribuition, thus employing non parametric tests. For comparing the groups of algorithms we use the Friedman's Anova. This choice is due to having more than two groups of paired samples. After this step, we proceed to test the pairs of the algorithms with the Wilcoxon Signed Ranks test. This is again chosen because the samples are paired and the data does not follow a normal distribution. This conclusions are also valid for the Hybrid Techniques, whose normality tests are discussed in Subsection 6.2, from Chapter 6.

5.4 Symbolic Regression

In this Subsection we discuss the performance of the simple techniques in the three scenarios of the Symbolic Regression Benchmark Problem, measured with the Offline Performance and Best of Generation metrics.

5.4.1 Offline Performance

The Friedman's Anova test yields p-values of 0.046, 0.0682 and 0.0013, for the easy, medium and hard scenarios. As we have a 0.05 significance value, we conclude that there are statistically significant differences between the algorithms in the easy and hard scenarios, but not in the medium. This results are present in table A.13.

Afterwords, the Wilcoxon Signed Ranks determines that, in the easy scenario, the SGP is significantly worse than the Immigrants, because it has a p-value lower than our significance value corrected for continuity, i.e, 0.0167, and the sum of positive ranks is higher than the one of the negative ranks, i.e, the SGP has larger errors more often than the Immigrants. This difference has a medium effect size of -0.4976.

As the Friedman's Anova determined, the Wilcoxon Signed Ranks test yielded no statistically significant differences in the medium scenario. However, in the hardest scenario it determined that the SGP was significantly worse than the Triggered Hypermutation, with a p-value of 0.0077 and the sum of positive ranks being larger than the one of negative ranks. This difference has a medium effect size of -0.4863. The results of the Wilcoxon Signed Ranks for the easy, medium and hard scenarios can be seen in tables A.14, A.15 and A.16, respectively.

5.4.2 Best of Generation

The application of the Friedman's Anova revealed statistically significant differences between the algorithms in the three scenarios. However, when using the Offline Performance the test indicated that there were no differences between the algorithms in the medium scenario. This is due to that metric only analysing how good the best solution of each algorithm is immediately before an environmental change, whereas the Best of Generation analyses the performance of the best individual on each generation.

The Wilcoxon Signed Ranks test indicated that, in the easy scenario, the SGP is significantly worse than the Immigrants. On the other hand, in the medium and hard scenarios, the differences detected by the Friedman's Anova test are due to the SGP being statistically worse than the Fixed Memory. The results of the Friedman's Anova and the Wilcoxon Signed Ranks test can be seen in tables A.17 to A.20.

5.5 Classification

In this section we discuss the performance of the simple techniques in the three scenarios of the Classification benchmark problem, when measured with the Offline Performance and Best of Generation metrics.

5.5.1 Offline Performance

The Friedman's Anova test detected statistically significant differences between these algorithms in the three situations, yielding p-values of 0.0044, 0.0 and 0.0, for the easy, medium and hard scenarios, respectively. However, after applying the correction for continuity, and at a 95% confidence interval, we were unable to find any significant differences in the easy scenario. This may be due to the employed correction technique, the Bonferroni correction, being too restrictive, as we suspect that the differences detected by the Frieman's Anova are due to SGP being worse than the Immigrants, with a p-value of 0.0256. However, this p-value is larger than our significance value of 0.0167, thus not making it significant. In the medium scenario all of the three techniques proved to be superior to the SGP, with the Immigrants based algorithm being the most different. The same happened in the hardest scenario, with the difference that the most different from the SGP was the Fixed Memory. This may be due to the high frequency of changes causing the other algorithms to have little time to adapt to the new environment, thus giving an advantage to an algorithm with memory. The results of these tests can be seen in tables A.47 to A.50.

5.5.2 Best of Generation

We proceeded in the same manner as for the Offline Performance. The results of the Friedman's Anova are depicted in table A.51. For the easy, medium and hard scenarios, we obtained p-values of 0.0003, 0.0 and 0.0, being smaller than out significance value, thus enabling us to reject the null hypothesis that there are no differences between the algorithms.

We proceeded to apply the Wilcoxon Signed Ranks test, which results are depicted in tables A.52, A.53 and A.54. They show that, for the easiest scenario, that the only statistically significant difference is between SGP and Fixed Memory, being the SGP superior to the Fixed Memory. This may be due to the long periods between environmental changes allowing the standard algorithm to regain quality, whereas the Fixed Memory has its evolution biased from past knowledge, thus being more probable to getting trapped in local optima solutions.

In the medium scenario, both the Triggered Hypermutation and the Immigrants based algorithm are superior to the SGP, with the Fixed Memory not being sufficiently better than the SGP to be statistically significant. Finally, in the hardest scenario, all three algorithms are better than SGP.

5.6 Santa Fe Ant Trail

In this Subsection we discuss the performance of the simple techniques in the Santa Fe Ant Trail benchmark problem.

5.6.1 Offline Performance

Here we analyze the performances of the algorithms, measured with the Offline Performance metric. Firstly, the application of the Friedman's Anova determined that, at a 95% confidence interval, there are only statistically significant differences between the algorithms in the medium and hard scenarios, yielding p-values of 0.721, 0.0 and 0.0, for the easy, medium and hard scenario, respectively.

The Wilcoxon Signed Ranks test not only confirms the results of the Friedman's Anova, as well as determines that in the medium scenario the difference detected between the algorithms is due to the SGP being worse than Fixed Memory. In the hard scenario, the differences are due to the SGP being better than the Triggered Hypermutation and worse than the Fixed Memory. While the SGP being worse than Fixed Memory is not surprising, as the use of memory is favourable in scenarios with frequent changes, the results with the Triggered Hypermutation are unexpected. This may be due to the fact that in the hardest scenario, the environments change very frequently, about every 50 generations, and the period of hypermutation, i.e., the number of generations where the algorithm uses the hypermutation rate, being 25 generations, which in this scenario translates to about 50% of the period between changes. As the Triggered Hypermutation is a very disruptive mechanism, there is a possibility that the algorithm does not have time to stabilize before a new environmental change, thus yielding worse results than the less disruptive approaches. The fact that this is only true in this benchmark problem, strengthens the No Free Lunch theorem.

The results that lead to this conclusions are present in tables A.81 to A.84.

5.6.2 Best of Generation

We now discuss the performances of the simple techniques, when measured with the Best of Generation metric. The results of the Friedman's Anova test determine that there are statistically significant differences between the algorithms only in the medium and hard scenarios, thus being consistent with the results obtained with the Offline Performance.

The Wilcoxon Signed Ranks test determined that the differences in the medium and hard scenarios are due to the SGP being worse than the Fixed Memory, having a larger effect size in the hard scenario. The difference between using this metric and the Offline Performance is that the in the hard scenario there are no statistically significant differences between the SGP and Triggered Hypermutation, thus, on average, the best individual from the populations of these algorithms are not very different. The results of these tests are present in tables A.85 to A.88.

5.7 Summary

In this Section we resort to a more visual type of comparison to sum up the ones made in the previous Sections, using only the Wilcoxon Signed Ranks test. Here, a ▼ means that the first algorithm is worse than the second, i.e, in Table 5.2, the SGP is worse than the Triggered Hypermutation in the medium and hard scenarios of the Classification benchmark problem. The ▲ means that the first algorithm is better than the second and the • means that there is no statistically significant difference between the two algorithms. Due to space limitations, we refer to the algorithms by their previously defined acronyms.

As can be seen in tables 5.2 and 5.3, there is no consistency in the performance of the algorithms along the benchmark problems. However, considering the Offline Performance, we can safely say that for the Symbolic Regression, the Immigrants is better than SGP in the easy scenario, and the Triggered Hypermutation is better in the hard scenario. In the Classification problem all three algorithms are superior to the SGP in the medium and hard scenarios, not being any differences in the easy scenario. In the Santa Fe Ant Trail problem, the Fixed Memory is better than the SGP in the medium and hard scenarios, and, also in the hard scenario, the Triggered Hypermutation is worse than the SGP.

Using the Best of Generation metric, we can say that in the Symbolic Regression problem, the Immigrants variant is better than the SGP in the easy scenario and the Fixed Memory is better than the SGP in the medium and hard scenarios. In the Classification problem the Fixed Memory is worse than the SGP in the easy scenario, but better in the hard scenario. The Triggered Hypermutation and the Immigrants are better than the SGP in the medium and hard scenarios. Finally, in the Santa Fe Ant Trail, the Fixed Memory is better than the SGP in the medium and hard scenarios.

Depending on the problem at hand and the type of scenario, we can select different algorithms for maximize the quality of the solutions. More specifically, each algorithm is good in the following situations:

- SGP

  - Classification problem with slow and periodical changes, causing slight and cyclical modifications to the environment.

- Immigrants algorithm
  - Symbolic Regression problem, with slow and periodical changes that cyclically and slightly modify the environment
  - Classification problem with medium and high frequency of changes, taking place at periodical or random moments, causing cyclical with noise or random medications to the environment, with medium or high severity.

- Fixed Memory
  - Symbolic Regression, Classification and Santa Fe Ant Trail problems, with medium and high frequency changes, that take place at patterned, periodical or random moments, causing probabilistic, cyclical with noise, or random modifications to the environment, with medium and high severities

- Triggered Hypermutation
  - Classification problem with medium and high frequency of changes, taking place at periodical or random moments, causing cyclical with noise or random medications to the environment, with medium or high severity.

	Symbolic Regression			**Classification**			**Santa Fe Ant Trail**		
Algorithms	Easy	Medium	Hard	Easy	Medium	Hard	Easy	Medium	Hard
SGP vs TH	●	●	▼	●	▼	▼	●	●	▲
SGP vs I	▼	●	●	●	▼	▼	●	●	●
SGP vs FM	●	●	●	●	▼	▼	●	▼	▼

TABLE 5.2: Simple Techniques - Offline Performance

	Symbolic Regression			**Classification**			**Santa Fe Ant Trail**		
Algorithms	Easy	Medium	Hard	Easy	Medium	Hard	Easy	Medium	Hard
SGP vs TH	●	●	●	●	▼	▼	●	●	●
SGP vs I	▼	●	●	●	▼	▼	●	●	●
SGP vs FM	●	▼	▼	▲	●	▼	●	▼	▼

TABLE 5.3: Simple Techniques - Best of Generation

Chapter 6

Hybrid Techniques

In this chapter we describe the Hybrid Techniques and discuss their performance. The tables containing the results of the statistical tests can be seen in Appendix A.

6.1 Description

In this section we describe the algorithms whose creation was only planned after the intermediate report. The first one is based on a variation mechanism proposed for GAs. The others are created by combining two or more techniques from both the current and previous Chapter. The developed techniques are:

- Transformation (T)
- Transformation Memory (TM)
- Hypermutation Memory (HM)
- Immigrants Memory (IM)
- Random Immigrants Memory (RIM)
- Transformation Memory Hypermutation SGP (TMHS)
- Hypermutation Memory Transformation SGP (HMTS)
- Fixed Memory Transformation Hypermutation (FMTH)

6.1.1 Transformation

The first of these variants is the transposition of the algorithm proposed in [19] for GAs. Despite the Transformation not being an Hybrid Algorithm per se, we chose to discuss its results here, as its adaptation to GPs was not part of the original plan.

This algorithm is similar to the Standard GP, but instead of crossover, it uses a variation operator called transformation. This operator creates a new individual by replacing part of the genetic material of an existing individual with a segment existing in the environment. It is thus a form of asexual reproduction, needing only one parent to create an offspring.

This operator is applied with the same probability as the crossover is in the other algorithms. The gene segments are contained in a pool with a size of 40% the size of the population. Furthermore, as we use a tree representation, the segments are in fact subtrees, with depths no greater then 4 levels. Half of these segments are created randomly. The other half are subtrees from individuals selected by tournament from the population.

In their original work, the authors determined that this method was able to keep a good level of diversity in a GA's population, throughout the run. For this reason, it should perform better than the standard GP algorithm in Dynamic Environments.

The remaining aspects of the algorithm is similar to the Standard GP, as defined in Subsection 5.1.1, of Chapter 5.

6.1.2 Hypermutation Memory

This variant combines the Triggered Hypermutation and the Fixed Memory algorithms, defined in Subsections 5.1.2 and 5.1.4, keeping the parameters of the Simple Techniques unaltered. The reason for doing this, is the behaviour of the Triggered Hypermutation and Fixed Memory algorithms. In Figure 6.1, we can see that the Triggered Hypermutation (red line) is more disruptive, often being able to recover faster and achieving better solutions than the Fixed Memory (black line). However, after some generations, the Fixed Memory algorithm has solutions representative of each environment in its memory. By remembering those solutions it has a better performance after the environment changes, as it can simply resume the previous evolution. This is depicted in Figure 6.2. The combination of these two algorithms is expected to retain their qualities. This algorithm inspired the ones described in Subsections 6.1.3 to 6.1.5. In the remaining of this document we shall refer to it as HM.

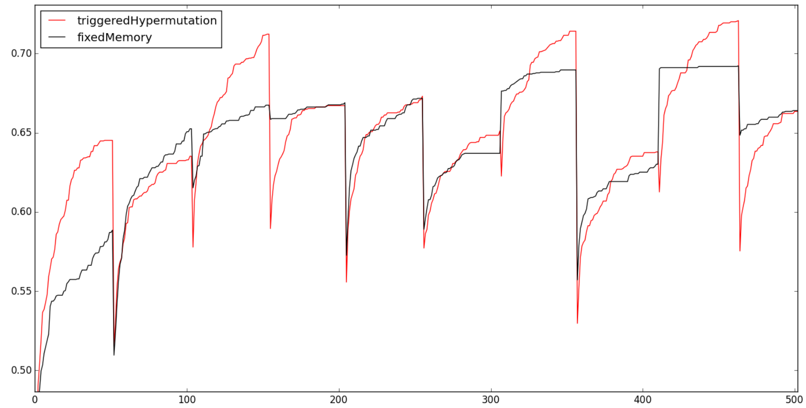


FIGURE 6.1: Average performance of the Best Individual of the components of the HM algorithm, in the beginning of the run in the hard scenario of the Santa Fe Ant Trail benchmark problem.

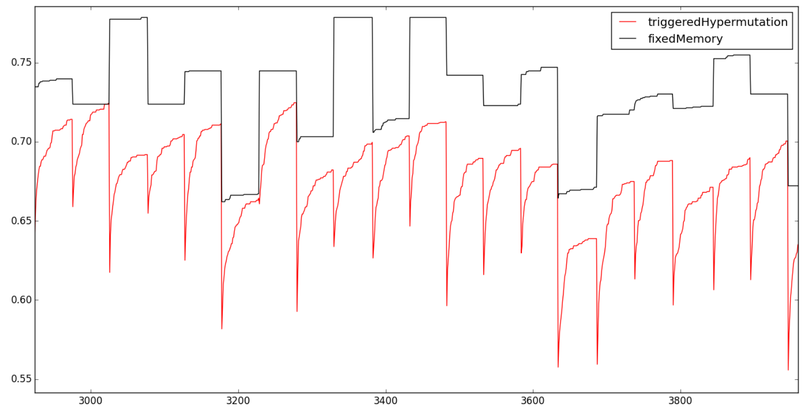


FIGURE 6.2: Average performance of the Best Individual of the components of the HM algorithm, in the middle of the run in the hard scenario of the Santa Fe Ant Trail benchmark problem.

6.1.3 Transformation Memory

This algorithm consists in the combination of the Transformation and Fixed Memory, as defined in Subsections 6.1.1 and 5.1.4, respectively. Thus, it is similar to the Fixed

Memory algorithm, being the differences in their behaviour due to different variation operators. In the remaining of this document we shall refer to this algorithm as TM.

6.1.4 Immigrants Memory

As the previous two algorithms, this variant is the combination of Immigrants and Fixed Memory, with equal parameters to those of the simple algorithms, defined in Subsections 5.1.3 and 5.1.4, respectively. In the remaining of this document we shall refer to this algorithm as IM.

6.1.5 Random Immigrants Memory

This algorithm is slightly different from the previous variants. Instead of combining two existing algorithms unchanged, it combines the Fixed Memory with a different version of the Immigrants algorithm, where no elitist immigrants are created. The same total number of immigrant individuals are created as in the original algorithm, but they are all randomly generated. The parameters are equal to those of the algorithms defined in Subsections 5.1.3 and 5.1.4.

This algorithm was developed under the theory that in cases where the new environment is very different from the previous, the optimum solution would also be very different. Thus, the random immigrants should be more beneficial than the elitist. Furthermore, the variation operators applied to the individuals may create elitist immigrants, considering that the odds allow for only the mutation operator to be applied. In the remaining of this document we shall refer to this algorithm as RIM.

6.1.6 Hypermutation Memory, Transformation and Standard GP

This algorithm evolves three sub-populations, each with its own technique. The first sub-population is evolved with the combination of Triggered Hypermutation and Fixed Memory, as described in Subsection 6.1.2, The second sub-population is evolved with Transformation, described in Subsection 6.1.1, and the Third sub-population is evolved with the Standard GP, described in Subsection 5.1.1. The reason for using this three approaches is to maximize the exploration and exploitation capabilities of the algorithm.

Figures 6.3 and 6.4 depict the average fitness of the best individual of its components, over the 30 runs, in the beginning and middle of those runs. In these Figures it is clear that the Hypermutation Memory is more disruptive than the SGP and the Transformation, thus being better for exploration, while these latter algorithms are better for exploitation. The Transformation is usually slower to regain quality than the other algorithms, however, in some situations it achieves better performance than the SGP before the environment changes and, in two situations, it even outperforms the HM. While in the beginning of the run the use of memory provides no substancial gains, in the middle the same can not be said. For these reasons, we chose to combine these techniques into a single algorithm.

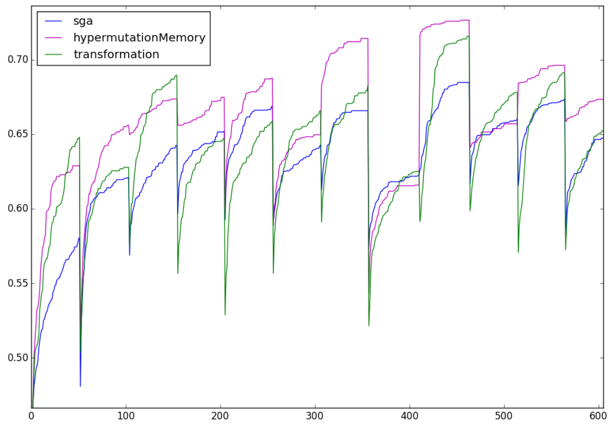


FIGURE 6.3: Average performance of the Best Individual of the components of the HMTS algorithm, in the beginning of the run in the hard scenario of the Santa Fe Ant Trail benchmark problem.

The first two sub-populations are evolved in isolation. The only exception is when updating the memory, where the individual to be stored is the best from the three sub-populations. The third sub-population is not evolved in isolation from the other two. In fact, when creating the offspring, one individual from each sub-population is selected, by means of tournament, to be a candidate for parent. The two parents are the best individuals from the previous three. The other exception to the isolation of the sub-populations is the output of the algorithm, as the solution yielded at each generation is the best individual from the three sub-populations. The generic pseudo code of this hybrid technique is represented in Agorithm 3, being Hypermutation Memory the *algA*, Transformation the *algB* and SGP the *algC*.

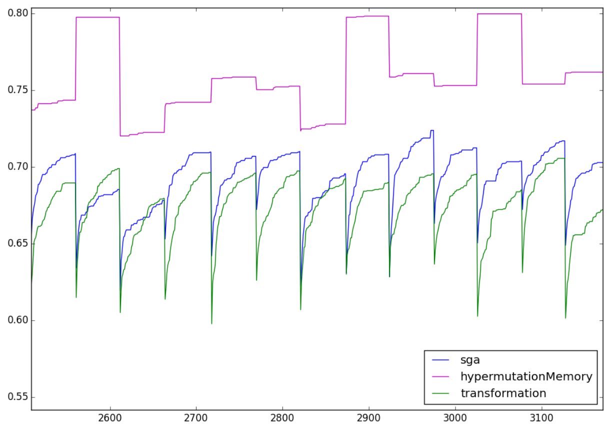


FIGURE 6.4: Average performance of the Best Individual of the components of the HMTS algorithm, in the middle of the run in the hard scenario of the Santa Fe Ant Trail benchmark problem.

The parameters for this algorithm are equal to the ones used by the simple techniques, with the exception of the memory size and gene pool size, that are percentages of the sizes of the sub-populations, instead of being percentages of the total number of individuals. In the remaining of this document we shall refer to this algorithm as HMTS.

Algorithm 3 Three-component GP generic algorithm

Require: $pop, nGens, algA, algB, algC$
Ensure: *bestIndividual*
  $subPopA, subPopB, subPopC \leftarrow splitPop(pop)$
  while $currGen \leqslant nGens$ **do**
    $subPopA \leftarrow isolatedGen(subPopA, algA)$
    $subPopB \leftarrow isolatedGen(subPopB, algB)$
    $parentsC \leftarrow selectParents(subPopA, subPopB, subPopC)$
    $offspring \leftarrow variation(parentsC)$
    $offspringFitness \leftarrow evaluate(offspring)$
    $subPopC \leftarrow survivorsSelection(subPopC, offspring, offspringFitness)$
    $bestIndividual \leftarrow selectMostFit(subPopA, subPopB, subPopC)$
  end while
  return *bestIndividual*

Where *algA*, *algB* and *algC* are the three algorithms to be used in *subPopA*, *subPopB*, *subPopC*, i.e., the three sub-populations. The function *isolatedGen* makes a full generation of the algorithm in isolation.

6.1.7 Transformation Memory, triggered Hypermutation and Standard GP

This algorithm is similar to the one from the previous Subsection. The difference is in the algorithms that evolve each sub-population. In this case, the Fixed Memory is combined with the Transformation, as described in Subsection 6.1.3, to evolve the first sub-population. The second sub-population is evolved with Triggered Hypermutation, while the third sub-population uses the Standard GP, described in Subsections 5.1.2 and 5.1.1, respectively. The parameters for this algorithm are equal to the ones used by the simple techniques, with the exception of the memory size and gene pool size, that are percentages of the sizes of the sub-populations, instead of being percentages of the total number of individuals. In the remaining of this document we shall refer to this algorithm as TMHS.

6.1.8 Fixed Memory, Transformation and Triggered Hypermutation

The last algorithm is similar to the previous two techniques. It evolves three equal sized sub-populations, each with its own algorithm. The first one is evolved with Fixed Memory, as defined in Subsection 5.1.4. The second and third sub-populations are evolved with Transformation and Triggered Hypermutation, respectively from Subsections 6.1.1 and 5.1.2. In the remaining of this document we shall refer to this algorithm as FMTH.

6.2 Normality Tests

Simmilarly to the Simple techniques, the Kolmogorov-Smirnov test determines that the data collected from all algorithms follows non normal distributions, hence the usage of the non parametric tests of Friedman's Anova and Wilcoxon Signed Ranks, depending on the number of algorithms to compare simultaneously. Further discussion on this tests is present in Section 4.5 of Chapter 4. The results of this tests for the Symbolic Regression, Classification and Santa Fe Ant Trail benchmark problems are available in tables A.7 to A.12, A.41 to A.46 and A.75 to A.80, respectively. Furthermore, the results of the descriptive statistics of the Symbolic Regression, Classification and Santa Fe Ant

Trail benchmark problems are present in tables A.1 to A.6 ,A.35 to A.40 and A.69 to A.74, respectively.

6.3 Comparing Algorithms

In this Section we compare and statistically analyse the performance of the Hybrid Techniques with the algorithms that constitute them. The exception is the comparison of the Transformation which is compared with the four Simple Techniques because, it is not composed by simpler techniques, but rather an approach similar to the SGP with a variation operator that has not yet been used in GPs.

Before moving to the statistical analysis, we present the performance of the HM (Figures 6.5 and 6.6) and HMTS (Figures 6.7 and 6.8), when compared to their components. We only present these graphs as a complement to Figures 6.1 to 6.4. In fact, despite being good visual aids for explaining the algorithms and assessing their quality, they do not provide us with a solid and concrete analysis. For that reason, we present the statistical analysis of their results in the following Sections. Nevertheless, as hypothesised in Sections 6.1.2 and 6.1.6, the combination of those algorithms seem to produce better techniques.

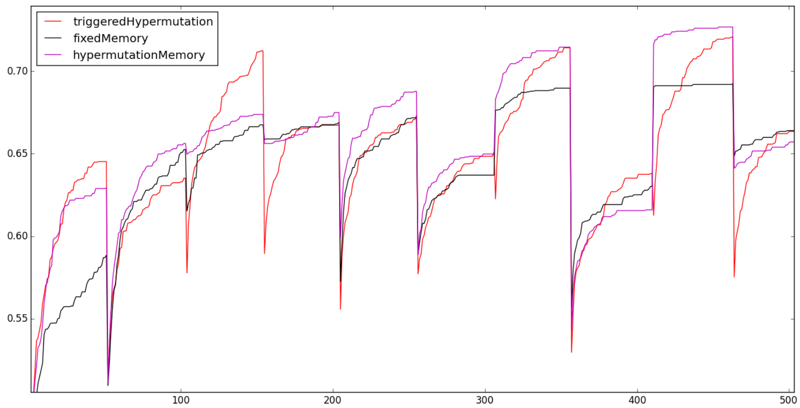


FIGURE 6.5: Average performance of the Best Individual of the components of the HM algorithm, in the beginning of the run in the hard scenario of the Santa Fe Ant Trail benchmark problem.

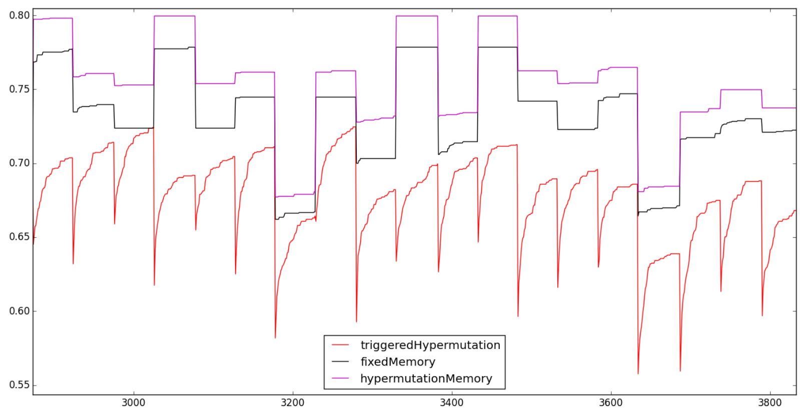


FIGURE 6.6: Average performance of the Best Individual of the components of the HM algorithm, in the middle of the run in the hard scenario of the Santa Fe Ant Trail benchmark problem.

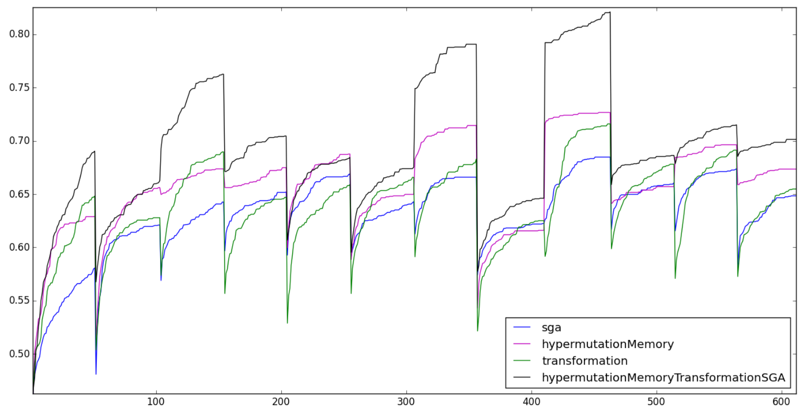


FIGURE 6.7: Average performance of the Best Individual of the components of the HMTS algorithm, in the beginning of the run in the hard scenario of the Santa Fe Ant Trail benchmark problem.

6.3.1 Symbolic Regression

In this Subsection we discuss the performance of the Hybrid Techniques in the three scenarios of the Symbolic Regression benchmark problem.

6.3.1.1 Offline Performance

When measuring the algorithms' performance with the Offline Performance metric, the Friedman's Anova detected statistically significant differences within all groups of algorithms, except, in the easy scenario, when comparing the Hypermutation Memory with

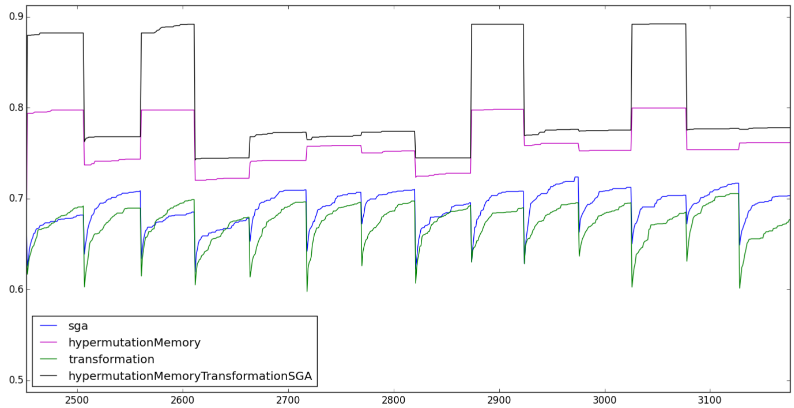


FIGURE 6.8: Average performance of the Best Individual of the components of the HMTS algorithm, in the middle of the run in the hard scenario of the Santa Fe Ant Trail benchmark problem.

the Triggered Hypermutation and Fixed Memory and between the Immigrants Memory, Immigrants and Fixed Memory.

The application of the Wilcoxon Signed Ranks test yielded the following results:

- Easy Scenario
  - The Transformation is worse than the Triggered Hypermutation and the Immigrants;
  - The Immigrants Memory is worse than the Immigrants;
  - The Random Immigrants Memory is worse than the Immigrants;
  - The Transformation Memory is worse than Fixed Memory;
  - The HMTS is better than SGP, Hypermutation Memory and Transformation;
  - The TMHS is better than Transformation Memory;
  - The FMTH is better than Transformation;
- Medium Scenario
  - The Transformation is worse than SGP, Triggered Hypermutation, Immigrants and Fixed Memory;
  - The Hypermutation Memory is better than Triggered Hypermutation;
  - The Immigrants Memory is better than the Immigrants;

  - The Random Immigrants Memory is worse than Immigrants Memory, Immigrants and Fixed Memory;
  - The Transformation Memory is better than Transformation but worse than Fixed Memory;
  - The HTMS is better than Transformation;
  - The TMHS is better than SGP and Transformation Memory;
  - The FMTH is better than Transformation;

- Hard Scenario
  - The Transformation is worse than SGP, Triggered Hypermutation, Immigrants and Fixed Memory;
  - The Hypermutation Memory is better than Triggered Hypermutation and Fixed Memory;
  - The Immigrants Memory is better than the Immigrants;
  - The Random Immigrants Memory is worse than Immigrants Memory, Immigrants and Fixed Memory;
  - The Transformation Memory is better than Transformation;
  - The HMTS is better than SGP and Transformation;
  - The TMHS is better than SGP, Transformation Memory and Triggered Hypermutation;
  - The FMTH is better than Transformation;

The results of this tests can be seen in tables A.21 to A.24.

6.3.1.2 Best of Generation

Using the Best of Generation metric to assess the quality of the algorithms, the Friedman's Anova determined that there were statistically significant differences when comparing all groups of algorithms, except when comparing the Hypermutation Memory with Triggered Hypermutation and Fixed Memory, the Immigrants Memory, with Immigrants and Fixed Memory and the Transformation Memory with Fixed Memory and Transformation, in the easy scenario.

The Wilcoxon Signed Ranks test yielded the following statistically significant results:

- Easy Scenario
  - The Transformation is worse than the Immigrants;
  - The Random Immigrants Memory is worse than the Immigrants and Fixed Memory;
  - The HTMS is better than SGP and Transformation;
  - The TMHS is better than SGP and Transformation Memory;
  - The FMTH is better than the Transformation and Triggered Hypermutation;
- Medium Scenario
  - The Transformation is worse than the SGP, the Triggered Hypermutation, Immigrants and Fixed Memory;
  - The Hypermutation Memory is better than Triggered Hypermutation;
  - The Immigrants Memory is better than Immigrants;
  - The Random Immigrants Memory is worse than Immigrants Memory, Immigrants and Fixed Memory;
  - The Transformation Memory is worse than Fixed Memory, but better than Transformation;
  - The HTMS is better than SGP and Transformation;
  - The TMHS is better than SGP, Transformation Memory and Triggered Hypermutation;
  - The FMTH is better than Transformation and Triggered Hypermutation;
- Hard Scenario
  - The Transformation is worse than the SGP, the Triggered Hypermutation, Immigrants and Fixed Memory;
  - The Hypermutation Memory is better than Triggered Hypermutation;
  - The Immigrants Memory is better than Immigrants;
  - The Random Immigrants Memory is worse than Immigrants Memory, Immigrants and Fixed Memory;
  - The Transformation Memory is better than Transformation;
  - The HTMS is better than SGP and Transformation;

  - The TMHS is better than SGP, and Triggered Hypermutation;
  - The FMTH is better than Transformation and Triggered Hypermutation;

The results of this tests can be seen in tables A.25 to A.28.

6.3.2 Classification

6.3.2.1 Offline Performance

As in the simple techniques, we start off by testing if there are differences in the groups of algorithms, using the Friedman's Anova. The results for this test can be seen in table A.55.

In the easy scenario, the Friedman's Anova detected no statistically significant differences only when comparing the Hypermutation Memory with Triggered Hypermutation and Fixed Memory, and the TMHS with SGP, Transformation Memory and Triggered Hypermutation. In the medium and hard scenarios this test found significant differences within all groups of algorithms.

By applying the Wilcoxon Signed Ranks test, we obtain the following statistically significant results:

- Easy scenario
  - The Transformation is better than the Fixed Memory;
  - The Immigrants Memory algorithm is worse than the Immigrants;
  - The Random Immigrants Memory is worse than the Immigrants;
  - The Transformation Memory is better than the Fixed Memory;
  - The HMTS is better than the SGP and Hypermutation Memory;
  - The FMTH is better than the Fixed Memory;
- Medium Scenario
  - The transformation is better than the SGP, but worse than the Immigrants;
  - The Hypermutation Memory is better than Triggered Hypermutation and Fixed Memory;

  - The Immigrants Memory is worse than the Immigrants and Fixed Memory;
  - The Random Immigrants Memory is worse than the Immigrants;
  - The Transformation Memory is better than the Fixed Memory and Transformation;
  - The HMTS is better than SGP and Transformation;
  - The TMHS is better than the SGP and Triggered Hypermutation;
  - The FMTH is better than Triggered Hypermutation and Transformation;

- Hard Scenario
  - The Transformation is better than the SGP but worse than the Fixed Memory;
  - The Hypermutation Memory is better than the Triggered Hypermutation and Fixed Memory;
  - The Immigrants Memory algorithm is better than the Immigrants but worse than Fixed Memory;
  - The Random Immigrants Memory is worse than the Fixed Memory but better than the Immigrants;
  - The Transformation Memory is better than the Transformation and Fixed Memory;
  - The HMTS is better than the SGP and Transformation;
  - The TMHS is better than the SGP and Triggered Hypermutation;
  - The FMTH is better than Triggered Hypermutation and Transformation;

6.3.2.2 Best of Generation

Here we compare the performance of the Hybrid Techniques with their simpler counterparts, when measured with the Best of Generation metric.

The Friedman's Anova test revealed that there are only no differences when comparing the TMHS with the SGP, Transformation Memory and Triggered Hypermutation, in the easy scenario. Furthermore, in the medium scenario, there are also no significant differences when comparing the FMTH with Fixed Memory, Triggered Hypermutation and Transformation. In the hardest scenario, the Friedman's Anova indicates that there are differences within all groups of algorithms.

We then proceeded to compare the pairs of algorithms with the Wilcoxon Signed Ranks test. In the easiest scenario, it yielded the following statistically significant conclusions:

- The Transformation is better than Fixed Memory;
- The Immigrants Memory is worse than the Immigrants;
- The Random Immigrants Memory is worse than the Immigrants;
- The Transformation Memory is better than Fixed Memory;
- The HMTS is better than SGP and Hypermutation Memory;
- The FMTH is better than Fixed Memory;

In the medium scenario, the statistically significant conclusions that we can draw from this test are:

- The Transformation is better than SGP;
- The Hypermutation Memory is better than Fixed Memory;
- The Immigrants Memory is worse than Immigrants and Fixed Memory;
- The Random Immigrants Memory is worse than the Immigrants;
- Transformation Memory is better than Fixed Memory;
- The HMTS is better than SGP;
- The TMHS is better than SGP;
- The FMTH is better than Fixed Memory;

Finally, in the hardest scenario, the statistically significant results are:

- The Transformation is better than SGP but worse than Triggered Hypermutation;

- The Hypermutation Memory is better than Triggered Hypermutation and Fixed Memory;
- The Immigrants Memory is worse than Immigrants and Fixed Memory;
- The Random Immigrants Memory is worse than Immigrants Memory, Immigrants and Fixed Memory;
- The Transformation Memory is better than Fixed Memory and Transformation;
- The HMTS is better than SGP and Transformation but worse than Hypermutation Memory;
- The TMHS is better than SGP;
- The FMTH is better than Transformation;

As we hypothesized, the Transformation is able to outperform the SGP in some situations.

6.3.3 Santa Fe Ant Trail

In this Subsection we discuss the statistical analysis of the algorithms' performances in the three scenarios of the Santa Fe Ant Trail.

6.3.3.1 Offline Performance

Here we discuss the results of the statistical analysis made to the performance of the algortihms, when measured with the Offline Performance. By the application of the Friedman's Anova we determined that, in the easy scenario, there are statistically signifficant differences when comparing:

- Random Immigrants Memory, Immigrants Memory, Immigrants and Fixed Memory;
- HMTS, SGP, Hypermutation Memory and Transformation;
- TMHS, SGP, Transformation Memory and Triggered Hypermutation;

In the medium and the hard scenarios, this test found significant differences within all groups of algorithms.

The Wilcoxon Signed Ranks test revealed that the differences found by the Friedman's Anova in the easy scenario are due to:

- The Random Immigrants Memory being better than the Immigrants Memory, the Immigrants and the Fixed Memory;
- The HTMS being better than SGP and Hypermutation Memory;
- The TMHS being better than the SGP;

In the medium scenario there were differences within all groups of algorithms. The Wilcoxon Signed ranks test revealed that those differences are due to:

- The Transformation being worse than Fixed Memory;
- The Hypermutation Memory being better than Triggered Hypermutation;
- The Immigrants Memory being better than Immigrants ;
- The Random Immigrants Memory being better than Immigrants Memory, Immigrants and Fixed Memory;
- The Transformation Memory being better than Transformation;
- The HMTS being better than SGP, Hypermutation Memory and Transformation;
- The TMHS being better than SGP and Triggered Hypermutation;
- The FMTH being better than Triggered Hypermutation and Transformation;

Finally, in the hard scenario, this test found the following statistically significant differences:

- The Transformation is worse than SGP, Immigrants and Fixed Memory;
- The Hypermutation Memory is better than Triggered Hypermutation;
- The Immigrants Memory is better than Immigrants;

- The Random Immigrants Memory is better than Immigrants;
- The Transformation Memory is better than Transformation;
- The HMTS is better than SGP, Hypermutation Memory, and Transformation;
- The TMHS is better than SGP, Transformation Memory and Triggered Hypermutation;
- The FMTH is better than Fixed Memory, Triggered Hypermutation and Transformation;

The results of the statistical tests can be seen in tables A.89 to A.92.

6.3.3.2 Best of Generation

We shall now discuss the results of the analysis of the algorithms' performance, measured with the Best of Generation metric. As in the previous Sections, we start by applying the Friedman's Anova. This test determines that, in the easy scenario, there were only statistically significant differences between the Random Immigrants Memory, Immigrants, Immigrants Memory and Fixed Memory, the HMTS, SGP, Hypermutation Memory and Transformation, and between the TMHS, SGP, Transformation Memory and Triggered Hypermutation. In the medium and hard scenarios there were differences within all groups of algorithms. As expected, as the complexity of the scenarios increases, the more different the algorithms perform.

The Wilcoxon Signed Ranks test confirms the results of the Friedman's Anova, only finding differences in the easy scenario, between the Random Immigrants Memory, Immigrants, Immigrants Memory and Fixed Memory, between the TMHS, SGP, Transformation Memory and Triggered Hypermutation and the FMTH, Fixed Memory, Triggered Hypermutation and Transformation. In the medium and hard scenarios, there are differences in all groups of algorithms.

In the Easy, the Wilcoxon Signed Ranks test determined that the differences are due to:

- The Random Immigrants Memory being better than the Immigrants Memory, Immigrants and Fixed Memory;

- The HMTS being better than SGP and Hypermutation Memory;
- The TMHS being better than the SGP;

In the medium scenario, the Wilcoxon Signed Ranks test determined the following statistically significant differences:

- The Transformation is worse than Fixed Memory;
- The Hypermutation Memory is better than Triggered Hypermutation;
- The Immigrants Memory is better than Immigrants;
- The Random Immigrants Memory is better than Immigrants Memory, Immigrants and Fixed Memory;
- The Transformation Memory is better than Transformation;
- The HMTS is better than SGP, Hypermutation Memory and Transformation;
- The TMHS is better than SGP and Triggered Hypermutation;
- The FMTH is better than Triggered Hypermutation and Transformation;

Finally, in the hard scenario, this test determined the following statistically significant differences:

- The Transformation is worse than Fixed Memory;
- The Hypermutation Memory is better than Triggered Hypermutation;
- The Immigrants Memory is better than Immigrants;
- The Random Immigrants Memory is better than Immigrants;
- The Transformation Memory is better than Transformation;
- The HMTS is better than SGP, Hypermutation Memory and Transformation;
- The TMHS is better than SGP, Transformation Memory and Triggered Hypermutation;

- The FMTH is better than Fixed Memory, Triggered Hypermutation and Transformation;

The results of this tests can be seen in tables A.93 to A.96.

6.4 Search for the Best Algorithm

In this section we discuss the results obtained per benchmark problem, fitness metric and difficulty level, with the goal of discovering if there is a best algorithm for each case.

6.4.1 Symbolic Regression

In this Subsection we discuss the comparison between all algorithms in the Symbolic Regression Benchmark Problem.

6.4.1.1 Offline Performance

The Friedman's Anova test determined that there are differences between the algorithms in the three scenarios, yielding values of Chi-Square probability and p-value of 57.0769230769 and 3.22052233678e-08, 122.251282051 and 6.38621002694e-21, 177.435897436 and 3.46591083246e-32, for the easy, medium and hard scenarios, respectively.

The application of the Wilcoxon Signed Ranks test lead to the following conclusions:

- Easy Scenario
  - The Immigrants is better than Transformation Memory;
  - The Transformation is worse than HMTS;
  - The Random Immigrants Memory is worse than HMTS and FMTH;
  - The Transformation Memory is worse than HMTS, TMHS and FMTH;
- Medium Scenario
  - The SGP is better than Transformation and Random Immigrants Memory;

  - The Triggered Hypermutation is better than the Transformation and the Random Immigrants Memory;
  - The Immigrants is worse than Hypermutation Memory;
  - The Fixed Memory is better than Transformation and Random Immigrants Memory;
  - The Transformation is worse than Hypermutation Memory, Immigrants Memory, HMTS, TMHS and FMTH;
  - The Hypermutation Memory is better than Random Immigrants Memory and Transformation Memory;
  - The Immigrants Memory is better than Random Immigrants Memory;
  - The Random Immigrants Memory is worse than HMTS, TMHS and FMTH;
  - The Transformation Memory is worse than HMTS, TMHS and FMTH;

- Hard Scenario
  - The SGP is better than Transformation and Random Immigrants Memory, but worse than Hypermutation Memory, HMTS and TMHS;
  - The Triggered Hypermutation is better than Transformation and Random Immigrants Memory, but worse than Hypermutation Memory;
  - The Immigrants is worse than Hypermutation Memory, Immigrants Memory, HMTS, TMHS and FMTH, but better than Random Immigrants Memory;
  - The Fixed Memory is better than Transformation and Random Immigrants Memory;
  - The Transformation is worse than Hypermutation Memory, Immigrants Memory, Transformation Memory, HMTS, TMHS and FMTH but better than Random Immigrants Memory;
  - The Hypermutation Memory is better than Random Immigrants Memory and Transformation Memory;
  - The Immigrants Memory is better than Random Immigrants Memory;
  - The Random Immigrants Memory is worse than Transformation Memory, HMTS, TMHS and FMTH;

The results of this tests can be seen in tables A.29 to A.31.

6.4.1.2 Best of Generation

When measuring the algorithms' performances with the Best of Generation metric, the Friedman's Anova test determined to be differences between the algorithms in the three scenarios, having values of Chi-Square probability and p-value of 72.0153846154 and 5.0480329586e-11, 163.441025641 and 2.63114468615e-29, 193.194871795 and 1.91534860629e-35, for the easy, medium and hard scenarios, respectively.

The Wilcoxon Signed Ranks test determined that there were the following statistically significant differences between the algorithms:

- Easy Scenario
  - The Transformation is worse than HMTS, TMHS and FMTH;
  - The Hypermutation Memory is better than Random Immigrants Memory;
  - The Random Immigrants Memory is worse than HMTS, TMHS and FMTH;
  - The Transformation Memory is worse than HMTS and TMHS;
- Medium Scenario
  - The SGP is worse than Fixed Memory, Hypermutation Memory and TMHS, but better than Transformation and Random Immigrants Memory;
  - The Triggered Hypermutation is better than Transformation and Random Immigrants Memory but worse then Hypermutation Memory;
  - The Immigrants is worse than Fixed Memory, Hypermutation Memory, HMTS, TMHS and FMTH, but better than Transformation and Random Immigrants Memory;
  - The Fixed Memory is better than Transformation, Random Immigrants Memory and Transformation Memory;
  - The Transformation is worse than Hypermutation Memory, Immigrants Memory, Transformation Memory, HMTS, TMHS and FMTH;
  - The Hypermutation Memory is better than Random Immigrants Memory and Transformation Memory;
  - The Immigrants Memory is better than Random Immigrants Memory;

  - The Random Immigrants Memory is worse than Transformation Memory, HMTS, TMHS and FMTH;

- Hard Scenario

  - The SGP is better than Transformation and Random Immigrants Memory, but worse than Hypermutation Memory, Immigrants Memory, HMTS, TMHS and FMTH;
  - The Triggered Hypermutation is better than Transformation and Random Immigrants Memory, but worse than Hypermutation Memory and HMTS;
  - The Immigrants is worse than Fixed Memory, Hypermutation Memory, Immigrants Memory, HMTS, TMHS and FMTH, but better than Random Immigrants Memory;
  - The Fixed Memory is better than Transformation and Random Immigrants Memory;
  - The Transformation is worse than Hypermutation Memory, Immigrants Memory, Transformation Memory, HMTS, TMHS and FMTH, but better than Random Immigrants Memory;
  - The Hypermutation Memory is better than Random Immigrants Memory and Transformation Memory;
  - The Immigrants Memory is better than Random Immigrants Memory;
  - The Random Immigrants Memory is worse than Transformation Memory, HMTS, TMHS and FMTH;

The results of this tests can be seen in tables A.32 to A.34.

6.4.2 Classification

In this Subsection we discuss the comparison of the performances of all algorithms, measured with the Offline Performance and Best of Generation metrics, in the Classification benchmark problem.

6.4.2.1 Offline Performance

The application of the Friedman's Anova test determined that there are differences between the algorithms in the three scenarios, as it yielded values of Chi-Square probability and p-value of 67.3028678014 and 3.96495697474e-10, 94.3384615385 and 2.34311937337e-15, 264.230769231 and 2.90658658064e-50, for the easy, medium and hard scenarios, respectively.

In the easiest scenario only a few algorithms distinguished themselves from the others, as can been seen in table A.63. However, we were able to draw the following conclusions:

- The Immigrants based algorithm is better than Immigrants Memory and Random Immigrants Memory algorithms;
- The Fixed Memory algorithm is worse than the HMTS and the TMHS;
- The Transformation is better than the Random Immigrants Memory;
- The Immigrants Memory is worse than the Transformation Memory and HMTS;
- The Random Immigrants Memory algorithm is worse than Transformation Memory, HMTS, TMHS and FMTH;

The results from the medium scenario are depicted in table A.64. In it, can be seen that there were some more differences between the algorithms. This is expected, as the difficulty increases, the less powerful algorithms tend to fall behind the others. We can draw the following statistically significant conclusions:

- The SGP is worse than Immigrants, Fixed Memory, Hypermutation Memory, Transformation Memory, HMTS, TMHS and FMTH;
- The Triggered Hypermutation is worse than Transformation Memory;
- The Immigrants is better than Immigrants Memory;
- The Transformation is worse than Transformation Memory, HMTS, TMHS and FMTH;
- The Hypermutation Memory is better than Immigrants Memory;

- The Immigrants Memory is worse than Transformation Memory, HMTS, TMHS and FMTH;
- The Random Immigrants Memory is worse than TMHS;

The results from the hard scenario are depicted in table A.65. In this scenario, the difference between the algorithms are:

- The SGP is worse than the Immigrants, Fixed Memory, Hypermutation Memory, Immigrants Memory, Random Immigrants Memory, Transformation Memory, HMTS, TMHS, FMTH;
- The Triggered Hypermutation is worse than Fixed Memory, Hypermutation Memory, Immigrants Memory, Ransom Immigrants Memory, Transformation Memory, HMTS, TMHS, FMTH;
- The Immigrants approach is worse than Fixed Memory, Hypermutation Memory, Immigrants Memory, Random Immigrants Memory, Transformation Memory, HMTS, TMHS, FMTH;
- The Fixed Memory is better than the Transformation, Immigrants Memory and Random Immigrants Memory;
- The Transformation is worse than Hypermutation Memory, Immigrants Memory, Random Immigrants Memory, Transformation Memory, HMTS, TMHS, FMTH;
- The Hypermutation Memory is better than Immigrants Memory and Random Immigrants Memory;
- The Immigrants Memory is worse than the Transformation Memory, HMTS, TMHS, FMTH;
- The Random Immigrants Memory is worse than the Transformation Memory, HMTS, TMHS, FMTH;

6.4.2.2 Best of Generation

In this Subsection we discuss the results of the statistical analysis of the algorithms' performances in the Classification problem, using the Best of Generation metric. The Friedman's Anova found statistically significant differences between all algorithms in the three scenarios, as it yielded values of Chi-Square probability and p-value of 80.6162275589 and 1.12148764853e-12, 88.6 and 3.13376489045e-14, 197.148717949 and 2.9030704667e-36, for the easy, medium and hard scenarios, respectively.

The data for the Wilcoxon test in the easy, medium and hard scenarios is represented in tables A.66, A.67 and A.68, respectively.

As for the Offline Performance, there were not many differences between the algorithms in the easiest scenario for the Best of Generation. Nevertheless, the statistically significant differences were:

- The SGP is better than the Random Immigrants Memory;
- The Triggered Hypermutation is better than the Random Immigrants Memory;
- The Immigrants based algorithm is better than Fixed Memory, Immigrants Memory and Random Immigrants Memory;
- The Fixed Memory is worse than Transformation, HMTS and TMHS;
- The Transformation is better than Immigrants Memory and Random Immigrants Memory;
- The Immigrants Memory is worse than HMTS and TMHS;
- The Random Immigrants Memory is worse than Transformation Memory, HMTS, TMHS and FMTH;

In the medium scenario there were some more differences between the algorithms. The statistically signifficant conclusions that we can draw are:

- The SGP is worse than Triggered Hypermutation, Immigrants, Transformation, Hypermutation Memory, Transformation Memory, HMTS, TMHS and FMTH;
- The Triggered Hypermutation is better than Immigrants Memory;
- The Immigrants based algorithm is better than Fixed Memory, Immigrants Memory and Random Immigrants Memory;
- The Transformation is better than Immigrants Memory;

- The Hypermutation Memory is better than Immigrants Memory;
- The Immigrants Memory algorithm is worse than Transformation Memory, HMTS, TMHS and FMTH;
- The Random Immigrants Memory is worse than Transformation Memory, TMHS and FMTH;

In the hardest scenario we are able to identify more differences between the algorithms. The statistically significant conclusions are:

- The SGP is worse than every other algorithm, with the exception of the Immigrants Memory and the Random Immigrants Memory;
- The SGP is better than the Random Immigrants Memory;
- The Triggered Hypermutation is worse than Hypermutation Memory but better than Immigrants Memory and Random Immigrants Memory;
- The Immigrants algorithm is worse than Hypermutation Memory and HMTS, but better than Immigrants Memory and Random Immigrants Memory;
- The Fixed Memory is worse than Hypermutation Memory but better than Immigrants Memory and Random Immigrants Memory;
- The Transformation is worse than Hypermutation Memory, Transformation Memory, HMTS and TMHS but better than Immigrants Memory and Random Immigrants Memory;
- The Hypermutation Memory is better than Immigrants Memory, Random Immigrants Memory and FMTH;
- The Immigrants Memory is worse than Transformation Memory, HMTS, TMHS and FMTH;
- The Random Immigrants Memory is worse than Transformation Memory, HMTS, TMHS and FMTH;

6.4.3 Santa Fe Ant Trail

In this Subsection we discuss the comparison between all algorithms' performances in the Santa Fe Ant Trail benchmark Problem.

6.4.3.1 Offline Performance

The Friedman's Anova detected statistically significant differences between the algorithms in the three scenarios, yielding Chi-Square and p-values of 56.1689023679 and 4.73110286277e-08, 203.748717949 and 1.2398860755e-37, 255.574358974 and 1.89894340749e-48, in the easy, medium and hard scenarios, respectively.

The data for the Wilcoxon Signed Ranks test in the easy, medium and hard scenarios is present in tables A.97 to A.99. It yielded the following statistically signifficant results:

- Easy Scenario
  - The Random Immigrants Memory being better than SGP, Triggered Hypermutation, Immigrants, Fixed Memory, Hypermutation Memory and Transformation Memory;
  - The SGP being worse than HMTS and TMHS;
- Medium Scenario
  - The SGP being worse than Fixed Memory, Hypermutation Memory, Immigrants Memory, Random Immigrants Memory, Transformation Memory, HMTS, TMHS, FMTH;
  - The Triggered Hypermutation being worse than Fixed Memory, Hypermutation Memory, Immigrants Memory, Random Immigrants Memory, Transformation Memory, HMTS, TMHS and FMTH;
  - The Immigrants being worse than Fixed Memory, Hypermutation Memory, Immigrants Memory, Random Immigrants Memory, Transformation Memory, HMTS, TMHS and FMTH;
  - The Fixed Memory being better than Transformation, but worse than Random Immigrants Memory;

  - The Transformation being worse than Hypermutation Memory, Immigrants Memory, Random Immigrants Memory, Transformation Memory, HMTS, TMHS and FMTH;
  - The Hypermutation Memory being worse than Random Immgrants Memory;
  - The Immigrants Memory being worse than Random Immigrants memory;

- Hard Scenario
  - The SGP being worse than Fixed Memory, Hypermutation Memory, Immigrants Memory, Random Immigrants Memory, Transformation Memory, HMTS, TMHS and FMTH, but better than Transformation;
  - The Triggered Hypermutation being worse than Fixed Memory, Hypermutation Memory, Immigrants Memory, Random Immigrants Memory, Transformation Memory, HMTS, TMHS and FMTH;
  - The Immigrants being worse than Fixed Memory, Hypermutation Memory, Immigrants Memory, Random Immigrants Memory, Transformation Memory, HMTS, TMHS and FMTH;
  - The Fixed Memory being better than Transformation, but worse than HMTS, TMHS, FMTH;
  - The Transformation being worse than Hypermutation Memory, Immigrants Memory, Random Immigrants Memory, Transformation Memory, HMTS, TMHS and FMTH;
  - The Hypermutation Memory being worse than HMTS;
  - The Immigrants Memory being worse than HMTS, TMHS and FMTH;
  - The Random Immigrants Memory being worse than HMTS, TMHS and FMTH;
  - The Transformation Memory being worse than HMTS and TMHS;

6.4.3.2 Best of Generation

Using the Best of Generation, the Friedman's Anova determined that there are differences between the algorithms' performances in the three scenarios. It yielded values of Chi-Square and p-values of 50.8482340599 and 4.40339531499e-07, 122.81025641 and

4.92750996796e-21, 209.8 and 6.8547816932e-39, for the easy, medium and hard scenarios, respectively.

The data for the Wilcoxon Signed Ranks test is present in tables A.100 to A.102. This test determined that the differences found by the Friedman's Anova are due to:

- Easy Scenario
  - The Random Immigrants Memory being better than SGP, Triggered Hypermutation, Immigrants, Fixed Memory, Hypermutation Memory and Transformation Memory;
  - The HMTS beign better than SGP;
- Medium Scenario
  - The SGP being worse than Random Immigrants Memory, Transformation Memory, HMTS, TMHS and FMTH;
  - The Triggered Hypermutation being worse than Hypermutation Memory, Random Immigrants Memory, Transformation Memory, HMTS, TMHS and FMTH;
  - The Immigrants being worse than Random Immigrants Memory, Transformation Memory, HMTS, TMHS and FMTH;
  - The Fixed Memory being worse than Random Immigrants Memory;
  - The Transformation being worse than Random Immigrants Memory, Transformation Memory, HMTS, TMHS and FMTH;
  - The Hypermutation Memory being worse than Random Immigrants Memory;
  - The Immigrants Memory being worse than Random Immigrants Memory;
- Hard Scenario
  - The SGP being worse than Fixed Memory, Hypermutation Memory, Immigrants Memory, Random Immigrants Memory, Transformation Memory, HMTS, TMHS and FMTH;
  - The Triggered Hypermutation being worse than Fixed Memory, Hypermutation Memory, Immigrants Memory, Random Immigrants Memory, Transformation Memory, HMTS, TMHS and FMTH;

- The Immigrants being worse than Fixed Memory, Hypermutation Memory, Random Immigrants Memory, Transformation Memory, HMTS, TMHS and FMTH;
- The Fixed Memory being better than Transformation, but worse than HMTS, TMHS and FMTH;
- The Transformation being worse than Hypermutation Memory, Immigrants Memory, Random Immigrants Memory, Transformation Memory, HMTS, TMHS and FMTH;
- The Hypermutation Memory being worse than HMTS;
- The Immigrants Memory being worse than HMTS, TMHS and FMTH;
- The Random Immigrants Memory being worse than HMTS, TMHS and FMTH;
- The Transformation Memory being worse than HMTS, TMHS and FMTH;

6.5 Summary

In this Section we resort to a more visual type of comparison to sum up the ones made in the previous Sections, using only the Wilcoxon Signed Ranks test. The symbols in the tables have the same meaning as the ones in Section 5.7.

6.5.1 Comparison of the Hybrid Techniques

As in the Simple Techniques, here there is also no consistency in the performance of the algorithms throughout the benchmark problems. However, we can still draw some conclusions regarding the performance of some algorithms in the three benchmark problems.

When measuring the performance with the Offline Performance, we can say that:

- The Triggered Hypermutation and the Immigrants are never worse than the Transformation;
- The Hypermutation Memory is never worse than Triggered Hypermutation and Fixed Memory;

- The Immigrants Memory is never worse than the Fixed Memory;
- The Transformation Memory is never worse than the Transformation;
- The HMTS is never worse than the SGP, Hypermutation Memory and Transformation;
- The TMHS is never worse than the SGP, Transformation Memory and Triggered Hypermutation;
- The FMTH is never worse than the Fixed Memory, Triggered Hypermutation and the Transformation.

Considering the results with the Best of Generation, we can say that:

- The Triggered Hypermutation and the Immigrants are never worse than the Transformation;
- The Hypermutation Memory is never worse than the Triggered Hypermutation and the Fixed Memory;
- The Fixed Memory is never worse than the Immigrants Memory;
- The Transformation Memory is never worse than the Transformation;
- The HMTS is never worse than the SGP and the Transformation;
- The TMHS is never worse than the SGP, Transformation Memory and Triggered Hypermutation;
- The FMTH is never worse than the Transformation, Fixed Memory and Triggered Hypermutation.

The comparisons between these algorithms are represented in tables 6.1 and 6.2.

	Symbolic Regression			**Classification**			**Santa Fe Ant Trail**		
Algorithms	Easy	Medium	Hard	Easy	Medium	Hard	Easy	Medium	Hard
T vs SGP	●	▼	▼	●	▲	▲	●	●	▼
T vs TH	▼	▼	▼	●	●	●	●	●	●
T vs I	▼	▼	▼	●	▼	●	●	●	▼
T vs FM	●	▼	▼	▲	●	▼	●	▼	▼
HM vs TH	●	▲	▲	●	▲	▲	●	▲	▲

	Symbolic Regression			**Classification**			**Santa Fe Ant Trail**		
Algorithms	Easy	Medium	Hard	Easy	Medium	Hard	Easy	Medium	Hard
HM vs FM	●	●	▲	●	▲	▲	●	●	●
IM vs I	▼	▲	▲	▼	▼	▲	●	▲	▲
IM vs FM	●	●	●	●	▼	▼	●	●	●
RIM vs IM	●	▼	▼	●	●	●	▲	▲	●
RIM vs I	▼	▼	▼	▼	▼	▲	▲	▲	▲
RIM vs FM	●	▼	▼	●	●	▼	▲	▲	●
TM vs FM	▼	▼	●	▲	▲	▲	●	●	●
TM vs T	●	▲	▲	●	▲	▲	●	▲	▲
HMTS vs SGP	▲	●	▲	▲	▲	▲	▲	▲	▲
HMTS vs HM	▲	●	●	▲	●	●	▲	▲	▲
HMTS vs T	▲	▲	▲	●	▲	▲	●	▲	▲
TMHS vs SGP	●	▲	▲	●	▲	▲	▲	▲	▲
TMHS vs TM	▲	▲	▲	●	●	●	●	●	▲
TMHS vs TH	●	●	▲	●	▲	▲	●	▲	▲
FMTH vs FM	●	●	●	▲	●	●	●	●	▲
FMTH vs TH	●	●	●	●	▲	▲	●	▲	▲
FMTH vs T	▲	▲	▲	●	▲	▲	●	▲	▲

TABLE 6.1: Hybrid Techniques - Offline Performance

	Symbolic Regression			**Classification**			**Santa Fe Ant Trail**		
Algorithms	Easy	Medium	Hard	Easy	Medium	Hard	Easy	Medium	Hard
T vs SGP	●	▼	▼	●	▲	▲	●	●	●
T vs TH	●	▼	▼	●	●	▼	●	●	●
T vs I	▼	▼	▼	●	●	●	●	●	●
T vs FM	●	▼	▼	▲	●	●	●	▼	▼
HM vs TH	●	▲	▲	●	●	▲	●	▲	▲
HM vs FM	●	●	●	●	▲	▲	●	●	●
IM vs I	●	▲	▲	▼	▼	▼	●	▲	▲
IM vs FM	●	●	●	●	▼	▼	●	●	●
RIM vs IM	●	▼	▼	●	●	▼	▲	▲	●
RIM vs I	▼	▼	▼	▼	▼	▼	▲	▲	▲
RIM vs FM	▼	▼	▼	●	●	▼	▲	▲	●
TM vs FM	●	▼	●	▲	▲	▲	●	●	●
TM vs T	●	▲	▲	●	●	▲	●	▲	▲
HMTS vs SGP	▲	▲	▲	▲	▲	▲	▲	▲	▲
HMTS vs HM	●	●	●	▲	●	▼	▲	▲	▲
HMTS vs T	▲	▲	▲	●	●	▲	●	▲	▲
TMHS vs SGP	▲	▲	▲	●	▲	▲	▲	▲	▲
TMHS vs TM	▲	▲	●	●	●	●	●	●	▲

	Symbolic Regression			**Classification**			**Santa Fe Ant Trail**		
Algorithms	Easy	Medium	Hard	Easy	Medium	Hard	Easy	Medium	Hard
TMHS vs TH	●	▲	▲	●	●	●	●	▲	▲
FMTH vs FM	●	●	●	▲	▲	●	●	●	▲
FMTH vs TH	▲	▲	▲	●	●	●	●	▲	▲
FMTH vs T	▲	▲	▲	●	●	▲	●	▲	▲

Table 6.2: Hybrid Techniques - Best of Generation

6.5.2 Search for the Best Algorithm

In this Subsection we discuss the results of the comparison of the performances of all algorithms, measured with the Offline Performance and the Best of Generation metrics. Using the Offline Performance, there is no algorithm that is consistently better than all the others in all benchmark problems. However, there are some algorithms that are never worse than any other in a specific benchmark problem, not existing statistically significant differences between them:

- Symbolic Regression
  - Fixed Memory
  - Hypermutation Memory
  - Immigrants Memory
  - HMTS
  - TMHS
  - FMTH
- Classification
  - Hypermutation Memory
  - Transformation Memory
  - HMTS
  - TMHS
  - FMTH
- Santa Fe Ant Trail

  - HMTS
  - TMHS
  - FMTH

By measuring the algorithms' performances with the Best of Generation metric we obtain different results. As in the case of the Offline Performance, here there also are some algorithms that are never inferior to any others, in specific benchmark problems. They are:

- Symbolic Regression
  - Fixed Memory
  - Hypermutation Memory
  - Immigrants Memory
  - HMTS
  - TMHS
  - FMTH
- Classification
  - Hypermutation Memory
  - Transformation Memory
  - HMTS
  - TMHS
- Santa Fe Ant Trail
  - HMTS
  - TMHS
  - FMTH

	Symbolic Regression			**Classification**			**Santa Fe Ant Trail**		
Algorithms	Easy	Medium	Hard	Easy	Medium	Hard	Easy	Medium	Hard
SGP vs TH	●	●	●	●	●	●	●	●	●
SGP vs I	●	●	●	●	▼	▼	●	●	●

	Symbolic Regression			**Classification**			**Santa Fe Ant Trail**		
Algorithms	Easy	Medium	Hard	Easy	Medium	Hard	Easy	Medium	Hard
SGP vs FM	●	●	●	●	▼	▼	●	▼	▼
SGP vs T	●	▲	▲	●	●	●	●	●	▲
SGP vs HM	●	●	▼	●	▼	▼	●	▼	▼
SGP vs IM	●	●	●	●	●	▼	●	▼	▼
SGP vs RIM	●	▲	▲	●	●	▼	▼	▼	▼
SGP vs TM	●	●	●	●	▼	▼	●	▼	▼
SGP vs HMTS	●	●	▼	●	▼	▼	▼	▼	▼
SGP vs TMHS	●	●	▼	●	▼	▼	▼	▼	▼
SGP vs FMTH	●	●	●	●	▼	▼	●	▼	▼
TH vs I	●	●	●	●	●	●	●	●	●
TH vs FM	●	●	●	●	●	▼	●	▼	▼
TH vs T	●	▲	▲	●	●	●	●	●	●
TH vs HM	●	●	▼	●	●	▼	●	▼	▼
TH vs IM	●	●	●	●	●	▼	●	▼	▼
TH vs RIM	●	▲	▲	●	●	▼	▼	▼	▼
TH vs TM	●	●	●	●	▼	▼	●	▼	▼
TH vs HMTS	●	●	●	●	●	▼	●	▼	▼
TH vs TMHS	●	●	●	●	●	▼	●	▼	▼
TH vs FMTH	●	●	●	●	●	▼	●	▼	▼
I vs FM	●	●	●	●	●	▼	●	▼	▼
I vs T	●	●	●	●	●	●	●	●	●
I vs HM	●	▼	▼	●	●	▼	●	▼	▼
I vs IM	●	●	▼	▲	▲	▼	●	▼	▼
I vs RIM	●	●	▲	▲	●	▼	▼	▼	▼
I vs TM	▲	●	●	●	●	▼	●	▼	▼
I vs HMTS	●	●	▼	●	●	▼	●	▼	▼
I vs TMHS	●	●	▼	●	●	▼	●	▼	▼
I vs FMTH	●	●	▼	●	●	▼	●	▼	▼
FM vs T	●	▲	▲	●	●	▲	●	▲	▲
FM vs HM	●	●	●	●	●	●	●	●	●
FM vs IM	●	●	●	●	●	▲	●	●	●
FM vs RIM	●	▲	▲	●	●	▲	▼	▼	●
FM vs TM	●	●	●	●	●	●	●	●	●
FM vs HMTS	●	●	●	▼	●	●	●	●	▼
FM vs TMHS	●	●	●	▼	●	●	●	●	▼
FM vs FMTH	●	●	●	●	●	●	●	●	▼
T vs HM	●	▼	▼	●	●	▼	●	▼	▼
T vs IM	●	▼	▼	●	●	▼	●	▼	▼
T vs RIM	●	●	▲	▲	●	▼	●	▼	▼

	Symbolic Regression			**Classification**			**Santa Fe Ant Trail**		
Algorithms	Easy	Medium	Hard	Easy	Medium	Hard	Easy	Medium	Hard
T vs TM	●	●	▼	●	▼	▼	●	▼	▼
T vs HMTS	▼	▼	▼	●	▼	▼	●	▼	▼
T vs TMHS	●	▼	▼	●	▼	▼	●	▼	▼
T vs FMTH	●	▼	▼	●	▼	▼	●	▼	▼
HM vs IM	●	●	●	●	▲	▲	●	●	●
HM vs RIM	●	▲	▲	●	●	▲	▼	▼	●
HM vs TM	●	▲	▲	●	●	●	●	●	●
HM vs HMTS	●	●	●	●	●	●	●	●	▼
HM vs TMHS	●	●	●	●	●	●	●	●	●
HM vs FMTH	●	●	●	●	●	●	●	●	●
IM vs RIM	●	▲	▲	●	●	●	●	▼	●
IM vs TM	●	●	●	▼	▼	▼	●	●	●
IM vs HMTS	●	●	●	▼	▼	▼	●	●	▼
IM vs TMHS	●	●	●	●	▼	▼	●	●	▼
IM vs FMTH	●	●	●	●	▼	▼	●	●	▼
RIM vs TM	●	●	▼	▼	●	▼	▲	●	●
RIM vs HMTS	▼	▼	▼	▼	●	▼	●	●	▼
RIM vs TMHS	●	▼	▼	▼	▼	▼	●	●	▼
RIM vs FMTH	▼	▼	▼	▼	●	▼	●	●	▼
TM vs HMTS	▼	▼	●	●	●	●	●	●	▼
TM vs TMHS	▼	▼	●	●	●	●	●	●	▼
TM vs FMTH	▼	▼	●	●	●	●	●	●	●
HMTS vs TMHS	●	●	●	●	●	●	●	●	●
HMTS vs FMTH	●	●	●	●	●	●	●	●	●
TMHS vs FMTH	●	●	●	●	●	●	●	●	●

Table 6.3: Search for the Best Algorithm - Offline Performance

	Symbolic Regression			**Classification**			**Santa Fe Ant Trail**		
Algorithms	Easy	Medium	Hard	Easy	Medium	Hard	Easy	Medium	Hard
SGP vs TH	●	●	●	●	▼	▼	●	●	●
SGP vs I	●	●	●	●	▼	▼	●	●	●
SGP vs FM	●	▼	●	●	●	▼	●	●	▼
SGP vs T	●	▲	▲	●	▼	▼	●	●	●
SGP vs HM	●	▼	▼	●	▼	▼	●	●	▼
SGP vs IM	●	●	▼	●	●	●	●	●	▼
SGP vs RIM	●	▲	▲	▲	●	▲	▼	▼	▼
SGP vs TM	●	●	●	●	▼	▼	●	▼	▼
SGP vs HMTS	●	●	▼	●	▼	▼	▼	▼	▼
SGP vs TMHS	●	▼	▼	●	▼	▼	●	▼	▼

	Symbolic Regression			Classification			Santa Fe Ant Trail		
Algorithms	Easy	Medium	Hard	Easy	Medium	Hard	Easy	Medium	Hard
SGP vs FMTH	●	●	▼	●	▼	▼	●	▼	▼
TH vs I	●	●	●	●	●	●	●	●	●
TH vs FM	●	●	●	●	●	●	●	●	▼
TH vs T	●	▲	▲	●	●	●	●	●	●
TH vs HM	●	▼	▼	●	●	▼	●	▼	▼
TH vs IM	●	●	●	●	▲	▲	●	●	▼
TH vs RIM	●	▲	▲	▲	●	▲	▼	▼	▼
TH vs TM	●	●	●	●	●	●	●	▼	▼
TH vs HMTS	●	●	▼	●	●	●	●	▼	▼
TH vs TMHS	●	●	●	●	●	●	●	▼	▼
TH vs FMTH	●	●	●	●	●	●	●	▼	▼
I vs FM	●	▼	▼	▲	▲	●	●	●	▼
I vs T	●	▲	●	●	●	●	●	●	●
I vs HM	●	▼	▼	●	●	▼	●	●	▼
I vs IM	●	●	▼	▲	▲	▲	●	●	●
I vs RIM	●	▲	▲	▲	▲	▲	▼	▼	▼
I vs TM	●	●	●	●	●	●	●	▼	▼
I vs HMTS	●	▼	▼	●	●	▼	●	▼	▼
I vs TMHS	●	▼	▼	●	●	●	●	▼	▼
I vs FMTH	●	▼	▼	●	●	●	●	▼	▼
FM vs T	●	▲	▲	▼	●	●	●	●	▲
FM vs HM	●	●	●	●	●	▼	●	●	●
FM vs IM	●	●	●	●	●	▲	●	●	●
FM vs RIM	●	▲	▲	●	●	▲	▼	▼	●
FM vs TM	●	▲	●	●	●	●	●	●	●
FM vs HMTS	●	●	●	▼	●	●	●	●	▼
FM vs TMHS	●	●	●	▼	●	●	●	●	▼
FM vs FMTH	●	●	●	●	●	●	●	●	▼
T vs HM	●	▼	▼	●	●	▼	●	●	▼
T vs IM	●	▼	▼	▲	▲	▲	●	●	▼
T vs RIM	●	●	▲	▲	●	▲	●	▼	▼
T vs TM	●	▼	▼	●	●	▼	●	▼	▼
T vs HMTS	▼	▼	▼	●	●	▼	●	▼	▼
T vs TMHS	▼	▼	▼	●	●	▼	●	▼	▼
T vs FMTH	▼	▼	▼	●	●	●	●	▼	▼
HM vs IM	●	●	●	●	▲	▲	●	●	●
HM vs RIM	▲	▲	▲	●	●	▲	▼	▼	●
HM vs TM	●	▲	▲	●	●	●	●	●	●
HM vs HMTS	●	●	●	●	●	●	●	●	▼

	Symbolic Regression			**Classification**			**Santa Fe Ant Trail**		
Algorithms	Easy	Medium	Hard	Easy	Medium	Hard	Easy	Medium	Hard
HM vs TMHS	●	●	●	●	●	●	●	●	●
HM vs FMTH	●	●	●	●	●	▲	●	●	●
IM vs RIM	●	▲	▲	●	●	●	●	▼	●
IM vs TM	●	●	●	●	▼	▼	●	●	●
IM vs HMTS	●	●	●	▼	▼	▼	●	●	▼
IM vs TMHS	●	●	●	▼	▼	▼	●	●	▼
IM vs FMTH	●	●	●	●	▼	▼	●	●	▼
RIM vs TM	●	▼	▼	▼	▼	▼	▲	●	●
RIM vs HMTS	▼	▼	▼	▼	●	▼	●	●	▼
RIM vs TMHS	▼	▼	▼	▼	▼	▼	●	●	▼
RIM vs FMTH	▼	▼	▼	▼	▼	▼	●	●	▼
TM vs HMTS	▼	●	●	●	●	●	●	●	▼
TM vs TMHS	▼	●	●	●	●	●	●	●	▼
TM vs FMTH	●	●	●	●	●	●	●	●	▼
HMTS vs TMHS	●	●	●	●	●	●	●	●	●
HMTS vs FMTH	●	●	●	●	●	●	●	●	●
TMHS vs FMTH	●	●	●	●	●	●	●	●	●

TABLE 6.4: Search for the Best Algorithm - Best of Generation

As we said before, there is no algorithm that bests every other in the statistical tests made. However, in Table 6.5 we count number of times that each algorithm is better or worse than any other, regardless of the benchmark problem, scenario and fitness metric. From this perspective, we can consider the best algorithm to be the HMTS, as it is the one which is more often superior to others. On the other hand, the worst algorithm is the RIM, with a total of 50 times being significantly inferior to others. It is also noteworthy that the SGP is the algorithm which was superior to others the least amount of times. Furthermore, the simple techniques were also not very effective, being superior to others only 10, 13 and 15 times. This reinforces the thesis that further effort must be made on developping novel mechanisms to enable GPs to cope with Dynamic Environments. We highlighted the names of the four algorithms that seem to be the best. The HMTS and TMHS are never worse than any other, the FMTH is worse only once and the HM is worse only three times. We also highlighted the maximum values of superiority and inferiority.

Algorithm	**Superior**	**Inferior**
SGP	6	40

Algorithm	**Superior**	**Inferior**
TH	10	19
Immigrants	13	26
FM	15	11
Transformation	9	39
HM	28	3
IM	9	25
RIM	17	**50**
TM	20	9
HMTS	**35**	0
TMHS	33	0
FMTH	28	1

Table 6.5: This table displays the number of times an algorithm is significantly superior or inferior to others

In all three benchmark problems, the three component hybrid solutions, i.e, HMTS, TMHS and FMTH stand out as having good quality. This is due to the fact of these being balanced algorithms, that use components that favour exploration and exploitation, while retaining past knowledge. In the easier environments, with low frequency and severity environmental changes, some algorithms that have no memory manage to perform well. However, as the difficulty increases, i.e. the changes become more frequent and disruptive, the algorithms that do not use memory stop being able to produce good solutions before the next modification to the environment. In this scenarios the advantages of using memory become notorious. Even if the new environment is different than any of the ones previously seen, there is a chance that the knowledge stored in the memory is better than the current individuals and that its use helps the evolution.

The differences between metrics are notorious. In cases where we know when the changes will take place and are interested in the solution right before that same change, the algorithms found with the Offline Performance are better. On the other hand, if we are interested in a solution that produces good results in any given time, we are better of with the ones found with the Best of Generation. Tables 6.3 and 6.4 show the comparisons between the algorithms.

Chapter 7

Conclusions and Future Work

In this Chapter we discuss the conclusions of our work. After a thorough analysis of the algorithms' behavior, we came to some conclusions. The first being that there is a real need for developing mechanisms capable of handling dynamic environments, as the Standard GP algorithm performs quite poorly in most cases.

Some algorithms, such as the Triggered Hypermutation or the Immigrants, are quite disruptive, while others, such as the Fixed Memory or Transformation, tend to be less disruptive, often leading to local optima. Being very little disruptive, less than the SGP, the Transformation is good in fine tuning. For this reason it is expected to perform well when coupled with a more disruptive method. The algorithms that combine three mechanisms are usually better than the others, due to being able to combine knowledge with good capabilities of exploration and exploitation. However, in some situations, they perform worse than more disruptive techniques. Specifically, there is one case where the FMTH was inferior to the Hypermutation Memory. That may be due to despite incorporating that technique, its population is one third of the one evolved by the Hypermutation Memory, thus having less capacity to evolve to very different areas of the search space. Another probable cause is that when a change is detected, the individual from memory is only injected into the sub-population that is evolved with Fixed Memory. There may be a situation where this algorithm can not make as much use of the memory as the Hypermutation Memory, leading to a worse performance.

From this analysis we came up with a set of empirical guidelines:

- The hardest the scenario is, i.e, more frequent and violent changes in the environment, the more useful is memory, as it is likely to not be enough time to evolve good solutions and that something from the past is somewhat useful in the present.
- In easier and medium scenarios, simple techniques without memory may perform satisfactorily.
- Hybrid techniques that evolve multiple populations, i.e., HTMS, TMHS and FMTH, are usually good in most scenarios.
- It is important to identify the main goal of the evolution, as it will affect the choice of the algorithm. The quality of the solutions of some mechanisms may be quite poor after an environmental change, but recover well and be good before the next changes, while others may no be as good before the next change, but not also be as bad after it happens.
- The choice of the algorithm is problem and scenario dependent, as an example, the RIM performs poorly in the Symbolic Regression and Classification problems, but has good quality in the easy and medium scenarios of the Santa Fe Ant Trail.

In the future, further work should be made on analysing the characteristics of each algorithm, such as population diversity throughout the generations and exploration and exploitation capacities. This study would allow for better understanding how to combine techniques in order to construct algorithms that excel at both exploration and exploitation, while reusing past knowledge. Research should also be made on creating memory mechanisms that take advantage of past knowledge while preventing the convergence to local optima previously found. Furthermore, efforts should be made on creating mechanisms that allow for better understanding the difficulty of the environmental changes, that influence the degree of exploration and exploitation needed, and also if the memory is beneficial in a particular application.

References

[1] T. Baptista and E. Costa. Step evolution: Improving the performance of open-ended evolution simulations. In *Artificial Life (ALIFE), 2013 IEEE Symposium on*, pages 52–59, 2013. doi: 10.1109/ALIFE.2013.6602431.

[2] H. G. Cobb. An Investigation into the Use of Hypermutation as an Adaptive Operator in Genetic Algorithms Having Continuous, Time-Dependent Nonstationary Environments. Technical Report 6760 (NLR Memorandum), Naval Research Lab, Washington, D.C., 1990. URL `citeseer.ist.psu.edu/cobb90investigation.html`.

[3] H. G. Cobb and J. J. Grefenstette. Genetic algorithms for tracking changing environments. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 523–530. Morgan Kaufmann, 1993.

[4] D. E. Goldberg and R. E. Smith. *Nonstationary Function Optimization Using Genetic Algorithms with Dominance and Diploidy.*, pages 59–68. 1987.

[5] I. Gonçalves and S. Silva. Balancing learning and overfitting in genetic programming with interleaved sampling of training data. In K. Krawiec, A. Moraglio, T. Hu, A. Etaner-Uyar, and B. Hu, editors, *Genetic Programming*, volume 7831 of *Lecture Notes in Computer Science*, pages 73–84. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-37206-3. doi: 10.1007/978-3-642-37207-0_7. URL `http://dx.doi.org/10.1007/978-3-642-37207-0_7`.

[6] K. A. D. Jong. *An analysis of the behavior of a class of genetic adaptive systems.* PhD thesis, University of Michigan, 1975.

[7] M. Keijzer. Improving symbolic regression with interval arithmetic and linear scaling. In C. Ryan, T. Soule, M. Keijzer, E. Tsang, R. Poli, and E. Costa, editors, *Genetic Programming*, volume 2610 of *Lecture Notes in Computer Science*, pages 70–82. Springer Berlin Heidelberg, 2003. ISBN 978-3-540-00971-9. doi: 10.1007/3-540-36599-0_7. URL `http://dx.doi.org/10.1007/3-540-36599-0_7`.

[8] M. Keijzer. Scaled symbolic regression. *Genetic Programming and Evolvable Machines*, 5(3):259–269, Sept. 2004. ISSN 1389-2576. doi: 10.1023/B:GENP.0000030195.77571.f9. URL `http://dx.doi.org/10.1023/B:GENP.0000030195.77571.f9`.

[9] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, Cambridge, MA, USA, 1992. ISBN 0-262-11170-5.

[10] C. Liu, H. Liu, and J. Yang. A path planning method based on adaptive genetic algorithm for mobile robot. *Journal of information & computational science*, 8(5), 2011. ISSN 808-814.

[11] S. Luke and L. Panait. A comparison of bloat control methods for genetic programming. *Evol. Comput.*, 14(3):309–344, Sept. 2006. ISSN 1063-6560. doi: 10.1162/evco.2006.14.3.309. URL `http://dx.doi.org/10.1162/evco.2006.14.3.309`.

[12] J. McDermott, D. R. White, S. Luke, L. Manzoni, M. Castelli, L. Vanneschi, W. Jaśkowski, K. Krawiec, R. Harper, K. D. Jong, and U.-M. O'Reilly. Genetic programming needs better benchmarks. In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference*, pages 791–798, Philadelphia, 2012. ACM.

[13] R. W. Morrison. *Designing Evolutionary Algorithms for Dynamic Environments.* SpringerVerlag, 2004. ISBN 3540212310.

[14] K. P. Ng and K. C. Wong. A new diploid scheme and dominance change mechanism for non-stationary function optimization. In *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 159–166, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1-55860-370-0. URL `http://dl.acm.org/citation.cfm?id=645514.657904`.

[15] M. O'Neill, M. Nicolau, and A. Brabazon. Dynamic environments can speed up evolution with genetic programming. Technical report, 2011.

[16] M. Riekert, K. M. Malan, and A. Engelbrect. *Adaptive Genetic Programming for dynamic classification problems*, volume 0002, pages 674–681. IEEE, 2009. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4983010.

[17] S. Silva and E. Costa. Dynamic limits for bloat control in genetic programming and a review of past and current bloat theories. *Genetic Programming and Evolvable Machines*, 10(2):141–179, 2009. ISSN 1389-2576. doi: 10.1007/s10710-008-9075-9. URL http://dx.doi.org/10.1007/s10710-008-9075-9.

[18] S. Silva, S. Dignum, and L. Vanneschi. Operator equalisation for bloat free genetic programming and a survey of bloat control methods. *Genetic Programming and Evolvable Machines*, 13(2):197–238, 2012. ISSN 1389-2576. doi: 10.1007/s10710-011-9150-5. URL http://dx.doi.org/10.1007/s10710-011-9150-5.

[19] A. Simões and E. Costa. On biologically inspired genetic operators: Transformation in the standard genetic algorithm. *Evolutionary Computation*, pages 584–591, 2001.

[20] A. Simões and E. Costa. *Using genetic algorithms to deal with dynamic environments: A comparative study of several approaches based on promoting diversity*, pages 9–13. Morgan Kaufmann, 2002. URL http://evonet.lri.fr/summerschool2003/resources/problems/dynamic/Gecco02_DyKP.pdf.

[21] A. B. Simões. *Improving Memory-based Evolutionary Algorithms for Dynamic Environments*. March 2010.

[22] R. E. Smith and D. E. Goldberg. Diploidy and dominance in artificial genetic search. *Complex Systems*, 6(3):251–285, 1992.

[23] L. Vanneschi and G. Cuccu. A study of genetic programming variable population size for dynamic optimization problems, 2004.

[24] L. Vanneschi, M. Castelli, and S. Silva. Measuring bloat, overfitting and functional complexity in genetic programming. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, GECCO '10, pages 877–884, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0072-8. doi: 10.1145/1830483.1830643. URL http://dx.doi.org/10.1145/1830483.1830643.

[25] F. Vavak, T. C. Fogarty, and K. A. Jukes. A genetic algorithm with variable range of local search for tracking changing environments. *Lecture Notes in Computer Science*, 1141:376–385, 1996.

[26] N. Wagner, Z. Michalewicz, M. J. Khouja, and R. R. McGregor. Time series forecasting for dynamic environments: The dyfor genetic program model. *IEEE Trans. Evolutionary Computation*, pages 433–452, 2007.

[27] K. Weicker. *Performance Measures for Dynamic Environments*, pages 64–73. Springer Berlin Heidelberg, 2002.

[28] D. R. White, J. McDermott, M. Castelli, L. Manzoni, B. W. Goldman, G. Kronberger, W. Jaśkowski, U.-M. O'Reilly, and S. Luke. Better gp benchmarks: Community survey results and proposals. *Genetic Programming and Evolvable Machines*, 14:3–29, 2013. doi: 10.1007/s10710-012-9177-2.

[29] S. Yang. Memory-based immigrants for genetic algorithms in dynamic environments. In *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, GECCO '05, pages 1115–1122, New York, NY, USA, 2005. ACM. ISBN 1-59593-010-8. doi: 10.1145/1068009.1068196. URL `http://doi.acm.org/10.1145/1068009.1068196`.

[30] S. Yang and R. Tinós. A hybrid immigrants scheme for genetic algorithms in dynamic environments. *International Journal of Automation and Computing*, 4 (3):243–254, 2007. ISSN 1476-8186. doi: 10.1007/s11633-007-0243-9. URL `http://dx.doi.org/10.1007/s11633-007-0243-9`.

[31] Z. Yin, A. Brabazon, C. O'Sullivan, and M. O'Neill. Genetic programming for dynamic environments. In *2nd International Symposium "Advances in Artificial Intelligence and Applications"*, volume 2, pages 437–446, Wisla, Poland, October 15-17 2007. URL `http://www.proceedings2007.imcsit.org/pliks/18.pdf`.

Appendix A

Statistic's Data

Symbolic Regression

Descriptive Statistics

Offline Performance

Algorithm	Min	Max	Mean	Std	Skewness	Kurtosis
SGP	0.0028	0.0335	0.0143	0.0084	0.8418	-0.1967
fM	0.0031	0.0863	0.0163	0.0159	2.9183	9.8126
tH	0.0026	0.0331	0.0125	0.0077	0.8781	-0.0102
I	0.0008	0.0889	0.0121	0.0161	3.6893	14.5304
T	0.0012	0.0535	0.0209	0.0134	0.7054	-0.02
tM	0.0024	0.0831	0.0245	0.0165	1.5854	3.5737
iM	0.0006	0.0768	0.0198	0.0174	1.2835	1.711
rIM	0.0041	0.0524	0.0184	0.0122	1.0947	1.0526
TMHS	0.0008	0.0375	0.0107	0.0093	1.3253	1.0515
HMTS	0.0009	0.0255	0.0081	0.0071	1.1178	0.2192
FMTH	0.0008	0.0337	0.0092	0.0075	1.4678	2.0923

Table A.1: Easy Scenario

Algorithm	Min	Max	Mean	Std	Skewness	Kurtosis
SGP	0.0053	0.0298	0.0197	0.0069	-0.4497	-0.8532
fM	0.0046	0.0347	0.0174	0.0068	0.5693	0.1619
tH	0.006	0.056	0.02	0.0103	1.4926	3.0787
I	0.0072	0.053	0.0233	0.0094	1.1002	1.568
T	0.0161	0.1213	0.0383	0.0228	1.936	3.9645
tM	0.0126	0.0501	0.0251	0.0099	0.9363	0.1528
iM	0.0033	0.0584	0.0176	0.0117	1.9127	3.8086
rIM	0.016	1.3507	0.1265	0.3091	3.3087	9.3877

Algorithm	Min	Max	Mean	Std	Skewness	Kurtosis
TMHS	0.007	0.025	0.0155	0.005	0.3102	-0.9999
HMTS	0.0048	0.0244	0.0158	0.0042	-0.3463	0.5898
FMTH	0.0063	0.03	0.0153	0.0058	0.5186	-0.5151

Table A.2: Medium Scenario

Algorithm	Min	Max	Mean	Std	Skewness	Kurtosis
SGP	0.0182	0.0682	0.0374	0.013	0.4204	-0.4502
fM	0.0101	0.0772	0.0295	0.0148	1.2013	1.7503
tH	0.0155	0.049	0.0303	0.0096	0.1705	-0.9629
I	0.0131	0.5281	0.0589	0.0914	4.561	20.2666
T	0.0291	1.0546	0.1128	0.195	4.0138	15.7115
tM	0.0122	0.0648	0.0348	0.0139	0.3286	-0.597
iM	0.0042	0.0491	0.028	0.0113	-0.169	-0.6646
rIM	0.1109	1.0991	0.8325	0.3303	-1.1703	-0.2528
TMHS	0.011	0.041	0.0241	0.0076	-0.068	-0.5534
HMTS	0.0084	0.0382	0.0235	0.0074	0.2244	-0.6582
FMTH	0.0083	0.0591	0.0254	0.0099	1.2165	2.5324

Table A.3: Hard Scenario

Best of Generation

Algorithm	Min	Max	Mean	Std	Skewness	Kurtosis
SGP	0.0073	0.048	0.0244	0.011	0.4655	-0.709
fM	0.0049	0.0967	0.0225	0.0181	2.4016	6.8753
tH	0.006	0.0505	0.0223	0.0102	0.5719	-0.0146
I	0.0043	0.1018	0.0198	0.018	3.2057	11.65
T	0.007	0.072	0.0319	0.0167	0.5304	-0.0905
tM	0.0064	0.088	0.0308	0.0183	1.198	1.8903
iM	0.0035	0.0864	0.0285	0.0226	1.2096	0.642
rIM	0.0087	0.0979	0.0373	0.0231	0.982	0.4347
TMHS	0.0029	0.0496	0.0159	0.0116	1.3474	1.1827
HMTS	0.003	0.0396	0.0136	0.0084	1.1032	1.2675
FMTH	0.0028	0.0874	0.016	0.0153	3.342	12.8055

Table A.4: Easy Scenario

Algorithm	Min	Max	Mean	Std	Skewness	Kurtosis
SGP	0.0093	0.064	0.0346	0.0109	0.1212	0.7282
fM	0.0089	0.0481	0.0237	0.008	0.7965	1.1704
tH	0.0134	0.0683	0.034	0.0116	0.8629	1.1244
I	0.0185	0.0974	0.04	0.0156	1.9369	4.6144
T	0.0287	0.1471	0.0602	0.0274	1.4576	1.9282
tM	0.0168	0.0676	0.0343	0.0122	1.0716	0.8704
iM	0.0067	0.0928	0.0267	0.0167	2.4786	6.8213
rIM	0.0315	1.3644	0.1505	0.309	3.3018	9.3308
TMHS	0.0125	0.035	0.0246	0.0059	0.0142	-1.0739

Algorithm	Min	Max	Mean	Std	Skewness	Kurtosis
HMTS	0.011	0.0395	0.0259	0.0057	-0.2694	0.6071
FMTH	0.0131	0.0684	0.0268	0.0105	1.963	5.6388

Table A.5: Medium Scenario

Algorithm	Min	Max	Mean	Std	Skewness	Kurtosis
SGP	0.0285	0.0915	0.0563	0.016	0.2604	-0.4498
fM	0.0164	0.0912	0.0402	0.0169	1.1075	1.2047
tH	0.0287	0.0711	0.05	0.0116	-0.0002	-0.7856
I	0.0247	0.5193	0.0765	0.0859	4.5766	20.5814
T	0.0485	1.0496	0.1343	0.1883	4.0631	16.1264
tM	0.021	0.093	0.0472	0.0167	0.5041	0.1421
iM	0.0104	0.0606	0.0388	0.0128	-0.379	-0.4919
rIM	0.1201	1.0921	0.8236	0.3228	-1.1779	-0.2269
TMHS	0.0214	0.0587	0.0385	0.0096	-0.1098	-0.6766
HMTS	0.0185	0.0547	0.0376	0.0095	0.128	-0.7206
FMTH	0.0169	0.0752	0.0401	0.0115	0.7598	1.388

Table A.6: Hard Scenario

Normality tests

Offline Performance

Algorithm	D	P-value
SGP	0.4899	0.0
fM	0.5162	0.0
tH	0.4505	0.0
I	0.5223	0.0
T	0.4314	0.0
tM	0.3529	0.0008
iM	0.4482	0.0
rIM	0.4256	0.0
TMHS	0.496	0.0
HMTS	0.4133	0.0
FMTH	0.5027	0.0

Table A.7: Easy Scenario

Algorithm	D	P-value
SGP	0.4882	0.0
FM	0.3753	0.0003
TH	0.4061	0.0001
I	0.4174	0.0
T	0.516	0.0
TM	0.4347	0.0
IM	0.4011	0.0001

Algorithm	D	P-value
RIM	0.7994	0.0
TMHS	0.4316	0.0
HMTS	0.3821	0.0002
FMTH	0.4497	0.0

Table A.8: Medium Scenario

Algorithm	D	P-value
SGP	0.3564	0.0006
FM	0.4126	0.0
TH	0.4127	0.0
I	0.5664	0.0
T	0.6167	0.0
TM	0.4235	0.0
IM	0.3636	0.0005
RIM	0.6037	0.0
TMHS	0.3822	0.0002
HMTS	0.446	0.0
FMTH	0.4579	0.0

Table A.9: Hard Scenario

Best of Generation

Algorithm	D	P-value
SGP	0.4138	0.0
FM	0.5222	0.0
TH	0.4314	0.0
I	0.4419	0.0
T	0.397	0.0001
TM	0.3356	0.0016
IM	0.4367	0.0
RIM	0.4247	0.0
TMHS	0.4549	0.0
HMTS	0.454	0.0
FMTH	0.5034	0.0

Table A.10: Easy Scenario

Algorithm	D	P-value
SGP	0.3828	0.0002
FM	0.4375	0.0
TH	0.4144	0.0
I	0.3811	0.0002
T	0.4574	0.0
TM	0.4427	0.0
IM	0.3511	0.0008
RIM	0.7352	0.0

Algorithm	D	P-value
TMHS	0.4276	0.0
HMTS	0.3479	0.0009
FMTH	0.4414	0.0

Table A.11: Medium Scenario

Algorithm	D	P-value
SGP	0.3704	0.0003
FM	0.4199	0.0
TH	0.4163	0.0
I	0.5155	0.0
T	0.6004	0.0
TM	0.3912	0.0001
IM	0.3916	0.0001
RIM	0.6002	0.0
TMHS	0.411	0.0
HMTS	0.3905	0.0001
FMTH	0.3959	0.0001

Table A.12: Hard Scenario

Simple Techniques

Offline Performance

Algorithms	Scenario	Chi-prob	P-value
SGP, TH, I, FM	Easy	8.0	0.046
SGP, TH, I, FM	Medium	7.12	0.0682
SGP, TH, I, FM	Hard	15.64	0.0013

Table A.13: Friedman's Anova

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-1.3472	0.1779	298.0	167.0	0.0167	0
SGP-I	-2.7253	0.0064	365.0	100.0	0.0167	-0.4976
SGP-FM	-0.0926	0.9263	237.0	228.0	0.0167	0

Table A.14: Wilcoxon Signed Ranks - Easy

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-0.6068	0.544	262.0	203.0	0.0167	0
SGP-I	-2.026	0.0428	134.0	331.0	0.0167	0
SGP-FM	-1.3884	0.165	300.0	165.0	0.0167	0

Table A.15: Wilcoxon Signed Ranks - Medium

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-2.6636	0.0077	362.0	103.0	0.0167	-0.4863
SGP-I	-0.7302	0.4653	197.0	268.0	0.0167	0
SGP-FM	-2.17	0.03	338.0	127.0	0.0167	0

Table A.16: Wilcoxon Signed Ranks - Hard

Best of Generation

Algorithms	Scenario	Chi-prob	P-value
SGP, TH, I, FM	Easy	11.16	0.0109
SGP, TH, I, FM	Medium	26.68	0.0
SGP, TH, I, FM	Hard	19.4	0.0002

Table A.17: Friedman's Anova

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-1.2238	0.221	292.0	173.0	0.0167	0
SGP-I	-2.8693	0.0041	372.0	93.0	0.0167	-0.5239
SGP-FM	-1.2032	0.2289	291.0	174.0	0.0167	0

Table A.18: Wilcoxon Signed Ranks - Easy

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-0.6685	0.5038	265.0	200.0	0.0167	0
SGP-I	-2.1083	0.035	130.0	335.0	0.0167	0
SGP-FM	-3.8977	0.0001	422.0	43.0	0.0167	-0.7116

Table A.19: Wilcoxon Signed Ranks - Medium

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-2.0877	0.0368	334.0	131.0	0.0167	0
SGP-I	-0.977	0.3286	185.0	280.0	0.0167	0
SGP-FM	-3.2601	0.0011	391.0	74.0	0.0167	-0.5952

Table A.20: Wilcoxon Signed Ranks - Hard

Hybrid Techniques

Offline Performance

Algorithms	Scenario	Chi-prob	P-value
T, SGP, TH, I, FM	Easy	15.1467	0.0044
HM, TH, FM	Easy	0.8667	0.6483
IM, I, FM	Easy	3.8	0.1496
RIM, IM, I, FM	Easy	9.28	0.0258
TM, FM, T	Easy	7.8	0.0202

Algorithms	Scenario	Chi-prob	P-value
HMTS, SGP, HM, T	Easy	21.16	0.0001
TMHS, SGP, TM, TH	Easy	19.36	0.0002
FMTH, FM, TH, T	Easy	15.48	0.0014
T, SGP, TH, I, FM	Medium	37.6	0.0
HM, TH, FM	Medium	6.8667	0.0323
IM, I, FM	Medium	8.6	0.0136
RIM, IM, I, FM	Medium	40.04	0.0
TM, FM, T	Medium	22.8667	0.0
HMTS, SGP, HM, T	Medium	43.72	0.0
TMHS, SGP, TM, TH	Medium	18.92	0.0003
FMTH, FM, TH, T	Medium	36.12	0.0
T, SGP, TH, I, FM	Hard	50.7733	0.0
HM, TH, FM	Hard	16.4667	0.0003
IM, I, FM	Hard	14.8667	0.0006
RIM, IM, I, FM	Hard	62.92	0.0
TM, FM, T	Hard	31.2	0.0
HMTS, SGP, HM, T	Hard	62.08	0.0
TMHS, SGP, TM, TH	Hard	15.12	0.0017
FMTH, FM, TH, T	Hard	49.96	0.0

Table A.21: Friedman's Anova

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
T-SGP	-1.7997	0.0719	320.0	145.0	0.0125	0
T-TH	-2.5813	0.0098	358.0	107.0	0.0125	-0.4713
T-I	-2.9721	0.003	377.0	88.0	0.0125	-0.5426
T-FM	-1.6969	0.0897	315.0	150.0	0.0125	0
HM-TH	-0.4834	0.6288	256.0	209.0	0.025	0
HM-FM	-0.8742	0.382	190.0	275.0	0.025	0
IM-I	-2.4168	0.0157	350.0	115.0	0.025	-0.4412
IM-FM	-1.1621	0.2452	289.0	176.0	0.025	0
RIM-IM	-0.072	0.9426	229.0	236.0	0.0167	0
RIM-I	-2.5196	0.0117	355.0	110.0	0.0167	-0.46
RIM-FM	-1.4912	0.1359	305.0	160.0	0.0167	0
TM-FM	-2.4168	0.0157	350.0	115.0	0.025	-0.4412
TM-T	-0.7919	0.4284	271.0	194.0	0.025	0
HMTS-SGP	-2.7047	0.0068	101.0	364.0	0.0167	-0.4938
HMTS-HM	-2.4579	0.014	113.0	352.0	0.0167	-0.4488
HMTS-T	-4.0211	0.0001	37.0	428.0	0.0167	-0.7342
TMHS-SGP	-2.1905	0.0285	126.0	339.0	0.0167	0
TMHS-TM	-4.0417	0.0001	36.0	429.0	0.0167	-0.7379
TMHS-TH	-0.9153	0.36	188.0	277.0	0.0167	0
FMTH-FM	-2.3345	0.0196	119.0	346.0	0.0167	0
FMTH-TH	-2.0877	0.0368	131.0	334.0	0.0167	0
FMTH-T	-3.3012	0.001	72.0	393.0	0.0167	-0.6027

Table A.22: Wilcoxon Signed Ranks - Easy

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
T-SGP	-4.3296	0.0	443.0	22.0	0.0125	-0.7905
T-TH	-3.9388	0.0001	424.0	41.0	0.0125	-0.7191
T-I	-3.1161	0.0018	384.0	81.0	0.0125	-0.5689
T-FM	-4.3502	0.0	444.0	21.0	0.0125	-0.7942
HM-TH	-2.4168	0.0157	115.0	350.0	0.025	-0.4412
HM-FM	-1.8409	0.0656	143.0	322.0	0.025	0
IM-I	-2.2522	0.0243	123.0	342.0	0.025	-0.4112
IM-FM	-0.4011	0.6884	213.0	252.0	0.025	0
RIM-IM	-4.3296	0.0	443.0	22.0	0.0167	-0.7905
RIM-I	-3.3424	0.0008	395.0	70.0	0.0167	-0.6102
RIM-FM	-4.6587	0.0	459.0	6.0	0.0167	-0.8506
TM-FM	-2.8899	0.0039	373.0	92.0	0.025	-0.5276
TM-T	-2.643	0.0082	104.0	361.0	0.025	-0.4825
HMTS-SGP	-2.3345	0.0196	119.0	346.0	0.0167	0
HMTS-HM	-1.1621	0.2452	289.0	176.0	0.0167	0
HMTS-T	-4.6382	0.0	7.0	458.0	0.0167	-0.8468
TMHS-SGP	-2.5196	0.0117	110.0	355.0	0.0167	-0.46
TMHS-TM	-4.124	0.0	32.0	433.0	0.0167	-0.7529
TMHS-TH	-1.6969	0.0897	150.0	315.0	0.0167	0
FMTH-FM	-0.833	0.4048	192.0	273.0	0.0167	0
FMTH-TH	-2.2934	0.0218	121.0	344.0	0.0167	0
FMTH-T	-4.3708	0.0	20.0	445.0	0.0167	-0.798

Table A.23: Wilcoxon Signed Ranks - Medium

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
T-SGP	-4.124	0.0	433.0	32.0	0.0125	-0.7529
T-TH	-4.7821	0.0	465.0	0.0	0.0125	-0.8731
T-I	-3.2601	0.0011	391.0	74.0	0.0125	-0.5952
T-FM	-4.597	0.0	456.0	9.0	0.0125	-0.8393
HM-TH	-4.2062	0.0	28.0	437.0	0.025	-0.7679
HM-FM	-2.6225	0.0087	105.0	360.0	0.025	-0.4788
IM-I	-3.4246	0.0006	66.0	399.0	0.025	-0.6252
IM-FM	-0.1954	0.8451	242.0	223.0	0.025	0
RIM-IM	-4.7821	0.0	465.0	0.0	0.0167	-0.8731
RIM-I	-4.7821	0.0	465.0	0.0	0.0167	-0.8731
RIM-FM	-4.7821	0.0	465.0	0.0	0.0167	-0.8731
TM-FM	-1.7175	0.0859	316.0	149.0	0.025	0
TM-T	-4.2474	0.0	26.0	439.0	0.025	-0.7755
HMTS-SGP	-3.7332	0.0002	51.0	414.0	0.0167	-0.6816
HMTS-HM	-1.0798	0.2802	285.0	180.0	0.0167	0
HMTS-T	-4.7821	0.0	0.0	465.0	0.0167	-0.8731
TMHS-SGP	-4.1034	0.0	33.0	432.0	0.0167	-0.7492
TMHS-TM	-3.2189	0.0013	76.0	389.0	0.0167	-0.5877
TMHS-TH	-2.5608	0.0104	108.0	357.0	0.0167	-0.4675
FMTH-FM	-1.2444	0.2134	172.0	293.0	0.0167	0
FMTH-TH	-1.882	0.0598	141.0	324.0	0.0167	0
FMTH-T	-4.6587	0.0	6.0	459.0	0.0167	-0.8506

Table A.24: Wilcoxon Signed Ranks - Hard

Best of Generation

Algorithms	Scenario	Chi-prob	P-value
T, SGP, TH, I, FM	Easy	18.3733	0.001
HM, TH, FM	Easy	1.8667	0.3932
IM, I, FM	Easy	3.8	0.1496
RIM, IM, I, FM	Easy	16.52	0.0009
TM, FM, T	Easy	5.2667	0.0718
HMTS, SGP, HM, T	Easy	27.88	0.0
TMHS, SGP, TM, TH	Easy	14.68	0.0021
FMTH, FM, TH, T	Easy	21.52	0.0001
T, SGP, TH, I, FM	Medium	57.4133	0.0
HM, TH, FM	Medium	13.0667	0.0015
IM, I, FM	Medium	20.0	0.0
RIM, IM, I, FM	Medium	55.32	0.0
TM, FM, T	Medium	39.2	0.0
HMTS, SGP, HM, T	Medium	55.32	0.0
TMHS, SGP, TM, TH	Medium	20.56	0.0001
FMTH, FM, TH, T	Medium	48.04	0.0
T, SGP, TH, I, FM	Hard	56.5333	0.0
HM, TH, FM	Hard	30.8667	0.0
IM, I, FM	Hard	16.2667	0.0003
RIM, IM, I, FM	Hard	63.76	0.0
TM, FM, T	Hard	40.8667	0.0
HMTS, SGP, HM, T	Hard	69.48	0.0
TMHS, SGP, TM, TH	Hard	18.68	0.0003
FMTH, FM, TH, T	Hard	54.68	0.0

TABLE A.25: Friedman's Anova

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
T-SGP	-1.6969	0.0897	315.0	150.0	0.0125	0
T-TH	-2.2522	0.0243	342.0	123.0	0.0125	0
T-I	-3.0338	0.0024	380.0	85.0	0.0125	-0.5539
T-FM	-2.1905	0.0285	339.0	126.0	0.0125	0
HM-TH	-0.7096	0.4779	198.0	267.0	0.025	0
HM-FM	-1.121	0.2623	178.0	287.0	0.025	0
IM-I	-1.882	0.0598	324.0	141.0	0.025	0
IM-FM	-1.2855	0.1986	295.0	170.0	0.025	0
RIM-IM	-1.738	0.0822	317.0	148.0	0.0167	0
RIM-I	-3.0955	0.002	383.0	82.0	0.0167	-0.5652
RIM-FM	-3.2395	0.0012	390.0	75.0	0.0167	-0.5915
TM-FM	-1.9026	0.0571	325.0	140.0	0.025	0
TM-T	-0.4217	0.6733	212.0	253.0	0.025	0
HMTS-SGP	-3.2601	0.0011	74.0	391.0	0.0167	-0.5952
HMTS-HM	-2.3756	0.0175	117.0	348.0	0.0167	0
HMTS-T	-4.597	0.0	9.0	456.0	0.0167	-0.8393
TMHS-SGP	-2.9927	0.0028	87.0	378.0	0.0167	-0.5464
TMHS-TM	-3.5686	0.0004	59.0	406.0	0.0167	-0.6515
TMHS-TH	-2.1288	0.0333	129.0	336.0	0.0167	0

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
FMTH-FM	-2.3139	0.0207	120.0	345.0	0.0167	0
FMTH-TH	-2.787	0.0053	97.0	368.0	0.0167	-0.5088
FMTH-T	-3.5892	0.0003	58.0	407.0	0.0167	-0.6553

Table A.26: Wilcoxon Signed Ranks - Easy

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
T-SGP	-4.0828	0.0	431.0	34.0	0.0125	-0.7454
T-TH	-4.0622	0.0	430.0	35.0	0.0125	-0.7417
T-I	-3.4041	0.0007	398.0	67.0	0.0125	-0.6215
T-FM	-4.7616	0.0	464.0	1.0	0.0125	-0.8693
HM-TH	-3.5686	0.0004	59.0	406.0	0.025	-0.6515
HM-FM	-0.6685	0.5038	200.0	265.0	0.025	0
IM-I	-3.3012	0.001	72.0	393.0	0.025	-0.6027
IM-FM	-0.4217	0.6733	253.0	212.0	0.025	0
RIM-IM	-4.6176	0.0	457.0	8.0	0.0167	-0.8431
RIM-I	-3.692	0.0002	412.0	53.0	0.0167	-0.6741
RIM-FM	-4.7821	0.0	465.0	0.0	0.0167	-0.8731
TM-FM	-3.4863	0.0005	402.0	63.0	0.025	-0.6365
TM-T	-3.836	0.0001	46.0	419.0	0.025	-0.7004
HMTS-SGP	-3.2395	0.0012	75.0	390.0	0.0167	-0.5915
HMTS-HM	-1.6146	0.1064	311.0	154.0	0.0167	0
HMTS-T	-4.7821	0.0	0.0	465.0	0.0167	-0.8731
TMHS-SGP	-3.4246	0.0006	66.0	399.0	0.0167	-0.6252
TMHS-TM	-3.3424	0.0008	70.0	395.0	0.0167	-0.6102
TMHS-TH	-3.1984	0.0014	77.0	388.0	0.0167	-0.5839
FMTH-FM	-1.1004	0.2712	286.0	179.0	0.0167	0
FMTH-TH	-2.643	0.0082	104.0	361.0	0.0167	-0.4825
FMTH-T	-4.4942	0.0	14.0	451.0	0.0167	-0.8205

Table A.27: Wilcoxon Signed Ranks - Medium

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
T-SGP	-4.2885	0.0	441.0	24.0	0.0125	-0.783
T-TH	-4.7821	0.0	465.0	0.0	0.0125	-0.8731
T-Immigrants	-3.3218	0.0009	394.0	71.0	0.0125	-0.6065
T-FM	-4.7616	0.0	464.0	1.0	0.0125	-0.8693
HM-TH	-4.7821	0.0	0.0	465.0	0.025	-0.8731
HM-FM	-2.2317	0.0256	124.0	341.0	0.025	0
IM-I	-4.0417	0.0001	36.0	429.0	0.025	-0.7379
IM-FM	-0.4217	0.6733	253.0	212.0	0.025	0
RIM-IM	-4.7821	0.0	465.0	0.0	0.0167	-0.8731
RIM-I	-4.7821	0.0	465.0	0.0	0.0167	-0.8731
RIM-FM	-4.7821	0.0	465.0	0.0	0.0167	-0.8731
TM-FM	-2.026	0.0428	331.0	134.0	0.025	0
TM-T	-4.6587	0.0	6.0	459.0	0.025	-0.8506
HMTS-SGP	-3.8771	0.0001	44.0	421.0	0.0167	-0.7079
HMTS-HM	-2.0671	0.0387	333.0	132.0	0.0167	0
HMTS-T	-4.7821	0.0	0.0	465.0	0.0167	-0.8731

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
TMHS-SGP	-4.1445	0.0	31.0	434.0	0.0167	-0.7567
TMHS-TM	-2.2934	0.0218	121.0	344.0	0.0167	0
TMHS-TH	-3.1572	0.0016	79.0	386.0	0.0167	-0.5764
FMTH-FM	-0.1543	0.8774	240.0	225.0	0.0167	0
FMTH-TH	-2.5196	0.0117	110.0	355.0	0.0167	-0.46
FMTH-T	-4.7616	0.0	1.0	464.0	0.0167	-0.8693

Table A.28: Wilcoxon Signed Ranks - Hard

Search for the Best Algorithm

Offline Performance

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-1.3472	0.1779	298.0	167.0	0.0008	0
SGP-I	-2.7253	0.0064	365.0	100.0	0.0008	0
SGP-FM	-0.0926	0.9263	237.0	228.0	0.0008	0
SGP-T	-1.7997	0.0719	145.0	320.0	0.0008	0
SGP-HM	-0.7096	0.4779	267.0	198.0	0.0008	0
SGP-IM	-1.0181	0.3086	183.0	282.0	0.0008	0
SGP-RIM	-1.1415	0.2536	177.0	288.0	0.0008	0
SGP-TM	-3.2601	0.0011	74.0	391.0	0.0008	0
SGP-HMTS	-2.7047	0.0068	364.0	101.0	0.0008	0
SGP-TMHS	-2.1905	0.0285	339.0	126.0	0.0008	0
SGP-FMTH	-2.4373	0.0148	351.0	114.0	0.0008	0
TH-I	-1.5529	0.1204	308.0	157.0	0.0008	0
TH-FM	-0.8536	0.3933	191.0	274.0	0.0008	0
TH-T	-2.5813	0.0098	107.0	358.0	0.0008	0
TH-HM	-0.4834	0.6288	209.0	256.0	0.0008	0
TH-IM	-1.6969	0.0897	150.0	315.0	0.0008	0
TH-RIM	-1.7997	0.0719	145.0	320.0	0.0008	0
TH-TM	-3.3012	0.001	72.0	393.0	0.0008	0
TH-HMTS	-1.9643	0.0495	328.0	137.0	0.0008	0
TH-TMHS	-0.9153	0.36	277.0	188.0	0.0008	0
TH-FMTH	-2.0877	0.0368	334.0	131.0	0.0008	0
I-FM	-1.5118	0.1306	159.0	306.0	0.0008	0
I-T	-2.9721	0.003	88.0	377.0	0.0008	0
I-HM	-1.4501	0.147	162.0	303.0	0.0008	0
I-IM	-2.4168	0.0157	115.0	350.0	0.0008	0
I-RIM	-2.5196	0.0117	110.0	355.0	0.0008	0
I-TM	-3.6303	0.0003	56.0	409.0	0.0008	-0.6628
I-HMTS	-0.8742	0.382	275.0	190.0	0.0008	0
I-TMHS	-0.4628	0.6435	210.0	255.0	0.0008	0
I-FMTH	-0.5451	0.5857	259.0	206.0	0.0008	0
FM-T	-1.6969	0.0897	150.0	315.0	0.0008	0
FM-HM	-0.8742	0.382	275.0	190.0	0.0008	0
FM-IM	-1.1621	0.2452	176.0	289.0	0.0008	0
FM-RIM	-1.4912	0.1359	160.0	305.0	0.0008	0

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
FM-TM	-2.4168	0.0157	115.0	350.0	0.0008	0
FM-HMTS	-2.5402	0.0111	356.0	109.0	0.0008	0
FM-TMHS	-1.9848	0.0472	329.0	136.0	0.0008	0
FM-FMTH	-2.3345	0.0196	346.0	119.0	0.0008	0
T-HM	-1.6558	0.0978	313.0	152.0	0.0008	0
T-IM	-0.5862	0.5577	261.0	204.0	0.0008	0
T-RIM	-0.977	0.3286	280.0	185.0	0.0008	0
T-TM	-0.7919	0.4284	194.0	271.0	0.0008	0
T-HMTS	-4.0211	0.0001	428.0	37.0	0.0008	-0.7342
T-TMHS	-2.7664	0.0057	367.0	98.0	0.0008	0
T-FMTH	-3.3012	0.001	393.0	72.0	0.0008	0
HM-IM	-1.1621	0.2452	176.0	289.0	0.0008	0
HM-RIM	-1.5118	0.1306	159.0	306.0	0.0008	0
HM-TM	-2.3962	0.0166	116.0	349.0	0.0008	0
HM-HMTS	-2.4579	0.014	352.0	113.0	0.0008	0
HM-TMHS	-1.3267	0.1846	297.0	168.0	0.0008	0
HM-FMTH	-2.3756	0.0175	348.0	117.0	0.0008	0
IM-RIM	-0.072	0.9426	236.0	229.0	0.0008	0
IM-TM	-1.2855	0.1986	170.0	295.0	0.0008	0
IM-HMTS	-2.6019	0.0093	359.0	106.0	0.0008	0
IM-TMHS	-2.2728	0.023	343.0	122.0	0.0008	0
IM-FMTH	-2.5196	0.0117	355.0	110.0	0.0008	0
RIM-TM	-1.5323	0.1254	158.0	307.0	0.0008	0
RIM-HMTS	-3.4041	0.0007	398.0	67.0	0.0008	-0.6215
RIM-TMHS	-2.8076	0.005	369.0	96.0	0.0008	0
RIM-FMTH	-3.7743	0.0002	416.0	49.0	0.0008	-0.6891
TM-HMTS	-4.0211	0.0001	428.0	37.0	0.0008	-0.7342
TM-TMHS	-4.0417	0.0001	429.0	36.0	0.0008	-0.7379
TM-FMTH	-3.7537	0.0002	415.0	50.0	0.0008	-0.6853
HMTS-TMHS	-1.3061	0.1915	169.0	296.0	0.0008	0
HMTS-FMTH	-0.7096	0.4779	198.0	267.0	0.0008	0
TMHS-FMTH	-0.5451	0.5857	259.0	206.0	0.0008	0

TABLE A.29: Wilcoxon Signed Ranks - Easy Scenario

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-0.6068	0.544	262.0	203.0	0.0008	0
SGP-I	-2.026	0.0428	134.0	331.0	0.0008	0
SGP-FM	-1.3884	0.165	300.0	165.0	0.0008	0
SGP-T	-4.3296	0.0	22.0	443.0	0.0008	-0.7905
SGP-HM	-2.7253	0.0064	365.0	100.0	0.0008	0
SGP-IM	-1.5118	0.1306	306.0	159.0	0.0008	0
SGP-RIM	-4.1034	0.0	33.0	432.0	0.0008	-0.7492
SGP-TM	-2.2728	0.023	122.0	343.0	0.0008	0
SGP-HMTS	-2.3345	0.0196	346.0	119.0	0.0008	0
SGP-TMHS	-2.5196	0.0117	355.0	110.0	0.0008	0
SGP-FMTH	-2.3139	0.0207	345.0	120.0	0.0008	0
TH-I	-1.7586	0.0786	147.0	318.0	0.0008	0
TH-FM	-1.0798	0.2802	285.0	180.0	0.0008	0

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
TH-T	-3.9388	0.0001	41.0	424.0	0.0008	-0.7191
TH-HM	-2.4168	0.0157	350.0	115.0	0.0008	0
TH-IM	-1.1621	0.2452	289.0	176.0	0.0008	0
TH-RIM	-4.3091	0.0	23.0	442.0	0.0008	-0.7867
TH-TM	-2.2934	0.0218	121.0	344.0	0.0008	0
TH-HMTS	-1.4295	0.1529	302.0	163.0	0.0008	0
TH-TMHS	-1.6969	0.0897	315.0	150.0	0.0008	0
TH-FMTH	-2.2934	0.0218	344.0	121.0	0.0008	0
I-FM	-2.3345	0.0196	346.0	119.0	0.0008	0
I-T	-3.1161	0.0018	81.0	384.0	0.0008	0
I-HM	-3.7949	0.0001	417.0	48.0	0.0008	-0.6928
I-IM	-2.2522	0.0243	342.0	123.0	0.0008	0
I-RIM	-3.3424	0.0008	70.0	395.0	0.0008	0
I-TM	-0.5245	0.5999	207.0	258.0	0.0008	0
I-HMTS	-3.3629	0.0008	396.0	69.0	0.0008	0
I-TMHS	-3.3218	0.0009	394.0	71.0	0.0008	0
I-FMTH	-3.0544	0.0023	381.0	84.0	0.0008	0
FM-T	-4.3502	0.0	21.0	444.0	0.0008	-0.7942
FM-HM	-1.8409	0.0656	322.0	143.0	0.0008	0
FM-IM	-0.4011	0.6884	252.0	213.0	0.0008	0
FM-RIM	-4.6587	0.0	6.0	459.0	0.0008	-0.8506
FM-TM	-2.8899	0.0039	92.0	373.0	0.0008	0
FM-HMTS	-0.8536	0.3933	274.0	191.0	0.0008	0
FM-TMHS	-0.977	0.3286	280.0	185.0	0.0008	0
FM-FMTH	-0.833	0.4048	273.0	192.0	0.0008	0
T-HM	-4.7616	0.0	464.0	1.0	0.0008	-0.8693
T-IM	-4.0005	0.0001	427.0	38.0	0.0008	-0.7304
T-RIM	-0.7919	0.4284	194.0	271.0	0.0008	0
T-TM	-2.643	0.0082	361.0	104.0	0.0008	0
T-HMTS	-4.6382	0.0	458.0	7.0	0.0008	-0.8468
T-TMHS	-4.5353	0.0	453.0	12.0	0.0008	-0.828
T-FMTH	-4.3708	0.0	445.0	20.0	0.0008	-0.798
HM-IM	-1.1827	0.2369	175.0	290.0	0.0008	0
HM-RIM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
HM-TM	-3.8977	0.0001	43.0	422.0	0.0008	-0.7116
HM-HMTS	-1.1621	0.2452	176.0	289.0	0.0008	0
HM-TMHS	-1.1621	0.2452	176.0	289.0	0.0008	0
HM-FMTH	-0.8742	0.382	190.0	275.0	0.0008	0
IM-RIM	-4.3296	0.0	22.0	443.0	0.0008	-0.7905
IM-TM	-2.7459	0.006	99.0	366.0	0.0008	0
IM-HMTS	-0.2982	0.7655	218.0	247.0	0.0008	0
IM-TMHS	-0.0926	0.9263	237.0	228.0	0.0008	0
IM-FMTH	-0.689	0.4908	266.0	199.0	0.0008	0
RIM-TM	-2.9104	0.0036	374.0	91.0	0.0008	0
RIM-HMTS	-4.6587	0.0	459.0	6.0	0.0008	-0.8506
RIM-TMHS	-4.6587	0.0	459.0	6.0	0.0008	-0.8506
RIM-FMTH	-4.5765	0.0	455.0	10.0	0.0008	-0.8355
TM-HMTS	-3.4658	0.0005	401.0	64.0	0.0008	-0.6328
TM-TMHS	-4.124	0.0	433.0	32.0	0.0008	-0.7529

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
TM-FMTH	-4.2062	0.0	437.0	28.0	0.0008	-0.7679
HMTS-TMHS	-0.0103	0.9918	232.0	233.0	0.0008	0
HMTS-FMTH	-0.9564	0.3389	279.0	186.0	0.0008	0
TMHS-FMTH	-0.4011	0.6884	252.0	213.0	0.0008	0

TABLE A.30: Wilcoxon Signed Ranks - Medium Scenario

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-2.6636	0.0077	362.0	103.0	0.0008	0
SGP-I	-0.7302	0.4653	197.0	268.0	0.0008	0
SGP-FM	-2.17	0.03	338.0	127.0	0.0008	0
SGP-T	-4.124	0.0	32.0	433.0	0.0008	-0.7529
SGP-HM	-4.6587	0.0	459.0	6.0	0.0008	-0.8506
SGP-IM	-3.1367	0.0017	385.0	80.0	0.0008	0
SGP-RIM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
SGP-TM	-0.5039	0.6143	257.0	208.0	0.0008	0
SGP-HMTS	-3.7332	0.0002	414.0	51.0	0.0008	-0.6816
SGP-TMHS	-4.1034	0.0	432.0	33.0	0.0008	-0.7492
SGP-FMTH	-3.2601	0.0011	391.0	74.0	0.0008	0
TH-I	-3.1367	0.0017	80.0	385.0	0.0008	0
TH-FM	-0.6068	0.544	262.0	203.0	0.0008	0
TH-T	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
TH-HM	-4.2062	0.0	437.0	28.0	0.0008	-0.7679
TH-IM	-0.8124	0.4165	272.0	193.0	0.0008	0
TH-RIM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
TH-TM	-1.2855	0.1986	170.0	295.0	0.0008	0
TH-HMTS	-2.6636	0.0077	362.0	103.0	0.0008	0
TH-TMHS	-2.5608	0.0104	357.0	108.0	0.0008	0
TH-FMTH	-1.882	0.0598	324.0	141.0	0.0008	0
I-FM	-2.7047	0.0068	364.0	101.0	0.0008	0
I-T	-3.2601	0.0011	74.0	391.0	0.0008	0
I-HM	-4.2885	0.0	441.0	24.0	0.0008	-0.783
I-IM	-3.4246	0.0006	399.0	66.0	0.0008	-0.6252
I-RIM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
I-TM	-1.3884	0.165	300.0	165.0	0.0008	0
I-HMTS	-4.2679	0.0	440.0	25.0	0.0008	-0.7792
I-TMHS	-4.0828	0.0	431.0	34.0	0.0008	-0.7454
I-FMTH	-3.9388	0.0001	424.0	41.0	0.0008	-0.7191
FM-T	-4.597	0.0	9.0	456.0	0.0008	-0.8393
FM-HM	-2.6225	0.0087	360.0	105.0	0.0008	0
FM-IM	-0.1954	0.8451	223.0	242.0	0.0008	0
FM-RIM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
FM-TM	-1.7175	0.0859	149.0	316.0	0.0008	0
FM-HMTS	-2.1083	0.035	335.0	130.0	0.0008	0
FM-TMHS	-1.2032	0.2289	291.0	174.0	0.0008	0
FM-FMTH	-1.2444	0.2134	293.0	172.0	0.0008	0
T-HM	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
T-IM	-4.5559	0.0	454.0	11.0	0.0008	-0.8318
T-RIM	-4.7616	0.0	1.0	464.0	0.0008	-0.8693

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
T-TM	-4.2474	0.0	439.0	26.0	0.0008	-0.7755
T-HMTS	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
T-TMHS	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
T-FMTH	-4.6587	0.0	459.0	6.0	0.0008	-0.8506
HM-IM	-3.1367	0.0017	80.0	385.0	0.0008	0
HM-RIM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
HM-TM	-3.6303	0.0003	56.0	409.0	0.0008	-0.6628
HM-HMTS	-1.0798	0.2802	180.0	285.0	0.0008	0
HM-TMHS	-1.3061	0.1915	169.0	296.0	0.0008	0
HM-FMTH	-1.4501	0.147	162.0	303.0	0.0008	0
IM-RIM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
IM-TM	-2.2934	0.0218	121.0	344.0	0.0008	0
IM-HMTS	-1.6146	0.1064	311.0	154.0	0.0008	0
IM-TMHS	-1.265	0.2059	294.0	171.0	0.0008	0
IM-FMTH	-0.9153	0.36	277.0	188.0	0.0008	0
RIM-TM	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
RIM-HMTS	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
RIM-TMHS	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
RIM-FMTH	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
TM-HMTS	-3.2189	0.0013	389.0	76.0	0.0008	0
TM-TMHS	-3.2189	0.0013	389.0	76.0	0.0008	0
TM-FMTH	-2.9927	0.0028	378.0	87.0	0.0008	0
HMTS-TMHS	-0.3394	0.7343	216.0	249.0	0.0008	0
HMTS-FMTH	-0.7096	0.4779	198.0	267.0	0.0008	0
TMHS-FMTH	-0.072	0.9426	229.0	236.0	0.0008	0

Table A.31: Wilcoxon Signed Ranks - Hard Scenario

Best of Generation

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-1.2238	0.221	292.0	173.0	0.0008	0
SGP-I	-2.8693	0.0041	372.0	93.0	0.0008	0
SGP-FM	-1.2032	0.2289	291.0	174.0	0.0008	0
SGP-T	-1.6969	0.0897	150.0	315.0	0.0008	0
SGP-HM	-1.594	0.1109	310.0	155.0	0.0008	0
SGP-IM	-0.3599	0.7189	215.0	250.0	0.0008	0
SGP-RIM	-1.9848	0.0472	136.0	329.0	0.0008	0
SGP-TM	-1.7586	0.0786	147.0	318.0	0.0008	0
SGP-HMTS	-3.2601	0.0011	391.0	74.0	0.0008	0
SGP-TMHS	-2.9927	0.0028	378.0	87.0	0.0008	0
SGP-FMTH	-2.9721	0.003	377.0	88.0	0.0008	0
TH-I	-1.9848	0.0472	329.0	136.0	0.0008	0
TH-FM	-0.3599	0.7189	250.0	215.0	0.0008	0
TH-T	-2.2522	0.0243	123.0	342.0	0.0008	0
TH-HM	-0.7096	0.4779	267.0	198.0	0.0008	0
TH-IM	-0.7507	0.4528	196.0	269.0	0.0008	0
TH-RIM	-2.5813	0.0098	107.0	358.0	0.0008	0
TH-TM	-2.17	0.03	127.0	338.0	0.0008	0

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
TH-HMTS	-2.787	0.0053	368.0	97.0	0.0008	0
TH-TMHS	-2.1288	0.0333	336.0	129.0	0.0008	0
TH-FMTH	-2.787	0.0053	368.0	97.0	0.0008	0
I-FM	-1.0181	0.3086	183.0	282.0	0.0008	0
I-T	-3.0338	0.0024	85.0	380.0	0.0008	0
I-HM	-0.7713	0.4405	195.0	270.0	0.0008	0
I-IM	-1.882	0.0598	141.0	324.0	0.0008	0
I-RIM	-3.0955	0.002	82.0	383.0	0.0008	0
I-TM	-3.0338	0.0024	85.0	380.0	0.0008	0
I-HMTS	-1.9231	0.0545	326.0	139.0	0.0008	0
I-TMHS	-0.7713	0.4405	270.0	195.0	0.0008	0
I-FMTH	-1.5529	0.1204	308.0	157.0	0.0008	0
FM-T	-2.1905	0.0285	126.0	339.0	0.0008	0
FM-HM	-1.121	0.2623	287.0	178.0	0.0008	0
FM-IM	-1.2855	0.1986	170.0	295.0	0.0008	0
FM-RIM	-3.2395	0.0012	75.0	390.0	0.0008	0
FM-TM	-1.9026	0.0571	140.0	325.0	0.0008	0
FM-HMTS	-2.3345	0.0196	346.0	119.0	0.0008	0
FM-TMHS	-1.4912	0.1359	305.0	160.0	0.0008	0
FM-FMTH	-2.3139	0.0207	345.0	120.0	0.0008	0
T-HM	-2.2934	0.0218	344.0	121.0	0.0008	0
T-IM	-0.833	0.4048	273.0	192.0	0.0008	0
T-RIM	-0.6685	0.5038	200.0	265.0	0.0008	0
T-TM	-0.4217	0.6733	253.0	212.0	0.0008	0
T-HMTS	-4.597	0.0	456.0	9.0	0.0008	-0.8393
T-TMHS	-3.3835	0.0007	397.0	68.0	0.0008	-0.6177
T-FMTH	-3.5892	0.0003	407.0	58.0	0.0008	-0.6553
HM-IM	-1.3061	0.1915	169.0	296.0	0.0008	0
HM-RIM	-3.3835	0.0007	68.0	397.0	0.0008	-0.6177
HM-TM	-2.0054	0.0449	135.0	330.0	0.0008	0
HM-HMTS	-2.3756	0.0175	348.0	117.0	0.0008	0
HM-TMHS	-1.0593	0.2895	284.0	181.0	0.0008	0
HM-FMTH	-2.3551	0.0185	347.0	118.0	0.0008	0
IM-RIM	-1.738	0.0822	148.0	317.0	0.0008	0
IM-TM	-0.9976	0.3185	184.0	281.0	0.0008	0
IM-HMTS	-2.4991	0.0125	354.0	111.0	0.0008	0
IM-TMHS	-2.4991	0.0125	354.0	111.0	0.0008	0
IM-FMTH	-2.1288	0.0333	336.0	129.0	0.0008	0
RIM-TM	-0.7919	0.4284	271.0	194.0	0.0008	0
RIM-HMTS	-4.1445	0.0	434.0	31.0	0.0008	-0.7567
RIM-TMHS	-4.4325	0.0	448.0	17.0	0.0008	-0.8093
RIM-FMTH	-4.0211	0.0001	428.0	37.0	0.0008	-0.7342
TM-HMTS	-3.8566	0.0001	420.0	45.0	0.0008	-0.7041
TM-TMHS	-3.5686	0.0004	406.0	59.0	0.0008	-0.6515
TM-FMTH	-3.3424	0.0008	395.0	70.0	0.0008	0
HMTS-TMHS	-0.8536	0.3933	191.0	274.0	0.0008	0
HMTS-FMTH	-0.8124	0.4165	193.0	272.0	0.0008	0
TMHS-FMTH	-0.1543	0.8774	240.0	225.0	0.0008	0

Table A.32: Wilcoxon Signed Ranks - Easy Scenario

Alg1-Alg2	**Z**	**P-value**	**P rank**	**N rank**	**Alpha**	**R**
SGP-TH	-0.6685	0.5038	265.0	200.0	0.0008	0
SGP-I	-2.1083	0.035	130.0	335.0	0.0008	0
SGP-FM	-3.8977	0.0001	422.0	43.0	0.0008	-0.7116
SGP-T	-4.0828	0.0	34.0	431.0	0.0008	-0.7454
SGP-HM	-3.7126	0.0002	413.0	52.0	0.0008	-0.6778
SGP-IM	-2.787	0.0053	368.0	97.0	0.0008	0
SGP-RIM	-4.4736	0.0	15.0	450.0	0.0008	-0.8168
SGP-TM	-0.3599	0.7189	250.0	215.0	0.0008	0
SGP-HMTS	-3.2395	0.0012	390.0	75.0	0.0008	0
SGP-TMHS	-3.4246	0.0006	399.0	66.0	0.0008	-0.6252
SGP-FMTH	-2.5196	0.0117	355.0	110.0	0.0008	0
TH-I	-1.9026	0.0571	140.0	325.0	0.0008	0
TH-FM	-3.3629	0.0008	396.0	69.0	0.0008	0
TH-T	-4.0622	0.0	35.0	430.0	0.0008	-0.7417
TH-HM	-3.5686	0.0004	406.0	59.0	0.0008	-0.6515
TH-IM	-2.4579	0.014	352.0	113.0	0.0008	0
TH-RIM	-4.597	0.0	9.0	456.0	0.0008	-0.8393
TH-TM	-0.216	0.829	243.0	222.0	0.0008	0
TH-HMTS	-2.787	0.0053	368.0	97.0	0.0008	0
TH-TMHS	-3.1984	0.0014	388.0	77.0	0.0008	0
TH-FMTH	-2.643	0.0082	361.0	104.0	0.0008	0
I-FM	-4.1857	0.0	436.0	29.0	0.0008	-0.7642
I-T	-3.4041	0.0007	67.0	398.0	0.0008	-0.6215
I-HM	-4.597	0.0	456.0	9.0	0.0008	-0.8393
I-IM	-3.3012	0.001	393.0	72.0	0.0008	0
I-RIM	-3.692	0.0002	53.0	412.0	0.0008	-0.6741
I-TM	-1.6558	0.0978	313.0	152.0	0.0008	0
I-HMTS	-4.0828	0.0	431.0	34.0	0.0008	-0.7454
I-TMHS	-4.3502	0.0	444.0	21.0	0.0008	-0.7942
I-FMTH	-3.9183	0.0001	423.0	42.0	0.0008	-0.7154
FM-T	-4.7616	0.0	1.0	464.0	0.0008	-0.8693
FM-HM	-0.6685	0.5038	265.0	200.0	0.0008	0
FM-IM	-0.4217	0.6733	212.0	253.0	0.0008	0
FM-RIM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
FM-TM	-3.4863	0.0005	63.0	402.0	0.0008	-0.6365
FM-HMTS	-1.1827	0.2369	175.0	290.0	0.0008	0
FM-TMHS	-0.6068	0.544	203.0	262.0	0.0008	0
FM-FMTH	-1.1004	0.2712	179.0	286.0	0.0008	0
T-HM	-4.741	0.0	463.0	2.0	0.0008	-0.8656
T-IM	-4.2268	0.0	438.0	27.0	0.0008	-0.7717
T-RIM	-0.9976	0.3185	184.0	281.0	0.0008	0
T-TM	-3.836	0.0001	419.0	46.0	0.0008	-0.7004
T-HMTS	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
T-TMHS	-4.7616	0.0	464.0	1.0	0.0008	-0.8693
T-FMTH	-4.4942	0.0	451.0	14.0	0.0008	-0.8205
HM-IM	-1.4706	0.1414	161.0	304.0	0.0008	0
HM-RIM	-4.7616	0.0	1.0	464.0	0.0008	-0.8693
HM-TM	-3.6509	0.0003	55.0	410.0	0.0008	-0.6666
HM-HMTS	-1.6146	0.1064	154.0	311.0	0.0008	0

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
HM-TMHS	-1.2238	0.221	173.0	292.0	0.0008	0
HM-FMTH	-1.9026	0.0571	140.0	325.0	0.0008	0
IM-RIM	-4.6176	0.0	8.0	457.0	0.0008	-0.8431
IM-TM	-2.6225	0.0087	105.0	360.0	0.0008	0
IM-HMTS	-0.9564	0.3389	186.0	279.0	0.0008	0
IM-TMHS	-0.3394	0.7343	216.0	249.0	0.0008	0
IM-FMTH	-0.4011	0.6884	213.0	252.0	0.0008	0
RIM-TM	-4.6587	0.0	459.0	6.0	0.0008	-0.8506
RIM-HMTS	-4.7616	0.0	464.0	1.0	0.0008	-0.8693
RIM-TMHS	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
RIM-FMTH	-4.741	0.0	463.0	2.0	0.0008	-0.8656
TM-HMTS	-2.6225	0.0087	360.0	105.0	0.0008	0
TM-TMHS	-3.3424	0.0008	395.0	70.0	0.0008	0
TM-FMTH	-2.7664	0.0057	367.0	98.0	0.0008	0
HMTS-TMHS	-0.5862	0.5577	261.0	204.0	0.0008	0
HMTS-FMTH	-0.6068	0.544	262.0	203.0	0.0008	0
TMHS-FMTH	-0.4422	0.6583	211.0	254.0	0.0008	0

Table A.33: Wilcoxon Signed Ranks - Medium Scenario

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-2.0877	0.0368	334.0	131.0	0.0008	0
SGP-I	-0.977	0.3286	185.0	280.0	0.0008	0
SGP-FM	-3.2601	0.0011	391.0	74.0	0.0008	0
SGP-T	-4.2885	0.0	24.0	441.0	0.0008	-0.783
SGP-HM	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
SGP-IM	-4.1034	0.0	432.0	33.0	0.0008	-0.7492
SGP-RIM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
SGP-TM	-1.9437	0.0519	327.0	138.0	0.0008	0
SGP-HMTS	-3.8771	0.0001	421.0	44.0	0.0008	-0.7079
SGP-TMHS	-4.1445	0.0	434.0	31.0	0.0008	-0.7567
SGP-FMTH	-3.3835	0.0007	397.0	68.0	0.0008	-0.6177
TH-I	-2.787	0.0053	97.0	368.0	0.0008	0
TH-FM	-2.7047	0.0068	364.0	101.0	0.0008	0
TH-T	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
TH-HM	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
TH-IM	-3.1572	0.0016	386.0	79.0	0.0008	0
TH-RIM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
TH-TM	-1.1827	0.2369	290.0	175.0	0.0008	0
TH-HMTS	-3.7126	0.0002	413.0	52.0	0.0008	-0.6778
TH-TMHS	-3.1572	0.0016	386.0	79.0	0.0008	0
TH-FMTH	-2.5196	0.0117	355.0	110.0	0.0008	0
I-FM	-3.692	0.0002	412.0	53.0	0.0008	-0.6741
I-T	-3.3218	0.0009	71.0	394.0	0.0008	0
I-HM	-4.4942	0.0	451.0	14.0	0.0008	-0.8205
I-IM	-4.0417	0.0001	429.0	36.0	0.0008	-0.7379
I-RIM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
I-TM	-2.5402	0.0111	356.0	109.0	0.0008	0
I-HMTS	-4.4119	0.0	447.0	18.0	0.0008	-0.8055

Alg1-Alg2	**Z**	**P-value**	**P rank**	**N rank**	**Alpha**	**R**
I-TMHS	-4.2474	0.0	439.0	26.0	0.0008	-0.7755
I-FMTH	-4.3708	0.0	445.0	20.0	0.0008	-0.798
FM-T	-4.7616	0.0	1.0	464.0	0.0008	-0.8693
FM-HM	-2.2317	0.0256	341.0	124.0	0.0008	0
FM-IM	-0.4217	0.6733	212.0	253.0	0.0008	0
FM-RIM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
FM-TM	-2.026	0.0428	134.0	331.0	0.0008	0
FM-HMTS	-0.5245	0.5999	258.0	207.0	0.0008	0
FM-TMHS	-0.0514	0.959	230.0	235.0	0.0008	0
FM-FMTH	-0.1543	0.8774	225.0	240.0	0.0008	0
T-HM	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
T-IM	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
T-RIM	-4.7616	0.0	1.0	464.0	0.0008	-0.8693
T-TM	-4.6587	0.0	459.0	6.0	0.0008	-0.8506
T-HMTS	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
T-TMHS	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
T-FMTH	-4.7616	0.0	464.0	1.0	0.0008	-0.8693
HM-IM	-2.8693	0.0041	93.0	372.0	0.0008	0
HM-RIM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
HM-TM	-3.548	0.0004	60.0	405.0	0.0008	-0.6478
HM-HMTS	-2.0671	0.0387	132.0	333.0	0.0008	0
HM-TMHS	-2.3756	0.0175	117.0	348.0	0.0008	0
HM-FMTH	-2.643	0.0082	104.0	361.0	0.0008	0
IM-RIM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
IM-TM	-2.4579	0.014	113.0	352.0	0.0008	0
IM-HMTS	-0.3188	0.7499	248.0	217.0	0.0008	0
IM-TMHS	-0.0514	0.959	230.0	235.0	0.0008	0
IM-FMTH	-0.3599	0.7189	215.0	250.0	0.0008	0
RIM-TM	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
RIM-HMTS	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
RIM-TMHS	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
RIM-FMTH	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
TM-HMTS	-2.3756	0.0175	348.0	117.0	0.0008	0
TM-TMHS	-2.2934	0.0218	344.0	121.0	0.0008	0
TM-FMTH	-1.7792	0.0752	319.0	146.0	0.0008	0
HMTS-TMHS	-0.2571	0.7971	220.0	245.0	0.0008	0
HMTS-FMTH	-0.8124	0.4165	193.0	272.0	0.0008	0
TMHS-FMTH	-0.3394	0.7343	249.0	216.0	0.0008	0

TABLE A.34: Wilcoxon Signed Ranks - Hard Scenario

Classification

Descriptive Statistics

Offline Performance

Algorithm	Min	Max	Mean	Std	Skewness	Kurtosis
SGP	0.7027	0.8236	0.7637	0.0257	-0.1066	0.1037
FM	0.6794	0.8055	0.7519	0.0306	-0.5532	-0.4429
TH	0.7158	0.8247	0.7657	0.0316	0.3334	-1.12
Immigrants	0.7071	0.8462	0.7829	0.0341	-0.2886	-0.4515
T	0.7223	0.8395	0.7784	0.0297	-0.1316	-0.6439
TM	0.7231	0.8198	0.7721	0.0288	-0.0622	-1.0291
HM	0.6701	0.8141	0.76	0.0335	-0.4908	-0.0595
IM	0.6517	0.8149	0.75	0.0316	-0.7475	1.5302
RIM	0.6848	0.7814	0.7402	0.0226	-0.3618	-0.1581
TMHS	0.7328	0.8348	0.7792	0.0216	0.2382	0.233
HMTS	0.7282	0.8302	0.7833	0.0223	-0.2975	-0.0921
FMTH	0.7237	0.8203	0.7744	0.0254	0.0143	-0.821

Table A.35: Offline Performance - Easy Scenario

Algorithm	Min	Max	Mean	Std	Skewness	Kurtosis
SGP	0.3855	0.4161	0.3998	0.0077	0.1187	-0.4102
FM	0.3888	0.4343	0.4094	0.0106	0.4999	-0.081
TH	0.3921	0.4235	0.4077	0.009	-0.0909	-1.0131
Immigrants	0.4011	0.4302	0.4138	0.0075	0.4706	-0.3297
T	0.3901	0.4229	0.4072	0.0078	0.0456	-0.4932
TM	0.3987	0.4429	0.4188	0.0113	0.2327	-0.6719
HM	0.3969	0.4329	0.4157	0.0087	-0.3473	-0.1838
IM	0.389	0.4211	0.4028	0.0088	0.2347	-1.0766
RIM	0.3851	0.4363	0.406	0.0113	0.6203	0.2485
TMHS	0.4021	0.4323	0.4159	0.007	0.2176	-0.3888
HMTS	0.3968	0.4387	0.415	0.0079	0.2657	1.3005
FMTH	0.3946	0.4307	0.4156	0.008	-0.4624	-0.1468

Table A.36: Offline Performance - Medium Scenario

Algorithm	Min	Max	Mean	Std	Skewness	Kurtosis
SGP	0.3716	0.3814	0.3759	0.0019	0.4762	0.7915
FM	0.3807	0.3888	0.3849	0.0021	0.0914	-0.7612
TH	0.3744	0.3824	0.378	0.0021	0.2715	-0.8087
Immigrants	0.3757	0.3813	0.3786	0.0013	-0.3095	-0.248
T	0.3736	0.382	0.3777	0.0022	0.0271	-0.8334
TM	0.3794	0.3902	0.3861	0.0023	-0.6946	0.6779
HM	0.3825	0.3906	0.3868	0.0019	-0.4838	-0.2127
IM	0.3798	0.3853	0.3829	0.0013	-0.1257	-0.362
RIM	0.3796	0.3858	0.3821	0.0013	0.5159	0.4469

Algorithm	Min	Max	Mean	Std	Skewness	Kurtosis
TMHS	0.3819	0.3897	0.3863	0.0018	-0.6264	0.0708
HMTS	0.3841	0.3904	0.3866	0.0016	0.4889	-0.7358
FMTH	0.3811	0.3895	0.3859	0.0019	-0.3426	0.034

Table A.37: Offline Performance - Hard Scenario

Best of Generation

Algorithm	Min	Max	Mean	Std	Skewness	Kurtosis
SGP	0.7157	0.8529	0.7831	0.0298	0.0724	0.3004
FM	0.6863	0.8165	0.762	0.0329	-0.477	-0.504
TH	0.7253	0.8444	0.7823	0.034	0.1667	-0.983
Immigrants	0.7222	0.8878	0.8028	0.0384	-0.168	-0.4801
T	0.7371	0.8725	0.8022	0.0353	-0.1551	-0.7319
TM	0.7293	0.8323	0.7833	0.0301	-0.1632	-0.97
HM	0.6792	0.827	0.7709	0.0355	-0.4639	-0.2976
IM	0.6566	0.8257	0.7604	0.0337	-0.659	1.333
RIM	0.6903	0.7934	0.7491	0.0245	-0.2864	-0.2503
TMHS	0.7439	0.8466	0.7941	0.0239	0.1054	-0.408
HMTS	0.735	0.8503	0.7973	0.0241	-0.503	0.3005
FMTH	0.7331	0.8399	0.7868	0.0278	0.0523	-0.7636

Table A.38: BoG - Easy Scenario

Algorithm	Min	Max	Mean	Std	Skewness	Kurtosis
SGP	0.7943	0.8672	0.8278	0.0178	0.2298	-0.4087
FM	0.7877	0.8974	0.8395	0.0256	0.3498	-0.2554
TH	0.815	0.885	0.8512	0.0206	-0.0861	-1.0213
Immigrants	0.8335	0.9009	0.863	0.0176	0.3645	-0.4259
T	0.8084	0.8825	0.8492	0.0179	0.0007	-0.7628
TM	0.818	0.9111	0.8616	0.0249	0.2975	-0.7431
HM	0.8101	0.893	0.8547	0.0204	-0.3021	-0.3885
IM	0.7925	0.8635	0.8228	0.0202	0.295	-1.0711
RIM	0.7834	0.8941	0.8323	0.0251	0.4572	-0.2108
TMHS	0.8243	0.8949	0.8565	0.0169	0.3005	-0.4761
HMTS	0.8103	0.9013	0.8539	0.0174	-0.0045	0.9343
FMTH	0.8223	0.8866	0.8569	0.0177	-0.2177	-0.9811

Table A.39: BoG - Medium Scenario

Algorithm	Min	Max	Mean	Std	Skewness	Kurtosis
SGP	0.7934	0.8186	0.8054	0.0061	0.4057	-0.4123
FM	0.8002	0.8289	0.8142	0.0084	0.2759	-1.075
TH	0.8066	0.8309	0.818	0.007	0.0491	-1.3032
Immigrants	0.8023	0.825	0.8144	0.004	-0.3124	2.4554
T	0.8025	0.8232	0.8131	0.0065	-0.2652	-1.3188
TM	0.8088	0.8314	0.82	0.0077	0.0289	-1.6492
HM	0.8136	0.838	0.8255	0.0078	-0.0297	-1.379

Algorithm	Min	Max	Mean	Std	Skewness	Kurtosis
IM	0.7904	0.8106	0.8039	0.0045	-1.0953	1.0299
RIM	0.7944	0.8058	0.7998	0.0031	0.3911	-0.9978
TMHS	0.8102	0.8287	0.8197	0.0058	0.0196	-1.4461
HMTS	0.8118	0.8296	0.8211	0.005	-0.2911	-1.0442
FMTH	0.8046	0.8281	0.818	0.0061	-0.279	-0.953

Table A.40: BoG - Hard Scenario

Normality Tests

Offline Performance

Algorithm	D	P-value
SGP	0.3922	0.0001
FM	0.3854	0.0002
TH	0.4564	0.0
Immigrants	0.3502	0.0009
T	0.35	0.0009
TM	0.3673	0.0004
HM	0.4502	0.0
IM	0.403	0.0001
RIM	0.4446	0.0
TMHS	0.3295	0.0021
HMTS	0.4067	0.0001
FMTH	0.4179	0.0

Table A.41: Kolmogorov-Smirnov - Easy Scenario

Algorithm	D	P-value
SGP	0.3801	0.0002
FM	0.3468	0.001
TH	0.3971	0.0001
Immigrants	0.3494	0.0009
T	0.4304	0.0
TM	0.3839	0.0002
HM	0.4156	0.0
IM	0.4146	0.0
RIM	0.3622	0.0005
TMHS	0.4192	0.0
HMTS	0.3813	0.0002
FMTH	0.3986	0.0001

Table A.42: Kolmogorov-Smirnov - Medium Scenario

Algorithm	D	P-value
SGP	0.3594	0.0006
FM	0.3544	0.0007

Algorithm	D	P-value
TH	0.3911	0.0001
Immigrants	0.3619	0.0005
T	0.4293	0.0
TM	0.4205	0.0
HM	0.4112	0.0
IM	0.3791	0.0002
RIM	0.3513	0.0008
TMHS	0.4244	0.0
HMTS	0.4634	0.0
FMTH	0.3549	0.0007

Table A.43: Kolmogorov-Smirnov - Hard Scenario

Best of Generation

Algorithm	D	P-value
SGP	0.3319	0.0019
FM	0.392	0.0001
TH	0.3763	0.0003
Immigrants	0.3435	0.0012
T	0.3892	0.0001
TM	0.3491	0.0009
HM	0.4509	0.0
IM	0.3464	0.001
RIM	0.3953	0.0001
TMHS	0.3514	0.0008
HMTS	0.4432	0.0
FMTH	0.4004	0.0001

Table A.44: Kolmogorov-Smirnov - Easy Scenario

Algorithm	D	P-value
SGP	0.4054	0.0001
FM	0.4039	0.0001
TH	0.3963	0.0001
Immigrants	0.3848	0.0002
T	0.4713	0.0
TM	0.3606	0.0005
HM	0.3931	0.0001
IM	0.4381	0.0
RIM	0.4012	0.0001
TMHS	0.4471	0.0
HMTS	0.3524	0.0008
FMTH	0.3996	0.0001

Table A.45: Kolmogorov-Smirnov - Medium Scenario

Algorithm	D	P-value
SGP	0.4093	0.0
FM	0.4487	0.0
TH	0.4247	0.0
Immigrants	0.3098	0.0047
T	0.5155	0.0
TM	0.4647	0.0
HM	0.4424	0.0
IM	0.4661	0.0
RIM	0.4586	0.0
TMHS	0.4399	0.0
HMTS	0.4243	0.0
FMTH	0.4209	0.0

Table A.46: Kolmogorov-Smirnov - Hard Scenario

Simple Approaches

Offline Performance

Algorithms	Scenario	Chi-prob	P-value
SGP, tH, Immigrants, fM	Easy	13.0936	0.0044
SGP, tH, Immigrants, fM	Medium	28.96	0.0
SGP, tH, Immigrants, fM	Hard	69.52	0.0

Table A.47: Friedman's Anova

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-0.0541	0.9569	215.0	220.0	0.0167	0
SGP-Immigrants	-2.2317	0.0256	124.0	341.0	0.0167	0
SGP-FM	-1.4706	0.1414	304.0	161.0	0.0167	0

Table A.48: Offline Performance - Easy

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-3.2807	0.001	73.0	392.0	0.0167	-0.599
SGP-Immigrants	-4.4119	0.0	18.0	447.0	0.0167	-0.8055
SGP-FM	-3.7743	0.0002	49.0	416.0	0.0167	-0.6891

Table A.49: Offline Performance - Medium

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-3.3218	0.0009	71.0	394.0	0.0167	-0.6065
SGP-Immigrants	-4.2268	0.0	27.0	438.0	0.0167	-0.7717
SGP-FM	-4.7821	0.0	0.0	465.0	0.0167	-0.8731

Table A.50: Offline Performance - Hard

Best of Generation

Algorithms	Scenario	Chi-prob	P-value
SGP, tH, Immigrants, fM	Easy	18.913	0.0003
SGP, tH, Immigrants, fM	Medium	31.0	0.0
SGP, tH, Immigrants, fM	Hard	34.76	0.0

Table A.51: Friedman's Anova

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-0.4	0.6891	236.0	199.0	0.0167	0
SGP-Immigrants	-2.2317	0.0256	124.0	341.0	0.0167	0
SGP-FM	-2.5608	0.0104	357.0	108.0	0.0167	-0.4675

Table A.52: BoG - Easy

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-3.9183	0.0001	42.0	423.0	0.0167	-0.7154
SGP-Immigrants	-4.4119	0.0	18.0	447.0	0.0167	-0.8055
SGP-FM	-2.3139	0.0207	120.0	345.0	0.0167	0

Table A.53: BoG - Medium

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-4.4119	0.0	18.0	447.0	0.0167	-0.8055
SGP-Immigrants	-4.3708	0.0	20.0	445.0	0.0167	-0.798
SGP-FM	-4.5148	0.0	13.0	452.0	0.0167	-0.8243

Table A.54: Wilcoxon Signed Ranks - BoG Hard

Hybrid Approaches

Offline Performance

Algorithms	Scenario	Chi-prob	P-value
T, SGP, TH, Immigrants, FM	Easy	17.3155	0.0017
HM, TH, FM	Easy	3.6807	0.1588
IM, Immigrants, FM	Easy	20.2667	0.0
RIM, IM, Immigrants, FM	Easy	28.48	0.0
TM, FM, T	Easy	6.0667	0.0482
HMTS, SGP, HM, T	Easy	16.68	0.0008
TMHS, SGP, TM, TH	Easy	3.8829	0.2744
FMTH, FM, TH, T	Easy	12.68	0.0054
T, SGP, TH, Immigrants, FM	Medium	30.4267	0.0
HM, TH, FM	Medium	11.4	0.0033
IM, Immigrants, FM	Medium	14.0667	0.0009
RIM, IM, Immigrants, FM	Medium	19.0	0.0003
TM, FM, T	Medium	11.4	0.0033

Algorithms	Scenario	Chi-prob	P-value
HMTS, SGP, HM, T	Medium	33.76	0.0
TMHS, SGP, TM, TH	Medium	36.0	0.0
FMTH, FM, TH, T	Medium	13.96	0.003
T, SGP, TH, Immigrants, FM	Hard	74.2667	0.0
HM, TH, FM	Hard	47.4	0.0
IM, Immigrants, FM	Hard	49.2667	0.0
RIM, IM, Immigrants, FM	Hard	66.52	0.0
TM, FM, T	Hard	43.4	0.0
HMTS, SGP, HM, T	Hard	74.32	0.0
TMHS, SGP, TM, TH	Hard	75.28	0.0
FMTH, FM, TH, T	Hard	70.2	0.0

Table A.55: Friedman's Anova

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
T-SGP	-2.0465	0.0407	332.0	133.0	0.0125	0
T-TH	-1.4501	0.147	303.0	162.0	0.0125	0
T-Immigrants	-0.2777	0.7813	219.0	246.0	0.0125	0
T-FM	-2.7664	0.0057	367.0	98.0	0.0125	-0.5051
HM-TH	-0.6479	0.517	201.0	264.0	0.025	0
HM-FM	-1.5893	0.112	291.0	144.0	0.025	0
IM-Immigrants	-3.4452	0.0006	65.0	400.0	0.025	-0.629
IM-FM	-0.2571	0.7971	220.0	245.0	0.025	0
RIM-IM	-1.9643	0.0495	137.0	328.0	0.0167	0
RIM-Immigrants	-4.124	0.0	32.0	433.0	0.0167	-0.7529
RIM-FM	-1.7586	0.0786	147.0	318.0	0.0167	0
TM-FM	-2.2522	0.0243	342.0	123.0	0.025	-0.4112
TM-T	-0.4422	0.6583	211.0	254.0	0.025	0
HMTS-SGP	-2.6019	0.0093	359.0	106.0	0.0167	-0.475
HMTS-HM	-2.7664	0.0057	367.0	98.0	0.0167	-0.5051
HMTS-T	-1.1415	0.2536	288.0	177.0	0.0167	0
TMHS-SGP	-2.2522	0.0243	342.0	123.0	0.0167	0
TMHS-TM	-0.9359	0.3493	278.0	187.0	0.0167	0
TMHS-TH	-1.6763	0.0937	314.0	151.0	0.0167	0
FMTH-FM	-2.9927	0.0028	378.0	87.0	0.0167	-0.5464
FMTH-TH	-1.3267	0.1846	297.0	168.0	0.0167	0
FMTH-T	-0.5451	0.5857	206.0	259.0	0.0167	0

Table A.56: Offline Performance - Easy

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
T-SGP	-3.1778	0.0015	387.0	78.0	0.0125	-0.5802
T-TH	-0.2982	0.7655	218.0	247.0	0.0125	0
T-Immigrants	-2.8281	0.0047	95.0	370.0	0.0125	-0.5163
T-FM	-0.7096	0.4779	198.0	267.0	0.0125	0
HM-TH	-3.1367	0.0017	385.0	80.0	0.025	-0.5727
HM-FM	-2.7253	0.0064	365.0	100.0	0.025	-0.4976
IM-Immigrants	-3.8566	0.0001	45.0	420.0	0.025	-0.7041
IM-FM	-2.2934	0.0218	121.0	344.0	0.025	-0.4187

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
RIM-IM	-1.0181	0.3086	282.0	183.0	0.0167	0
RIM-Immigrants	-2.8281	0.0047	95.0	370.0	0.0167	-0.5163
RIM-FM	-1.4706	0.1414	161.0	304.0	0.0167	0
TM-FM	-2.6842	0.0073	363.0	102.0	0.025	-0.4901
TM-T	-3.7949	0.0001	417.0	48.0	0.025	-0.6928
HMTS-SGP	-4.5353	0.0	453.0	12.0	0.0167	-0.828
HMTS-HM	-0.2571	0.7971	220.0	245.0	0.0167	0
HMTS-T	-3.4452	0.0006	400.0	65.0	0.0167	-0.629
TMHS-SGP	-4.5765	0.0	455.0	10.0	0.0167	-0.8355
TMHS-TM	-0.9359	0.3493	187.0	278.0	0.0167	0
TMHS-TH	-2.7664	0.0057	367.0	98.0	0.0167	-0.5051
FMTH-FM	-2.3756	0.0175	348.0	117.0	0.0167	0
FMTH-TH	-2.643	0.0082	361.0	104.0	0.0167	-0.4825
FMTH-T	-3.8566	0.0001	420.0	45.0	0.0167	-0.7041

Table A.57: Offline Performance - Medium

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
T-SGP	-2.931	0.0034	375.0	90.0	0.0125	-0.5351
T-TH	-1.0798	0.2802	180.0	285.0	0.0125	0
T-Immigrants	-1.738	0.0822	148.0	317.0	0.0125	0
T-FM	-4.7616	0.0	1.0	464.0	0.0125	-0.8693
HM-TH	-4.7821	0.0	465.0	0.0	0.025	-0.8731
HM-FM	-3.1367	0.0017	385.0	80.0	0.025	-0.5727
IM-Immigrants	-4.7821	0.0	465.0	0.0	0.025	-0.8731
IM-FM	-3.6097	0.0003	57.0	408.0	0.025	-0.659
RIM-IM	-2.2728	0.023	122.0	343.0	0.0167	0
RIM-Immigrants	-4.7821	0.0	465.0	0.0	0.0167	-0.8731
RIM-FM	-4.1651	0.0	30.0	435.0	0.0167	-0.7604
TM-FM	-2.3551	0.0185	347.0	118.0	0.025	-0.43
TM-T	-4.7821	0.0	465.0	0.0	0.025	-0.8731
HMTS-SGP	-4.7821	0.0	465.0	0.0	0.0167	-0.8731
HMTS-HM	-0.7919	0.4284	194.0	271.0	0.0167	0
HMTS-T	-4.7821	0.0	465.0	0.0	0.0167	-0.8731
TMHS-SGP	-4.7821	0.0	465.0	0.0	0.0167	-0.8731
TMHS-TM	-0.3599	0.7189	250.0	215.0	0.0167	0
TMHS-TH	-4.7821	0.0	465.0	0.0	0.0167	-0.8731
FMTH-FM	-2.2728	0.023	343.0	122.0	0.0167	0
FMTH-TH	-4.7821	0.0	465.0	0.0	0.0167	-0.8731
FMTH-T	-4.7821	0.0	465.0	0.0	0.0167	-0.8731

Table A.58: Offline Performance - Hard

Best of Generation

Algorithms	Scenario	Chi-prob	P-value
T, SGP, TH, Immigrants, FM	Easy	23.7529	0.0001
HM, TH, FM	Easy	7.3782	0.025

Algorithms	Scenario	Chi-prob	P-value
IM, Immigrants, FM	Easy	20.0	0.0
RIM, IM, Immigrants, FM	Easy	30.44	0.0
TM, FM, T	Easy	8.2667	0.016
HMTS, SGP, HM, T	Easy	17.0	0.0007
TMHS, SGP, TM, TH	Easy	3.0201	0.3885
FMTH, FM, TH, T	Easy	13.84	0.0031
T, SGP, TH, Immigrants, FM	Medium	35.2533	0.0
HM, TH, FM	Medium	9.6	0.0082
IM, Immigrants, FM	Medium	28.4667	0.0
RIM, IM, Immigrants, FM	Medium	32.68	0.0
TM, FM, T	Medium	7.4	0.0247
HMTS, SGP, HM, T	Medium	25.24	0.0
TMHS, SGP, TM, TH	Medium	26.28	0.0
FMTH, FM, TH, T	Medium	6.68	0.0828
T, SGP, TH, Immigrants, FM	Hard	40.3467	0.0
HM, TH, FM	Hard	28.4667	0.0
IM, Immigrants, FM	Hard	28.8667	0.0
RIM, IM, Immigrants, FM	Hard	59.08	0.0
TM, FM, T	Hard	15.2667	0.0005
HMTS, SGP, HM, T	Hard	61.64	0.0
TMHS, SGP, TM, TH	Hard	34.44	0.0
FMTH, FM, TH, T	Hard	9.72	0.0211

Table A.59: Friedman's Anova

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
T-SGP	-2.4373	0.0148	351.0	114.0	0.0125	0
T-TH	-1.9643	0.0495	328.0	137.0	0.0125	0
T-Immigrants	-0.1131	0.9099	238.0	227.0	0.0125	0
T-FM	-3.3835	0.0007	397.0	68.0	0.0125	-0.6177
HM-TH	-1.594	0.1109	155.0	310.0	0.025	0
HM-FM	-1.5893	0.112	291.0	144.0	0.025	0
IM-Immigrants	-3.7537	0.0002	50.0	415.0	0.025	-0.6853
IM-FM	-0.1954	0.8451	223.0	242.0	0.025	0
RIM-IM	-2.0054	0.0449	135.0	330.0	0.0167	0
RIM-Immigrants	-4.2268	0.0	27.0	438.0	0.0167	-0.7717
RIM-FM	-1.6969	0.0897	150.0	315.0	0.0167	0
TM-FM	-2.2728	0.023	343.0	122.0	0.025	-0.415
TM-T	-1.9437	0.0519	138.0	327.0	0.025	0
HMTS-SGP	-2.4168	0.0157	350.0	115.0	0.0167	-0.4412
HMTS-HM	-2.8693	0.0041	372.0	93.0	0.0167	-0.5239
HMTS-T	-0.2365	0.813	221.0	244.0	0.0167	0
TMHS-SGP	-1.8203	0.0687	321.0	144.0	0.0167	0
TMHS-TM	-1.4501	0.147	303.0	162.0	0.0167	0
TMHS-TH	-1.3678	0.1714	299.0	166.0	0.0167	0
FMTH-FM	-3.1367	0.0017	385.0	80.0	0.0167	-0.5727
FMTH-TH	-0.7713	0.4405	270.0	195.0	0.0167	0
FMTH-T	-2.0054	0.0449	135.0	330.0	0.0167	0

Table A.60: BoG - Easy

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
T-SGP	-3.9388	0.0001	424.0	41.0	0.0125	-0.7191
T-TH	-0.6273	0.5304	202.0	263.0	0.0125	0
T-Immigrants	-2.4168	0.0157	115.0	350.0	0.0125	0
T-FM	-1.9026	0.0571	325.0	140.0	0.0125	0
HM-TH	-0.8536	0.3933	274.0	191.0	0.025	0
HM-FM	-2.7047	0.0068	364.0	101.0	0.025	-0.4938
IM-Immigrants	-4.6176	0.0	8.0	457.0	0.025	-0.8431
IM-FM	-2.3139	0.0207	120.0	345.0	0.025	-0.4225
RIM-IM	-1.4706	0.1414	304.0	161.0	0.0167	0
RIM-Immigrants	-3.9594	0.0001	40.0	425.0	0.0167	-0.7229
RIM-FM	-1.265	0.2059	171.0	294.0	0.0167	0
TM-FM	-2.8281	0.0047	370.0	95.0	0.025	-0.5163
TM-T	-2.0671	0.0387	333.0	132.0	0.025	0
HMTS-SGP	-3.9594	0.0001	425.0	40.0	0.0167	-0.7229
HMTS-HM	-0.2777	0.7813	219.0	246.0	0.0167	0
HMTS-T	-0.7302	0.4653	268.0	197.0	0.0167	0
TMHS-SGP	-3.9388	0.0001	424.0	41.0	0.0167	-0.7191
TMHS-TM	-0.7919	0.4284	194.0	271.0	0.0167	0
TMHS-TH	-0.7096	0.4779	267.0	198.0	0.0167	0
FMTH-FM	-2.7047	0.0068	364.0	101.0	0.0167	-0.4938
FMTH-TH	-0.8947	0.3709	276.0	189.0	0.0167	0
FMTH-T	-1.9848	0.0472	329.0	136.0	0.0167	0

Table A.61: BoG - Medium

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
T-SGP	-3.9388	0.0001	424.0	41.0	0.0125	-0.7191
T-TH	-2.5402	0.0111	109.0	356.0	0.0125	-0.4638
T-Immigrants	-0.6479	0.517	201.0	264.0	0.0125	0
T-FM	-0.7096	0.4779	198.0	267.0	0.0125	0
HM-TH	-3.6714	0.0002	411.0	54.0	0.025	-0.6703
HM-FM	-4.4119	0.0	447.0	18.0	0.025	-0.8055
IM-Immigrants	-4.7204	0.0	3.0	462.0	0.025	-0.8618
IM-FM	-4.2679	0.0	25.0	440.0	0.025	-0.7792
RIM-IM	-3.3218	0.0009	71.0	394.0	0.0167	-0.6065
RIM-Immigrants	-4.7821	0.0	0.0	465.0	0.0167	-0.8731
RIM-FM	-4.7204	0.0	3.0	462.0	0.0167	-0.8618
TM-FM	-2.8899	0.0039	373.0	92.0	0.025	-0.5276
TM-T	-3.3835	0.0007	397.0	68.0	0.025	-0.6177
HMTS-SGP	-4.741	0.0	463.0	2.0	0.0167	-0.8656
HMTS-HM	-2.4579	0.014	113.0	352.0	0.0167	-0.4488
HMTS-T	-4.2268	0.0	438.0	27.0	0.0167	-0.7717
TMHS-SGP	-4.6382	0.0	458.0	7.0	0.0167	-0.8468
TMHS-TM	-0.216	0.829	222.0	243.0	0.0167	0
TMHS-TH	-0.9976	0.3185	281.0	184.0	0.0167	0
FMTH-FM	-1.9848	0.0472	329.0	136.0	0.0167	0
FMTH-TH	-0.1131	0.9099	227.0	238.0	0.0167	0
FMTH-T	-2.7459	0.006	366.0	99.0	0.0167	-0.5013

Table A.62: BoG - Hard

Search for the best approach

Offline Performance

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-0.0541	0.9569	215.0	220.0	0.0008	0
SGP-Immigrants	-2.2317	0.0256	124.0	341.0	0.0008	0
SGP-FM	-1.4706	0.1414	304.0	161.0	0.0008	0
SGP-T	-2.0465	0.0407	133.0	332.0	0.0008	0
SGP-HM	-0.3599	0.7189	250.0	215.0	0.0008	0
SGP-IM	-2.026	0.0428	331.0	134.0	0.0008	0
SGP-RIM	-2.7047	0.0068	364.0	101.0	0.0008	0
SGP-TM	-1.594	0.1109	155.0	310.0	0.0008	0
SGP-HMTS	-2.6019	0.0093	106.0	359.0	0.0008	0
SGP-TMHS	-2.2522	0.0243	123.0	342.0	0.0008	0
SGP-FMTH	-1.7586	0.0786	147.0	318.0	0.0008	0
TH-Immigrants	-1.9848	0.0472	136.0	329.0	0.0008	0
TH-FM	-1.738	0.0822	317.0	148.0	0.0008	0
TH-T	-1.4501	0.147	162.0	303.0	0.0008	0
TH-HM	-0.6479	0.517	264.0	201.0	0.0008	0
TH-IM	-1.6763	0.0937	314.0	151.0	0.0008	0
TH-RIM	-2.8899	0.0039	373.0	92.0	0.0008	0
TH-TM	-0.8947	0.3709	189.0	276.0	0.0008	0
TH-HMTS	-2.2111	0.027	125.0	340.0	0.0008	0
TH-TMHS	-1.6763	0.0937	151.0	314.0	0.0008	0
TH-FMTH	-1.3267	0.1846	168.0	297.0	0.0008	0
Immigrants-FM	-3.1572	0.0016	386.0	79.0	0.0008	0
Immigrants-T	-0.2777	0.7813	246.0	219.0	0.0008	0
Immigrants-HM	-2.4373	0.0148	351.0	114.0	0.0008	0
Immigrants-IM	-3.4452	0.0006	400.0	65.0	0.0008	-0.629
Immigrants-RIM	-4.124	0.0	433.0	32.0	0.0008	-0.7529
Immigrants-TM	-1.5118	0.1306	306.0	159.0	0.0008	0
Immigrants-HMTS	-0.1337	0.8936	226.0	239.0	0.0008	0
Immigrants-TMHS	-0.6273	0.5304	263.0	202.0	0.0008	0
Immigrants-FMTH	-0.7096	0.4779	267.0	198.0	0.0008	0
FM-T	-2.7664	0.0057	98.0	367.0	0.0008	0
FM-HM	-1.5893	0.112	144.0	291.0	0.0008	0
FM-IM	-0.2571	0.7971	245.0	220.0	0.0008	0
FM-RIM	-1.7586	0.0786	318.0	147.0	0.0008	0
FM-TM	-2.2522	0.0243	123.0	342.0	0.0008	0
FM-HMTS	-3.7332	0.0002	51.0	414.0	0.0008	-0.6816
FM-TMHS	-3.4041	0.0007	67.0	398.0	0.0008	-0.6215
FM-FMTH	-2.9927	0.0028	87.0	378.0	0.0008	0
T-HM	-1.9437	0.0519	327.0	138.0	0.0008	0
T-IM	-3.3012	0.001	393.0	72.0	0.0008	0
T-RIM	-4.1857	0.0	436.0	29.0	0.0008	-0.7642
T-TM	-0.4422	0.6583	254.0	211.0	0.0008	0
T-HMTS	-1.1415	0.2536	177.0	288.0	0.0008	0
T-TMHS	-0.0514	0.959	230.0	235.0	0.0008	0
T-FMTH	-0.5451	0.5857	259.0	206.0	0.0008	0

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
HM-IM	-1.4295	0.1529	302.0	163.0	0.0008	0
HM-RIM	-2.643	0.0082	361.0	104.0	0.0008	0
HM-TM	-0.9976	0.3185	184.0	281.0	0.0008	0
HM-HMTS	-2.7664	0.0057	98.0	367.0	0.0008	0
HM-TMHS	-2.4373	0.0148	114.0	351.0	0.0008	0
HM-FMTH	-2.0465	0.0407	133.0	332.0	0.0008	0
IM-RIM	-1.9643	0.0495	328.0	137.0	0.0008	0
IM-TM	-3.4246	0.0006	66.0	399.0	0.0008	-0.6252
IM-HMTS	-4.0005	0.0001	38.0	427.0	0.0008	-0.7304
IM-TMHS	-3.3424	0.0008	70.0	395.0	0.0008	0
IM-FMTH	-3.0133	0.0026	86.0	379.0	0.0008	0
RIM-TM	-3.5892	0.0003	58.0	407.0	0.0008	-0.6553
RIM-HMTS	-4.597	0.0	9.0	456.0	0.0008	-0.8393
RIM-TMHS	-4.4119	0.0	18.0	447.0	0.0008	-0.8055
RIM-FMTH	-4.1445	0.0	31.0	434.0	0.0008	-0.7567
TM-HMTS	-1.9643	0.0495	137.0	328.0	0.0008	0
TM-TMHS	-0.9359	0.3493	187.0	278.0	0.0008	0
TM-FMTH	-0.8124	0.4165	193.0	272.0	0.0008	0
HMTS-TMHS	-0.7713	0.4405	270.0	195.0	0.0008	0
HMTS-FMTH	-1.594	0.1109	310.0	155.0	0.0008	0
TMHS-FMTH	-0.7919	0.4284	271.0	194.0	0.0008	0

Table A.63: Offline Performance - Easy

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-3.2807	0.001	73.0	392.0	0.0008	0
SGP-Immigrants	-4.4119	0.0	18.0	447.0	0.0008	-0.8055
SGP-FM	-3.7743	0.0002	49.0	416.0	0.0008	-0.6891
SGP-T	-3.1778	0.0015	78.0	387.0	0.0008	0
SGP-HM	-4.5765	0.0	10.0	455.0	0.0008	-0.8355
SGP-IM	-1.1827	0.2369	175.0	290.0	0.0008	0
SGP-RIM	-2.3756	0.0175	117.0	348.0	0.0008	0
SGP-TM	-4.6999	0.0	4.0	461.0	0.0008	-0.8581
SGP-HMTS	-4.5353	0.0	12.0	453.0	0.0008	-0.828
SGP-TMHS	-4.5765	0.0	10.0	455.0	0.0008	-0.8355
SGP-FMTH	-4.597	0.0	9.0	456.0	0.0008	-0.8393
TH-Immigrants	-2.7459	0.006	99.0	366.0	0.0008	0
TH-FM	-0.6685	0.5038	200.0	265.0	0.0008	0
TH-T	-0.2982	0.7655	247.0	218.0	0.0008	0
TH-HM	-3.1367	0.0017	80.0	385.0	0.0008	0
TH-IM	-1.6352	0.102	312.0	153.0	0.0008	0
TH-RIM	-0.9564	0.3389	279.0	186.0	0.0008	0
TH-TM	-3.4658	0.0005	64.0	401.0	0.0008	-0.6328
TH-HMTS	-2.9721	0.003	88.0	377.0	0.0008	0
TH-TMHS	-2.7664	0.0057	98.0	367.0	0.0008	0
TH-FMTH	-2.643	0.0082	104.0	361.0	0.0008	0
Immigrants-FM	-1.9643	0.0495	328.0	137.0	0.0008	0
Immigrants-T	-2.8281	0.0047	370.0	95.0	0.0008	0
Immigrants-HM	-0.7096	0.4779	198.0	267.0	0.0008	0

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
Immigrants-IM	-3.8566	0.0001	420.0	45.0	0.0008	-0.7041
Immigrants-RIM	-2.8281	0.0047	370.0	95.0	0.0008	0
Immigrants-TM	-1.8614	0.0627	142.0	323.0	0.0008	0
Immigrants-HMTS	-0.8124	0.4165	193.0	272.0	0.0008	0
Immigrants-TMHS	-1.1415	0.2536	177.0	288.0	0.0008	0
Immigrants-FMTH	-1.2238	0.221	173.0	292.0	0.0008	0
FM-T	-0.7096	0.4779	267.0	198.0	0.0008	0
FM-HM	-2.7253	0.0064	100.0	365.0	0.0008	0
FM-IM	-2.2934	0.0218	344.0	121.0	0.0008	0
FM-RIM	-1.4706	0.1414	304.0	161.0	0.0008	0
FM-TM	-2.6842	0.0073	102.0	363.0	0.0008	0
FM-HMTS	-2.2934	0.0218	121.0	344.0	0.0008	0
FM-TMHS	-2.2111	0.027	125.0	340.0	0.0008	0
FM-FMTH	-2.3756	0.0175	117.0	348.0	0.0008	0
T-HM	-3.3012	0.001	72.0	393.0	0.0008	0
T-IM	-1.738	0.0822	317.0	148.0	0.0008	0
T-RIM	-0.7713	0.4405	270.0	195.0	0.0008	0
T-TM	-3.7949	0.0001	48.0	417.0	0.0008	-0.6928
T-HMTS	-3.4452	0.0006	65.0	400.0	0.0008	-0.629
T-TMHS	-3.836	0.0001	46.0	419.0	0.0008	-0.7004
T-FMTH	-3.8566	0.0001	45.0	420.0	0.0008	-0.7041
HM-IM	-4.0622	0.0	430.0	35.0	0.0008	-0.7417
HM-RIM	-2.8899	0.0039	373.0	92.0	0.0008	0
HM-TM	-1.1621	0.2452	176.0	289.0	0.0008	0
HM-HMTS	-0.2571	0.7971	245.0	220.0	0.0008	0
HM-TMHS	-0.1337	0.8936	226.0	239.0	0.0008	0
HM-FMTH	-0.0309	0.9754	231.0	234.0	0.0008	0
IM-RIM	-1.0181	0.3086	183.0	282.0	0.0008	0
IM-TM	-4.3708	0.0	20.0	445.0	0.0008	-0.798
IM-HMTS	-4.1034	0.0	33.0	432.0	0.0008	-0.7492
IM-TMHS	-4.5559	0.0	11.0	454.0	0.0008	-0.8318
IM-FMTH	-3.7949	0.0001	48.0	417.0	0.0008	-0.6928
RIM-TM	-3.3629	0.0008	69.0	396.0	0.0008	0
RIM-HMTS	-2.9516	0.0032	89.0	376.0	0.0008	0
RIM-TMHS	-3.5686	0.0004	59.0	406.0	0.0008	-0.6515
RIM-FMTH	-3.0544	0.0023	84.0	381.0	0.0008	0
TM-HMTS	-1.7792	0.0752	319.0	146.0	0.0008	0
TM-TMHS	-0.9359	0.3493	278.0	187.0	0.0008	0
TM-FMTH	-1.1004	0.2712	286.0	179.0	0.0008	0
HMTS-TMHS	-0.7302	0.4653	197.0	268.0	0.0008	0
HMTS-FMTH	-0.6068	0.544	203.0	262.0	0.0008	0
TMHS-FMTH	-0.2777	0.7813	219.0	246.0	0.0008	0

Table A.64: Offline Performance - Medium

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-3.3218	0.0009	71.0	394.0	0.0008	0
SGP-Immigrants	-4.2268	0.0	27.0	438.0	0.0008	-0.7717
SGP-FM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-T	-2.931	0.0034	90.0	375.0	0.0008	0
SGP-HM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
SGP-IM	-4.7616	0.0	1.0	464.0	0.0008	-0.8693
SGP-RIM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
SGP-TM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
SGP-HMTS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
SGP-TMHS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
SGP-FMTH	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
TH-Immigrants	-1.3884	0.165	165.0	300.0	0.0008	0
TH-FM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
TH-T	-1.0798	0.2802	285.0	180.0	0.0008	0
TH-HM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
TH-IM	-4.741	0.0	2.0	463.0	0.0008	-0.8656
TH-RIM	-4.5353	0.0	12.0	453.0	0.0008	-0.828
TH-TM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
TH-HMTS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
TH-TMHS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
TH-FMTH	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
Immigrants-FM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
Immigrants-T	-1.738	0.0822	317.0	148.0	0.0008	0
Immigrants-HM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
Immigrants-IM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
Immigrants-RIM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
Immigrants-TM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
Immigrants-HMTS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
Immigrants-TMHS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
Immigrants-FMTH	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
FM-T	-4.7616	0.0	464.0	1.0	0.0008	-0.8693
FM-HM	-3.1367	0.0017	80.0	385.0	0.0008	0
FM-IM	-3.6097	0.0003	408.0	57.0	0.0008	-0.659
FM-RIM	-4.1651	0.0	435.0	30.0	0.0008	-0.7604
FM-TM	-2.3551	0.0185	118.0	347.0	0.0008	0
FM-HMTS	-2.8693	0.0041	93.0	372.0	0.0008	0
FM-TMHS	-2.6019	0.0093	106.0	359.0	0.0008	0
FM-FMTH	-2.2728	0.023	122.0	343.0	0.0008	0
T-HM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
T-IM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
T-RIM	-4.7616	0.0	1.0	464.0	0.0008	-0.8693
T-TM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
T-HMTS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
T-TMHS	-4.7616	0.0	1.0	464.0	0.0008	-0.8693
T-FMTH	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
HM-IM	-4.597	0.0	456.0	9.0	0.0008	-0.8393
HM-RIM	-4.6587	0.0	459.0	6.0	0.0008	-0.8506
HM-TM	-0.8536	0.3933	274.0	191.0	0.0008	0
HM-HMTS	-0.7919	0.4284	271.0	194.0	0.0008	0
HM-TMHS	-0.8536	0.3933	274.0	191.0	0.0008	0
HM-FMTH	-1.6969	0.0897	315.0	150.0	0.0008	0
IM-RIM	-2.2728	0.023	343.0	122.0	0.0008	0

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
IM-TM	-4.5148	0.0	13.0	452.0	0.0008	-0.8243
IM-HMTS	-4.7616	0.0	1.0	464.0	0.0008	-0.8693
IM-TMHS	-4.6587	0.0	6.0	459.0	0.0008	-0.8506
IM-FMTH	-4.3913	0.0	19.0	446.0	0.0008	-0.8017
RIM-TM	-4.4736	0.0	15.0	450.0	0.0008	-0.8168
RIM-HMTS	-4.7616	0.0	1.0	464.0	0.0008	-0.8693
RIM-TMHS	-4.6999	0.0	4.0	461.0	0.0008	-0.8581
RIM-FMTH	-4.6382	0.0	7.0	458.0	0.0008	-0.8468
TM-HMTS	-0.6479	0.517	201.0	264.0	0.0008	0
TM-TMHS	-0.3599	0.7189	215.0	250.0	0.0008	0
TM-FMTH	-0.4217	0.6733	253.0	212.0	0.0008	0
HMTS-TMHS	-0.5245	0.5999	258.0	207.0	0.0008	0
HMTS-FMTH	-1.0593	0.2895	284.0	181.0	0.0008	0
TMHS-FMTH	-0.9359	0.3493	278.0	187.0	0.0008	0

Table A.65: Offline Performance - Hard

Best of Generation

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-0.4	0.6891	236.0	199.0	0.0008	0
SGP-Immigrants	-2.2317	0.0256	124.0	341.0	0.0008	0
SGP-FM	-2.5608	0.0104	357.0	108.0	0.0008	0
SGP-T	-2.4373	0.0148	114.0	351.0	0.0008	0
SGP-HM	-1.3884	0.165	300.0	165.0	0.0008	0
SGP-IM	-3.075	0.0021	382.0	83.0	0.0008	0
SGP-RIM	-3.5892	0.0003	407.0	58.0	0.0008	-0.6553
SGP-TM	-0.5039	0.6143	208.0	257.0	0.0008	0
SGP-HMTS	-2.4168	0.0157	115.0	350.0	0.0008	0
SGP-TMHS	-1.8203	0.0687	144.0	321.0	0.0008	0
SGP-FMTH	-0.5862	0.5577	204.0	261.0	0.0008	0
TH-Immigrants	-2.2522	0.0243	123.0	342.0	0.0008	0
TH-FM	-2.4579	0.014	352.0	113.0	0.0008	0
TH-T	-1.9643	0.0495	137.0	328.0	0.0008	0
TH-HM	-1.594	0.1109	310.0	155.0	0.0008	0
TH-IM	-2.2728	0.023	343.0	122.0	0.0008	0
TH-RIM	-3.3835	0.0007	397.0	68.0	0.0008	-0.6177
TH-TM	-0.216	0.829	222.0	243.0	0.0008	0
TH-HMTS	-1.8203	0.0687	144.0	321.0	0.0008	0
TH-TMHS	-1.3678	0.1714	166.0	299.0	0.0008	0
TH-FMTH	-0.7713	0.4405	195.0	270.0	0.0008	0
Immigrants-FM	-3.3835	0.0007	397.0	68.0	0.0008	-0.6177
Immigrants-T	-0.1131	0.9099	227.0	238.0	0.0008	0
Immigrants-HM	-2.9721	0.003	377.0	88.0	0.0008	0
Immigrants-IM	-3.7537	0.0002	415.0	50.0	0.0008	-0.6853
Immigrants-RIM	-4.2268	0.0	438.0	27.0	0.0008	-0.7717
Immigrants-TM	-2.3756	0.0175	348.0	117.0	0.0008	0
Immigrants-HMTS	-0.7507	0.4528	269.0	196.0	0.0008	0
Immigrants-TMHS	-0.9359	0.3493	278.0	187.0	0.0008	0

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
Immigrants-FMTH	-1.265	0.2059	294.0	171.0	0.0008	0
FM-T	-3.3835	0.0007	68.0	397.0	0.0008	-0.6177
FM-HM	-1.5893	0.112	144.0	291.0	0.0008	0
FM-IM	-0.1954	0.8451	242.0	223.0	0.0008	0
FM-RIM	-1.6969	0.0897	315.0	150.0	0.0008	0
FM-TM	-2.2728	0.023	122.0	343.0	0.0008	0
FM-HMTS	-3.836	0.0001	46.0	419.0	0.0008	-0.7004
FM-TMHS	-3.5275	0.0004	61.0	404.0	0.0008	-0.644
FM-FMTH	-3.1367	0.0017	80.0	385.0	0.0008	0
T-HM	-2.7253	0.0064	365.0	100.0	0.0008	0
T-IM	-4.2885	0.0	441.0	24.0	0.0008	-0.783
T-RIM	-4.5353	0.0	453.0	12.0	0.0008	-0.828
T-TM	-1.9437	0.0519	327.0	138.0	0.0008	0
T-HMTS	-0.2365	0.813	244.0	221.0	0.0008	0
T-TMHS	-0.7919	0.4284	271.0	194.0	0.0008	0
T-FMTH	-2.0054	0.0449	330.0	135.0	0.0008	0
HM-IM	-1.265	0.2059	294.0	171.0	0.0008	0
HM-RIM	-2.643	0.0082	361.0	104.0	0.0008	0
HM-TM	-0.9359	0.3493	187.0	278.0	0.0008	0
HM-HMTS	-2.8693	0.0041	93.0	372.0	0.0008	0
HM-TMHS	-2.7047	0.0068	101.0	364.0	0.0008	0
HM-FMTH	-2.1905	0.0285	126.0	339.0	0.0008	0
IM-RIM	-2.0054	0.0449	330.0	135.0	0.0008	0
IM-TM	-3.3424	0.0008	70.0	395.0	0.0008	0
IM-HMTS	-4.1445	0.0	31.0	434.0	0.0008	-0.7567
IM-TMHS	-3.4041	0.0007	67.0	398.0	0.0008	-0.6215
IM-FMTH	-3.075	0.0021	83.0	382.0	0.0008	0
RIM-TM	-3.5686	0.0004	59.0	406.0	0.0008	-0.6515
RIM-HMTS	-4.5765	0.0	10.0	455.0	0.0008	-0.8355
RIM-TMHS	-4.5353	0.0	12.0	453.0	0.0008	-0.828
RIM-FMTH	-4.3091	0.0	23.0	442.0	0.0008	-0.7867
TM-HMTS	-2.3551	0.0185	118.0	347.0	0.0008	0
TM-TMHS	-1.4501	0.147	162.0	303.0	0.0008	0
TM-FMTH	-0.8947	0.3709	189.0	276.0	0.0008	0
HMTS-TMHS	-0.3599	0.7189	250.0	215.0	0.0008	0
HMTS-FMTH	-1.738	0.0822	317.0	148.0	0.0008	0
TMHS-FMTH	-1.3472	0.1779	298.0	167.0	0.0008	0

Table A.66: BoG - Easy

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-3.9183	0.0001	42.0	423.0	0.0008	-0.7154
SGP-Immigrants	-4.4119	0.0	18.0	447.0	0.0008	-0.8055
SGP-FM	-2.3139	0.0207	120.0	345.0	0.0008	0
SGP-T	-3.9388	0.0001	41.0	424.0	0.0008	-0.7191
SGP-HM	-3.98	0.0001	39.0	426.0	0.0008	-0.7266
SGP-IM	-0.833	0.4048	273.0	192.0	0.0008	0
SGP-RIM	-0.5451	0.5857	206.0	259.0	0.0008	0
SGP-TM	-4.2268	0.0	27.0	438.0	0.0008	-0.7717

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-HMTS	-3.9594	0.0001	40.0	425.0	0.0008	-0.7229
SGP-TMHS	-3.9388	0.0001	41.0	424.0	0.0008	-0.7191
SGP-FMTH	-4.2885	0.0	24.0	441.0	0.0008	-0.783
TH-Immigrants	-2.2317	0.0256	124.0	341.0	0.0008	0
TH-FM	-1.9437	0.0519	327.0	138.0	0.0008	0
TH-T	-0.6273	0.5304	263.0	202.0	0.0008	0
TH-HM	-0.8536	0.3933	191.0	274.0	0.0008	0
TH-IM	-3.6714	0.0002	411.0	54.0	0.0008	-0.6703
TH-RIM	-2.9927	0.0028	378.0	87.0	0.0008	0
TH-TM	-1.6146	0.1064	154.0	311.0	0.0008	0
TH-HMTS	-0.3394	0.7343	216.0	249.0	0.0008	0
TH-TMHS	-0.7096	0.4779	198.0	267.0	0.0008	0
TH-FMTH	-0.8947	0.3709	189.0	276.0	0.0008	0
Immigrants-FM	-3.5275	0.0004	404.0	61.0	0.0008	-0.644
Immigrants-T	-2.4168	0.0157	350.0	115.0	0.0008	0
Immigrants-HM	-1.6558	0.0978	313.0	152.0	0.0008	0
Immigrants-IM	-4.6176	0.0	457.0	8.0	0.0008	-0.8431
Immigrants-RIM	-3.9594	0.0001	425.0	40.0	0.0008	-0.7229
Immigrants-TM	-0.2365	0.813	244.0	221.0	0.0008	0
Immigrants-HMTS	-2.0054	0.0449	330.0	135.0	0.0008	0
Immigrants-TMHS	-1.1415	0.2536	288.0	177.0	0.0008	0
Immigrants-FMTH	-0.8124	0.4165	272.0	193.0	0.0008	0
FM-T	-1.9026	0.0571	140.0	325.0	0.0008	0
FM-HM	-2.7047	0.0068	101.0	364.0	0.0008	0
FM-IM	-2.3139	0.0207	345.0	120.0	0.0008	0
FM-RIM	-1.265	0.2059	294.0	171.0	0.0008	0
FM-TM	-2.8281	0.0047	95.0	370.0	0.0008	0
FM-HMTS	-2.643	0.0082	104.0	361.0	0.0008	0
FM-TMHS	-2.3345	0.0196	119.0	346.0	0.0008	0
FM-FMTH	-2.7047	0.0068	101.0	364.0	0.0008	0
T-HM	-1.2238	0.221	173.0	292.0	0.0008	0
T-IM	-3.9594	0.0001	425.0	40.0	0.0008	-0.7229
T-RIM	-2.4991	0.0125	354.0	111.0	0.0008	0
T-TM	-2.0671	0.0387	132.0	333.0	0.0008	0
T-HMTS	-0.7302	0.4653	197.0	268.0	0.0008	0
T-TMHS	-2.0877	0.0368	131.0	334.0	0.0008	0
T-FMTH	-1.9848	0.0472	136.0	329.0	0.0008	0
HM-IM	-4.0828	0.0	431.0	34.0	0.0008	-0.7454
HM-RIM	-2.8281	0.0047	370.0	95.0	0.0008	0
HM-TM	-0.977	0.3286	185.0	280.0	0.0008	0
HM-HMTS	-0.2777	0.7813	246.0	219.0	0.0008	0
HM-TMHS	-0.216	0.829	222.0	243.0	0.0008	0
HM-FMTH	-0.216	0.829	222.0	243.0	0.0008	0
IM-RIM	-1.4706	0.1414	161.0	304.0	0.0008	0
IM-TM	-4.597	0.0	9.0	456.0	0.0008	-0.8393
IM-HMTS	-4.2474	0.0	26.0	439.0	0.0008	-0.7755
IM-TMHS	-4.6587	0.0	6.0	459.0	0.0008	-0.8506
IM-FMTH	-4.1445	0.0	31.0	434.0	0.0008	-0.7567
RIM-TM	-3.4246	0.0006	66.0	399.0	0.0008	-0.6252

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
RIM-HMTS	-3.0955	0.002	82.0	383.0	0.0008	0
RIM-TMHS	-3.5892	0.0003	58.0	407.0	0.0008	-0.6553
RIM-FMTH	-3.4041	0.0007	67.0	398.0	0.0008	-0.6215
TM-HMTS	-1.6146	0.1064	311.0	154.0	0.0008	0
TM-TMHS	-0.7919	0.4284	271.0	194.0	0.0008	0
TM-FMTH	-0.8124	0.4165	272.0	193.0	0.0008	0
HMTS-TMHS	-0.7302	0.4653	197.0	268.0	0.0008	0
HMTS-FMTH	-0.833	0.4048	192.0	273.0	0.0008	0
TMHS-FMTH	-0.4011	0.6884	213.0	252.0	0.0008	0

Table A.67: BoG - Medium

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-4.4119	0.0	18.0	447.0	0.0008	-0.8055
SGP-Immigrants	-4.3708	0.0	20.0	445.0	0.0008	-0.798
SGP-FM	-4.5148	0.0	13.0	452.0	0.0008	-0.8243
SGP-T	-3.9388	0.0001	41.0	424.0	0.0008	-0.7191
SGP-HM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
SGP-IM	-0.6479	0.517	264.0	201.0	0.0008	0
SGP-RIM	-3.4041	0.0007	398.0	67.0	0.0008	-0.6215
SGP-TM	-4.5353	0.0	12.0	453.0	0.0008	-0.828
SGP-HMTS	-4.741	0.0	2.0	463.0	0.0008	-0.8656
SGP-TMHS	-4.6382	0.0	7.0	458.0	0.0008	-0.8468
SGP-FMTH	-4.5353	0.0	12.0	453.0	0.0008	-0.828
TH-Immigrants	-2.6636	0.0077	362.0	103.0	0.0008	0
TH-FM	-2.1905	0.0285	339.0	126.0	0.0008	0
TH-T	-2.5402	0.0111	356.0	109.0	0.0008	0
TH-HM	-3.6714	0.0002	54.0	411.0	0.0008	-0.6703
TH-IM	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
TH-RIM	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
TH-TM	-1.3061	0.1915	169.0	296.0	0.0008	0
TH-HMTS	-2.0054	0.0449	135.0	330.0	0.0008	0
TH-TMHS	-0.9976	0.3185	184.0	281.0	0.0008	0
TH-FMTH	-0.1131	0.9099	238.0	227.0	0.0008	0
Immigrants-FM	-0.0103	0.9918	233.0	232.0	0.0008	0
Immigrants-T	-0.6479	0.517	264.0	201.0	0.0008	0
Immigrants-HM	-4.4325	0.0	17.0	448.0	0.0008	-0.8093
Immigrants-IM	-4.7204	0.0	462.0	3.0	0.0008	-0.8618
Immigrants-RIM	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
Immigrants-TM	-2.7253	0.0064	100.0	365.0	0.0008	0
Immigrants-HMTS	-3.8566	0.0001	45.0	420.0	0.0008	-0.7041
Immigrants-TMHS	-2.9927	0.0028	87.0	378.0	0.0008	0
Immigrants-FMTH	-2.7253	0.0064	100.0	365.0	0.0008	0
FM-T	-0.7096	0.4779	267.0	198.0	0.0008	0
FM-HM	-4.4119	0.0	18.0	447.0	0.0008	-0.8055
FM-IM	-4.2679	0.0	440.0	25.0	0.0008	-0.7792
FM-RIM	-4.7204	0.0	462.0	3.0	0.0008	-0.8618
FM-TM	-2.8899	0.0039	92.0	373.0	0.0008	0
FM-HMTS	-3.3012	0.001	72.0	393.0	0.0008	0

Alg1-Alg2	**Z**	**P-value**	**P rank**	**N rank**	**Alpha**	**R**
FM-TMHS	-2.3551	0.0185	118.0	347.0	0.0008	0
FM-FMTH	-1.9848	0.0472	136.0	329.0	0.0008	0
T-HM	-4.4119	0.0	18.0	447.0	0.0008	-0.8055
T-IM	-4.5353	0.0	453.0	12.0	0.0008	-0.828
T-RIM	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
T-TM	-3.3835	0.0007	68.0	397.0	0.0008	-0.6177
T-HMTS	-4.2268	0.0	27.0	438.0	0.0008	-0.7717
T-TMHS	-3.6509	0.0003	55.0	410.0	0.0008	-0.6666
T-FMTH	-2.7459	0.006	99.0	366.0	0.0008	0
HM-IM	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
HM-RIM	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
HM-TM	-2.9516	0.0032	376.0	89.0	0.0008	0
HM-HMTS	-2.4579	0.014	352.0	113.0	0.0008	0
HM-TMHS	-2.5813	0.0098	358.0	107.0	0.0008	0
HM-FMTH	-3.6714	0.0002	411.0	54.0	0.0008	-0.6703
IM-RIM	-3.3218	0.0009	394.0	71.0	0.0008	0
IM-TM	-4.7616	0.0	1.0	464.0	0.0008	-0.8693
IM-HMTS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
IM-TMHS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
IM-FMTH	-4.7204	0.0	3.0	462.0	0.0008	-0.8618
RIM-TM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
RIM-HMTS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
RIM-TMHS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
RIM-FMTH	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
TM-HMTS	-0.3394	0.7343	216.0	249.0	0.0008	0
TM-TMHS	-0.216	0.829	243.0	222.0	0.0008	0
TM-FMTH	-1.0593	0.2895	284.0	181.0	0.0008	0
HMTS-TMHS	-1.3472	0.1779	298.0	167.0	0.0008	0
HMTS-FMTH	-2.0465	0.0407	332.0	133.0	0.0008	0
TMHS-FMTH	-1.0593	0.2895	284.0	181.0	0.0008	0

Table A.68: BoG - Hard

Santa Fe Ant Trail

Descriptive Statistics

Offline Performance

Algorithm	Min	Max	Mean	Std	Skewness	Kurtosis
SGP	0.592	0.9792	0.7227	0.1001	1.2169	0.6972
FM	0.5701	0.9639	0.7568	0.1065	0.354	-0.9847
TH	0.5818	0.9792	0.7429	0.1186	0.3741	-1.192
I	0.5692	0.9772	0.7707	0.138	0.2909	-1.5187
T	0.6133	0.9722	0.8039	0.1274	0.0271	-1.6494
TM	0.6516	0.9659	0.7747	0.1012	0.7073	-0.9929
HM	0.5849	0.9671	0.7545	0.1024	0.6555	-0.7332
IM	0.5468	0.9743	0.8172	0.1256	-0.2235	-1.3065
RIM	0.7079	0.9792	0.9082	0.0703	-1.4475	1.6547
TMHS	0.6563	0.9722	0.8098	0.1131	0.2508	-1.5061
HMTS	0.5936	0.9722	0.835	0.1125	-0.2766	-1.3507
FMTH	0.6121	0.9722	0.8125	0.1169	-0.148	-1.3377

Table A.69: Easy Scenario

Algorithm	Min	Max	Mean	Std	Skewness	Kurtosis
SGP	0.5772	0.7317	0.6176	0.0366	1.6839	2.5284
FM	0.575	0.8631	0.7106	0.0771	0.5605	-0.7242
TH	0.5536	0.6707	0.6014	0.028	0.5714	-0.0547
I	0.5525	0.7796	0.6072	0.0397	2.5833	9.2385
T	0.5357	0.7056	0.598	0.032	1.0156	2.6359
TM	0.6492	0.9173	0.736	0.0724	1.0484	0.3576
HM	0.611	0.8706	0.7009	0.0674	1.2685	0.7914
IM	0.5952	0.8885	0.7141	0.0838	0.6798	-0.7799
RIM	0.596	0.894	0.7874	0.0579	-1.3946	2.5267
TMHS	0.6097	0.8668	0.7534	0.0721	-0.0665	-1.157
HMTS	0.6327	0.8747	0.7438	0.0646	0.3256	-0.8433
FMTH	0.6158	0.8645	0.7308	0.0688	0.5391	-0.8255

Table A.70: Medium Scenario

Algorithm	Min	Max	Mean	Std	Skewness	Kurtosis
SGP	0.5815	0.6722	0.6214	0.0228	0.3737	-0.3495
FM	0.6362	0.7542	0.7028	0.0319	-0.3511	-0.8222
TH	0.5529	0.6428	0.6055	0.0224	-0.1223	-0.6952
I	0.5716	0.6497	0.6111	0.0187	-0.3986	-0.4377
T	0.5492	0.6308	0.5925	0.0209	-0.1917	-0.4686
TM	0.653	0.7977	0.7153	0.0371	0.3815	-0.4669
HM	0.629	0.8215	0.7195	0.0381	-0.2242	1.1969
IM	0.6177	0.7472	0.6954	0.0317	-0.6472	-0.1161
RIM	0.6435	0.7733	0.7136	0.0346	-0.2635	-0.7773

Algorithm	Min	Max	Mean	Std	Skewness	Kurtosis
TMHS	0.6622	0.8287	0.7526	0.0382	-0.3245	0.2261
HMTS	0.6954	0.8039	0.756	0.0286	-0.4438	-0.465
FMTH	0.7005	0.8187	0.7496	0.0289	0.4119	-0.725

Table A.71: Hard Scenario

Best of Generation

Algorithm	Min	Max	Mean	Std	Skewness	Kurtosis
SGP	0.6558	0.996	0.7566	0.0959	1.2919	0.6836
FM	0.5887	0.9806	0.7771	0.1095	0.3373	-1.0701
TH	0.6183	0.996	0.7839	0.1116	0.3215	-1.3037
I	0.6228	0.9952	0.8039	0.1303	0.2153	-1.5324
T	0.6627	0.9972	0.8373	0.1222	-0.0094	-1.6546
TM	0.6654	0.9876	0.7949	0.1046	0.6802	-1.0236
HM	0.6015	0.9859	0.7751	0.1059	0.6567	-0.7991
IM	0.5574	0.9988	0.839	0.1274	-0.2824	-1.2418
RIM	0.7293	0.9972	0.9313	0.0677	-1.6297	2.3936
TMHS	0.6744	0.9973	0.8311	0.1148	0.2145	-1.5292
HMTS	0.6058	0.9954	0.8588	0.1163	-0.2951	-1.3542
FMTH	0.6259	0.9967	0.8353	0.1196	-0.1958	-1.3212

Table A.72: Easy Scenario

Algorithm	Min	Max	Mean	Std	Skewness	Kurtosis
SGP	0.6443	0.8136	0.6813	0.0425	1.7841	2.1934
FM	0.5926	0.8871	0.7298	0.0781	0.5713	-0.6996
TH	0.6242	0.7648	0.673	0.031	0.8589	0.6683
I	0.6323	0.871	0.6763	0.0427	3.0906	11.5788
T	0.6085	0.8056	0.6751	0.0349	1.5871	4.451
TM	0.667	0.9423	0.7551	0.0756	0.9558	0.1089
HM	0.6211	0.8916	0.7221	0.0712	1.1037	0.3717
IM	0.6012	0.9127	0.7332	0.0865	0.6318	-0.7139
RIM	0.6588	0.9138	0.8112	0.0536	-1.1493	1.3578
TMHS	0.619	0.8876	0.7796	0.0755	-0.1921	-1.0466
HMTS	0.6586	0.8925	0.769	0.066	0.2312	-1.0477
FMTH	0.6394	0.881	0.755	0.068	0.4488	-0.9568

Table A.73: Medium Scenario

Algorithm	Min	Max	Mean	Std	Skewness	Kurtosis
SGP	0.6394	0.7099	0.6732	0.019	0.1921	-0.8007
FM	0.6432	0.7693	0.7118	0.0326	-0.2583	-0.7897
TH	0.6372	0.7047	0.6734	0.0188	-0.0808	-0.8617
I	0.6307	0.6979	0.6719	0.0188	-0.7101	-0.5116
T	0.6147	0.6914	0.6613	0.0184	-0.4911	-0.1699
TM	0.6589	0.8077	0.7243	0.0379	0.339	-0.5083
HM	0.6373	0.832	0.7295	0.0386	-0.2226	1.2355

Algorithm	Min	Max	Mean	Std	Skewness	Kurtosis
IM	0.6253	0.7571	0.7045	0.0319	-0.6719	-0.059
RIM	0.6529	0.7869	0.724	0.0346	-0.3065	-0.6737
TMHS	0.6743	0.8413	0.7635	0.0387	-0.2883	0.1515
HMTS	0.7049	0.8167	0.7675	0.0288	-0.4305	-0.4998
FMTH	0.7116	0.8281	0.7611	0.0296	0.3484	-0.8813

Table A.74: Hard Scenario

Normality Tests

Offline Performance

Algorithm	D	P-value
SGP	0.4577	0.0
FM	0.4645	0.0
TH	0.4926	0.0
I	0.5494	0.0
T	0.4641	0.0
TM	0.5882	0.0
HM	0.4323	0.0
IM	0.4641	0.0
RIM	0.4936	0.0
TMHS	0.459	0.0
HMTS	0.4987	0.0
FMTH	0.4558	0.0

Table A.75: Kolmogorov-Smirnov - Easy Scenario

Algorithm	D	P-value
SGP	0.426	0.0
FM	0.4375	0.0
TH	0.4491	0.0
I	0.409	0.0
T	0.2811	0.0137
TM	0.45	0.0
HM	0.458	0.0
IM	0.5132	0.0
RIM	0.4299	0.0
TMHS	0.3995	0.0001
HMTS	0.4713	0.0
FMTH	0.4857	0.0

Table A.76: Kolmogorov-Smirnov - Medium Scenario

Algorithm	D	P-value
SGP	0.4517	0.0
FM	0.404	0.0001

Algorithm	D	P-value
TH	0.4164	0.0
I	0.4141	0.0
T	0.3896	0.0001
TM	0.3778	0.0002
HM	0.3587	0.0006
IM	0.3977	0.0001
RIM	0.3926	0.0001
TMHS	0.331	0.002
HMTS	0.4286	0.0
FMTH	0.4747	0.0

Table A.77: Kolmogorov-Smirnov - Hard Scenario

Best of Generation

Algorithm	D	P-value
SGP	0.4665	0.0
FM	0.4638	0.0
TH	0.5242	0.0
I	0.496	0.0
T	0.4664	0.0
TM	0.5821	0.0
HM	0.4319	0.0
IM	0.4631	0.0
RIM	0.4902	0.0
TMHS	0.4587	0.0
HMTS	0.499	0.0
FMTH	0.4639	0.0

Table A.78: Kolmogorov-Smirnov - Easy Scenario

Algorithm	D	P-value
SGP	0.5055	0.0
FM	0.4302	0.0
TH	0.4054	0.0001
I	0.4138	0.0
T	0.4101	0.0
TM	0.4394	0.0
HM	0.4517	0.0
IM	0.4616	0.0
RIM	0.4463	0.0
TMHS	0.3983	0.0001
HMTS	0.4634	0.0
FMTH	0.4331	0.0

Table A.79: Kolmogorov-Smirnov - Medium Scenario

Algorithm	D	P-value
SGP	0.3863	0.0002
FM	0.3638	0.0005
TH	0.4314	0.0
I	0.4585	0.0
T	0.397	0.0001
TM	0.3919	0.0001
HM	0.3102	0.0046
IM	0.4128	0.0
RIM	0.3667	0.0004
TMHS	0.3521	0.0008
HMTS	0.403	0.0001
FMTH	0.4665	0.0

Table A.80: Kolmogorov-Smirnov - Hard Scenario

Simple Techniques

Offline Performance

Algorithms	Scenario	Chi-prob	P-value
SGP, tH, I, fM	Easy	1.3344	0.721
SGP, tH, I, fM	Medium	39.88	0.0
SGP, tH, I, fM	Hard	56.52	0.0

Table A.81: Friedman's Anova

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-0.7892	0.43	181.0	254.0	0.0167	0
SGP-I	-1.3061	0.1915	169.0	296.0	0.0167	0
SGP-FM	-1.1827	0.2369	175.0	290.0	0.0167	0

Table A.82: Wilcoxon - Easy

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-1.3061	0.1915	296.0	169.0	0.0167	0
SGP-I	-1.5529	0.1204	308.0	157.0	0.0167	0
SGP-FM	-4.453	0.0	16.0	449.0	0.0167	-0.813

Table A.83: Wilcoxon - Medium

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-2.4373	0.0148	351.0	114.0	0.0167	-0.445
SGP-I	-1.9437	0.0519	327.0	138.0	0.0167	0
SGP-FM	-4.7821	0.0	0.0	465.0	0.0167	-0.8731

Table A.84: Wilcoxon - Hard

Best Of Generation

Algorithms	Scenario	Chi-prob	P-value
SGP, tH, I, fM	Easy	2.2977	0.513
SGP, tH, I, fM	Medium	14.36	0.0025
SGP, tH, I, fM	Hard	24.28	0.0

Table A.85: Friedman's Anova

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-1.1136	0.2655	166.0	269.0	0.0167	0
SGP-I	-1.2444	0.2134	172.0	293.0	0.0167	0
SGP-FM	-0.7507	0.4528	196.0	269.0	0.0167	0

Table A.86: Wilcoxon - Easy

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-0.0309	0.9754	234.0	231.0	0.0167	0
SGP-I	-0.4011	0.6884	252.0	213.0	0.0167	0
SGP-FM	-2.9927	0.0028	87.0	378.0	0.0167	-0.5464

Table A.87: Wilcoxon - Medium

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-0.2571	0.7971	220.0	245.0	0.0167	0
SGP-I	-0.2777	0.7813	246.0	219.0	0.0167	0
SGP-FM	-4.1857	0.0	29.0	436.0	0.0167	-0.7642

Table A.88: Wilcoxon - Hard

Hybrid Techniques

Offline Performance

Algorithms	Scenario	Chi-prob	P-value
T, SGP, TH, I, FM	Easy	6.8047	0.1466
HM, TH, FM	Easy	3.2667	0.1953
IM, I, FM	Easy	2.0667	0.3558
RIM, IM, I, FM	Easy	23.99	0.0
TM, FM, T	Easy	2.4	0.3012
HMTS, SGP, HM, T	Easy	13.64	0.0034
TMHS, SGP, TM, TH	Easy	12.3512	0.0063
FMTH, FM, TH, T	Easy	7.56	0.056
T, SGP, TH, I, FM	Medium	47.8133	0.0
HM, TH, FM	Medium	36.4667	0.0
IM, I, FM	Medium	28.8	0.0
RIM, IM, I, FM	Medium	45.88	0.0
TM, FM, T	Medium	43.4	0.0

Algorithms	Scenario	Chi-prob	P-value
HMTS, SGP, HM, T	Medium	60.28	0.0
TMHS, SGP, TM, TH	Medium	65.08	0.0
FMTH, FM, TH, T	Medium	63.72	0.0
T, SGP, TH, I, FM	Hard	69.2	0.0
HM, TH, FM	Hard	42.8667	0.0
IM, I, FM	Hard	39.8	0.0
RIM, IM, I, FM	Hard	53.08	0.0
TM, FM, T	Hard	45.0667	0.0
HMTS, SGP, HM, T	Hard	77.64	0.0
TMHS, SGP, TM, TH	Hard	73.32	0.0
FMTH, FM, TH, T	Hard	77.0	0.0

Table A.89: Friedman's Anova

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
T-SGP	-2.0465	0.0407	332.0	133.0	0.0125	0
T-TH	-1.7997	0.0719	320.0	145.0	0.0125	0
T-I	-1.265	0.2059	294.0	171.0	0.0125	0
T-FM	-1.265	0.2059	294.0	171.0	0.0125	0
HM-TH	-0.6068	0.544	262.0	203.0	0.025	0
HM-FM	-0.216	0.829	243.0	222.0	0.025	0
IM-I	-1.4912	0.1359	305.0	160.0	0.025	0
IM-FM	-1.7997	0.0719	320.0	145.0	0.025	0
RIM-IM	-2.7664	0.0057	367.0	98.0	0.0167	-0.5051
RIM-I	-3.73	0.0002	390.0	45.0	0.0167	-0.6926
RIM-FM	-4.2062	0.0	437.0	28.0	0.0167	-0.7679
TM-FM	-0.5656	0.5716	260.0	205.0	0.025	0
TM-T	-0.8124	0.4165	193.0	272.0	0.025	0
HMTS-SGP	-3.7743	0.0002	416.0	49.0	0.0167	-0.6891
HMTS-HM	-2.4373	0.0148	351.0	114.0	0.0167	-0.445
HMTS-T	-1.0387	0.2989	283.0	182.0	0.0167	0
TMHS-SGP	-3.4246	0.0006	399.0	66.0	0.0167	-0.6252
TMHS-TM	-1.3267	0.1846	297.0	168.0	0.0167	0
TMHS-TH	-2.2317	0.0256	341.0	124.0	0.0167	0
FMTH-FM	-1.738	0.0822	317.0	148.0	0.0167	0
FMTH-TH	-2.1905	0.0285	339.0	126.0	0.0167	0
FMTH-T	-0.5245	0.5999	258.0	207.0	0.0167	0

Table A.90: Wilcoxon - Easy

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
T-SGP	-2.026	0.0428	134.0	331.0	0.0125	0
T-TH	-0.5451	0.5857	206.0	259.0	0.0125	0
T-I	-0.8124	0.4165	193.0	272.0	0.0125	0
T-FM	-4.741	0.0	2.0	463.0	0.0125	-0.8656
HM-TH	-4.7616	0.0	464.0	1.0	0.025	-0.8693
HM-FM	-0.3599	0.7189	215.0	250.0	0.025	0
IM-I	-4.3913	0.0	446.0	19.0	0.025	-0.8017
IM-FM	-0.2982	0.7655	247.0	218.0	0.025	0

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
RIM-IM	-3.4452	0.0006	400.0	65.0	0.0167	-0.629
RIM-I	-4.6999	0.0	461.0	4.0	0.0167	-0.8581
RIM-FM	-3.4658	0.0005	401.0	64.0	0.0167	-0.6328
TM-FM	-1.2032	0.2289	291.0	174.0	0.025	0
TM-T	-4.7821	0.0	465.0	0.0	0.025	-0.8731
HMTS-SGP	-4.6587	0.0	459.0	6.0	0.0167	-0.8506
HMTS-HM	-2.7253	0.0064	365.0	100.0	0.0167	-0.4976
HMTS-T	-4.7821	0.0	465.0	0.0	0.0167	-0.8731
TMHS-SGP	-4.7821	0.0	465.0	0.0	0.0167	-0.8731
TMHS-TM	-1.1004	0.2712	286.0	179.0	0.0167	0
TMHS-TH	-4.7616	0.0	464.0	1.0	0.0167	-0.8693
FMTH-FM	-0.8947	0.3709	276.0	189.0	0.0167	0
FMTH-TH	-4.7616	0.0	464.0	1.0	0.0167	-0.8693
FMTH-T	-4.7821	0.0	465.0	0.0	0.0167	-0.8731

Table A.91: Wilcoxon - Medium

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
T-SGP	-4.0005	0.0001	38.0	427.0	0.0125	-0.7304
T-TH	-2.2522	0.0243	123.0	342.0	0.0125	0
T-I	-2.9721	0.003	88.0	377.0	0.0125	-0.5426
T-FM	-4.7821	0.0	0.0	465.0	0.0125	-0.8731
HM-TH	-4.741	0.0	463.0	2.0	0.025	-0.8656
HM-FM	-1.594	0.1109	310.0	155.0	0.025	0
IM-I	-4.6793	0.0	460.0	5.0	0.025	-0.8543
IM-FM	-1.0798	0.2802	180.0	285.0	0.025	0
RIM-IM	-2.1083	0.035	335.0	130.0	0.0167	0
RIM-I	-4.7821	0.0	465.0	0.0	0.0167	-0.8731
RIM-FM	-1.3678	0.1714	299.0	166.0	0.0167	0
TM-FM	-1.0593	0.2895	284.0	181.0	0.025	0
TM-T	-4.7821	0.0	465.0	0.0	0.025	-0.8731
HMTS-SGP	-4.7821	0.0	465.0	0.0	0.0167	-0.8731
HMTS-HM	-3.548	0.0004	405.0	60.0	0.0167	-0.6478
HMTS-T	-4.7821	0.0	465.0	0.0	0.0167	-0.8731
TMHS-SGP	-4.7616	0.0	464.0	1.0	0.0167	-0.8693
TMHS-TM	-3.3835	0.0007	397.0	68.0	0.0167	-0.6177
TMHS-TH	-4.7821	0.0	465.0	0.0	0.0167	-0.8731
FMTH-FM	-4.0211	0.0001	428.0	37.0	0.0167	-0.7342
FMTH-TH	-4.7821	0.0	465.0	0.0	0.0167	-0.8731
FMTH-T	-4.7821	0.0	465.0	0.0	0.0167	-0.8731

Table A.92: Wilcoxon - Hard

Best Of Generation

Algorithms	Scenario	Chi-prob	P-value
T, SGP, TH, I, FM	Easy	9.1686	0.057
HM, TH, FM	Easy	0.0667	0.9672

Algorithms	Scenario	Chi-prob	P-value
IM, I, FM	Easy	4.2	0.1225
RIM, IM, I, FM	Easy	25.08	0.0
TM, FM, T	Easy	2.8667	0.2385
HMTS, SGP, HM, T	Easy	12.04	0.0072
TMHS, SGP, TM, TH	Easy	8.0368	0.0453
FMTH, FM, TH, T	Easy	7.32	0.0624
T, SGP, TH, I, FM	Medium	15.4667	0.0038
HM, TH, FM	Medium	11.4	0.0033
IM, I, FM	Medium	8.6	0.0136
RIM, IM, I, FM	Medium	31.84	0.0
TM, FM, T	Medium	17.8667	0.0001
HMTS, SGP, HM, T	Medium	35.0	0.0
TMHS, SGP, TM, TH	Medium	39.12	0.0
FMTH, FM, TH, T	Medium	27.4	0.0
T, SGP, TH, I, FM	Hard	30.2667	0.0
HM, TH, FM	Hard	27.8	0.0
IM, I, FM	Hard	14.8667	0.0006
RIM, IM, I, FM	Hard	32.12	0.0
TM, FM, T	Hard	29.4	0.0
HMTS, SGP, HM, T	Hard	68.44	0.0
TMHS, SGP, TM, TH	Hard	59.8	0.0
FMTH, FM, TH, T	Hard	60.92	0.0

Table A.93: Friedman's Anova

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
T-SGP	-2.2934	0.0218	344.0	121.0	0.0125	0
T-TH	-1.6352	0.102	312.0	153.0	0.0125	0
T-I	-1.5118	0.1306	306.0	159.0	0.0125	0
T-FM	-1.6352	0.102	312.0	153.0	0.0125	0
HM-TH	-0.2571	0.7971	220.0	245.0	0.025	0
HM-FM	-0.2777	0.7813	246.0	219.0	0.025	0
IM-I	-0.977	0.3286	280.0	185.0	0.025	0
IM-FM	-1.8614	0.0627	323.0	142.0	0.025	0
RIM-IM	-2.7664	0.0057	367.0	98.0	0.0167	-0.5051
RIM-I	-3.7126	0.0002	413.0	52.0	0.0167	-0.6778
RIM-FM	-4.2062	0.0	437.0	28.0	0.0167	-0.7679
TM-FM	-0.6273	0.5304	263.0	202.0	0.025	0
TM-T	-1.1415	0.2536	177.0	288.0	0.025	0
HMTS-SGP	-3.6097	0.0003	408.0	57.0	0.0167	-0.659
HMTS-HM	-2.4579	0.014	352.0	113.0	0.0167	-0.4488
HMTS-T	-0.7302	0.4653	268.0	197.0	0.0167	0
TMHS-SGP	-3.0955	0.002	383.0	82.0	0.0167	-0.5652
TMHS-TM	-1.4501	0.147	303.0	162.0	0.0167	0
TMHS-TH	-1.7175	0.0859	316.0	149.0	0.0167	0
FMTH-FM	-1.7586	0.0786	318.0	147.0	0.0167	0
FMTH-TH	-1.8614	0.0627	323.0	142.0	0.0167	0
FMTH-T	-0.1131	0.9099	238.0	227.0	0.0167	0

Table A.94: Wilcoxon - Easy

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
T-SGP	-0.0103	0.9918	232.0	233.0	0.0125	0
T-TH	-0.072	0.9426	229.0	236.0	0.0125	0
T-I	-0.5039	0.6143	257.0	208.0	0.0125	0
T-FM	-2.9927	0.0028	87.0	378.0	0.0125	-0.5464
HM-TH	-3.5069	0.0005	403.0	62.0	0.025	-0.6403
HM-FM	-0.2571	0.7971	220.0	245.0	0.025	0
IM-I	-2.5813	0.0098	358.0	107.0	0.025	-0.4713
IM-FM	-0.2777	0.7813	246.0	219.0	0.025	0
RIM-IM	-3.4658	0.0005	401.0	64.0	0.0167	-0.6328
RIM-I	-4.3296	0.0	443.0	22.0	0.0167	-0.7905
RIM-FM	-3.6097	0.0003	408.0	57.0	0.0167	-0.659
TM-FM	-1.2444	0.2134	293.0	172.0	0.025	0
TM-T	-4.1857	0.0	436.0	29.0	0.025	-0.7642
HMTS-SGP	-3.9594	0.0001	425.0	40.0	0.0167	-0.7229
HMTS-HM	-2.6636	0.0077	362.0	103.0	0.0167	-0.4863
HMTS-T	-4.5559	0.0	454.0	11.0	0.0167	-0.8318
TMHS-SGP	-4.2062	0.0	437.0	28.0	0.0167	-0.7679
TMHS-TM	-1.4501	0.147	303.0	162.0	0.0167	0
TMHS-TH	-4.3502	0.0	444.0	21.0	0.0167	-0.7942
FMTH-FM	-1.121	0.2623	287.0	178.0	0.0167	0
FMTH-TH	-4.2679	0.0	440.0	25.0	0.0167	-0.7792
FMTH-T	-4.2679	0.0	440.0	25.0	0.0167	-0.7792

Table A.95: Wilcoxon - Medium

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
T-SGP	-2.3139	0.0207	120.0	345.0	0.0125	0
T-TH	-2.3756	0.0175	117.0	348.0	0.0125	0
T-I	-1.9026	0.0571	140.0	325.0	0.0125	0
T-FM	-4.3913	0.0	19.0	446.0	0.0125	-0.8017
HM-TH	-4.1651	0.0	435.0	30.0	0.025	-0.7604
HM-FM	-1.738	0.0822	317.0	148.0	0.025	0
IM-I	-3.3424	0.0008	395.0	70.0	0.025	-0.6102
IM-FM	-0.9976	0.3185	184.0	281.0	0.025	0
RIM-IM	-2.2111	0.027	340.0	125.0	0.0167	0
RIM-I	-4.7204	0.0	462.0	3.0	0.0167	-0.8618
RIM-FM	-1.5529	0.1204	308.0	157.0	0.0167	0
TM-FM	-1.1415	0.2536	288.0	177.0	0.025	0
TM-T	-4.7616	0.0	464.0	1.0	0.025	-0.8693
HMTS-SGP	-4.7821	0.0	465.0	0.0	0.0167	-0.8731
HMTS-HM	-3.6714	0.0002	411.0	54.0	0.0167	-0.6703
HMTS-T	-4.7821	0.0	465.0	0.0	0.0167	-0.8731
TMHS-SGP	-4.7204	0.0	462.0	3.0	0.0167	-0.8618
TMHS-TM	-3.4246	0.0006	399.0	66.0	0.0167	-0.6252
TMHS-TH	-4.7616	0.0	464.0	1.0	0.0167	-0.8693
FMTH-FM	-4.0828	0.0	431.0	34.0	0.0167	-0.7454
FMTH-TH	-4.7821	0.0	465.0	0.0	0.0167	-0.8731
FMTH-T	-4.7821	0.0	465.0	0.0	0.0167	-0.8731

Table A.96: Wilcoxon - Hard

Search for the Best Algorithm

Offline Performance

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-0.7892	0.43	181.0	254.0	0.0008	0
SGP-I	-1.3061	0.1915	169.0	296.0	0.0008	0
SGP-FM	-1.1827	0.2369	175.0	290.0	0.0008	0
SGP-T	-2.0465	0.0407	133.0	332.0	0.0008	0
SGP-HM	-1.4295	0.1529	163.0	302.0	0.0008	0
SGP-IM	-3.1778	0.0015	78.0	387.0	0.0008	0
SGP-RIM	-4.3787	0.0	15.0	420.0	0.0008	-0.8131
SGP-TM	-2.0054	0.0449	135.0	330.0	0.0008	0
SGP-HMTS	-3.7743	0.0002	49.0	416.0	0.0008	-0.6891
SGP-TMHS	-3.4246	0.0006	66.0	399.0	0.0008	-0.6252
SGP-FMTH	-3.0133	0.0026	86.0	379.0	0.0008	0
TH-I	-0.6273	0.5304	202.0	263.0	0.0008	0
TH-FM	-0.6068	0.544	203.0	262.0	0.0008	0
TH-T	-1.7997	0.0719	145.0	320.0	0.0008	0
TH-HM	-0.6068	0.544	203.0	262.0	0.0008	0
TH-IM	-2.3551	0.0185	118.0	347.0	0.0008	0
TH-RIM	-4.1841	0.0	24.0	411.0	0.0008	-0.777
TH-TM	-0.9976	0.3185	184.0	281.0	0.0008	0
TH-HMTS	-3.0338	0.0024	85.0	380.0	0.0008	0
TH-TMHS	-2.2317	0.0256	124.0	341.0	0.0008	0
TH-FMTH	-2.1905	0.0285	126.0	339.0	0.0008	0
I-FM	-0.4217	0.6733	253.0	212.0	0.0008	0
I-T	-1.265	0.2059	171.0	294.0	0.0008	0
I-HM	-0.3394	0.7343	249.0	216.0	0.0008	0
I-IM	-1.4912	0.1359	160.0	305.0	0.0008	0
I-RIM	-3.73	0.0002	45.0	390.0	0.0008	-0.6926
I-TM	-0.2982	0.7655	218.0	247.0	0.0008	0
I-HMTS	-1.9643	0.0495	137.0	328.0	0.0008	0
I-TMHS	-1.1621	0.2452	176.0	289.0	0.0008	0
I-FMTH	-1.3061	0.1915	169.0	296.0	0.0008	0
FM-T	-1.265	0.2059	171.0	294.0	0.0008	0
FM-HM	-0.216	0.829	222.0	243.0	0.0008	0
FM-IM	-1.7997	0.0719	145.0	320.0	0.0008	0
FM-RIM	-4.2062	0.0	28.0	437.0	0.0008	-0.7679
FM-TM	-0.5656	0.5716	205.0	260.0	0.0008	0
FM-HMTS	-2.3962	0.0166	116.0	349.0	0.0008	0
FM-TMHS	-2.0465	0.0407	133.0	332.0	0.0008	0
FM-FMTH	-1.738	0.0822	148.0	317.0	0.0008	0
T-HM	-1.3472	0.1779	298.0	167.0	0.0008	0
T-IM	-0.5039	0.6143	208.0	257.0	0.0008	0
T-RIM	-2.8899	0.0039	92.0	373.0	0.0008	0
T-TM	-0.8124	0.4165	272.0	193.0	0.0008	0
T-HMTS	-1.0387	0.2989	182.0	283.0	0.0008	0
T-TMHS	-0.1954	0.8451	223.0	242.0	0.0008	0
T-FMTH	-0.5245	0.5999	207.0	258.0	0.0008	0

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
HM-IM	-2.1288	0.0333	129.0	336.0	0.0008	0
HM-RIM	-4.4119	0.0	18.0	447.0	0.0008	-0.8055
HM-TM	-0.4011	0.6884	213.0	252.0	0.0008	0
HM-HMTS	-2.4373	0.0148	114.0	351.0	0.0008	0
HM-TMHS	-2.0054	0.0449	135.0	330.0	0.0008	0
HM-FMTH	-1.4706	0.1414	161.0	304.0	0.0008	0
IM-RIM	-2.7664	0.0057	98.0	367.0	0.0008	0
IM-TM	-1.2238	0.221	292.0	173.0	0.0008	0
IM-HMTS	-0.977	0.3286	185.0	280.0	0.0008	0
IM-TMHS	-0.3568	0.7213	234.0	201.0	0.0008	0
IM-FMTH	-0.1131	0.9099	227.0	238.0	0.0008	0
RIM-TM	-3.7743	0.0002	416.0	49.0	0.0008	-0.6891
RIM-HMTS	-2.6019	0.0093	359.0	106.0	0.0008	0
RIM-TMHS	-3.3218	0.0009	394.0	71.0	0.0008	0
RIM-FMTH	-2.8281	0.0047	370.0	95.0	0.0008	0
TM-HMTS	-2.9516	0.0032	89.0	376.0	0.0008	0
TM-TMHS	-1.3267	0.1846	168.0	297.0	0.0008	0
TM-FMTH	-1.2032	0.2289	174.0	291.0	0.0008	0
HMTS-TMHS	-0.7096	0.4779	267.0	198.0	0.0008	0
HMTS-FMTH	-0.7096	0.4779	267.0	198.0	0.0008	0
TMHS-FMTH	-0.1189	0.9053	212.0	223.0	0.0008	0

Table A.97: Wilcoxon Signed Ranks - Easy Scenario

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-1.3061	0.1915	296.0	169.0	0.0008	0
SGP-I	-1.5529	0.1204	308.0	157.0	0.0008	0
SGP-FM	-4.453	0.0	16.0	449.0	0.0008	-0.813
SGP-T	-2.026	0.0428	331.0	134.0	0.0008	0
SGP-HM	-4.3091	0.0	23.0	442.0	0.0008	-0.7867
SGP-IM	-4.5148	0.0	13.0	452.0	0.0008	-0.8243
SGP-RIM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
SGP-TM	-4.597	0.0	9.0	456.0	0.0008	-0.8393
SGP-HMTS	-4.6587	0.0	6.0	459.0	0.0008	-0.8506
SGP-TMHS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
SGP-FMTH	-4.4942	0.0	14.0	451.0	0.0008	-0.8205
TH-I	-0.6685	0.5038	200.0	265.0	0.0008	0
TH-FM	-4.6999	0.0	4.0	461.0	0.0008	-0.8581
TH-T	-0.5451	0.5857	259.0	206.0	0.0008	0
TH-HM	-4.7616	0.0	1.0	464.0	0.0008	-0.8693
TH-IM	-4.6382	0.0	7.0	458.0	0.0008	-0.8468
TH-RIM	-4.7616	0.0	1.0	464.0	0.0008	-0.8693
TH-TM	-4.7616	0.0	1.0	464.0	0.0008	-0.8693
TH-HMTS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
TH-TMHS	-4.7616	0.0	1.0	464.0	0.0008	-0.8693
TH-FMTH	-4.7616	0.0	1.0	464.0	0.0008	-0.8693
I-FM	-4.3091	0.0	23.0	442.0	0.0008	-0.7867
I-T	-0.8124	0.4165	272.0	193.0	0.0008	0
I-HM	-4.2474	0.0	26.0	439.0	0.0008	-0.7755

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
I-IM	-4.3913	0.0	19.0	446.0	0.0008	-0.8017
I-RIM	-4.6999	0.0	4.0	461.0	0.0008	-0.8581
I-TM	-4.741	0.0	2.0	463.0	0.0008	-0.8656
I-HMTS	-4.6587	0.0	6.0	459.0	0.0008	-0.8506
I-TMHS	-4.6382	0.0	7.0	458.0	0.0008	-0.8468
I-FMTH	-4.741	0.0	2.0	463.0	0.0008	-0.8656
FM-T	-4.741	0.0	463.0	2.0	0.0008	-0.8656
FM-HM	-0.3599	0.7189	250.0	215.0	0.0008	0
FM-IM	-0.2982	0.7655	218.0	247.0	0.0008	0
FM-RIM	-3.4658	0.0005	64.0	401.0	0.0008	-0.6328
FM-TM	-1.2032	0.2289	174.0	291.0	0.0008	0
FM-HMTS	-1.6969	0.0897	150.0	315.0	0.0008	0
FM-TMHS	-2.3345	0.0196	119.0	346.0	0.0008	0
FM-FMTH	-0.8947	0.3709	189.0	276.0	0.0008	0
T-HM	-4.6176	0.0	8.0	457.0	0.0008	-0.8431
T-IM	-4.5559	0.0	11.0	454.0	0.0008	-0.8318
T-RIM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
T-TM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
T-HMTS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
T-TMHS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
T-FMTH	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
HM-IM	-0.8742	0.382	190.0	275.0	0.0008	0
HM-RIM	-4.124	0.0	32.0	433.0	0.0008	-0.7529
HM-TM	-2.0465	0.0407	133.0	332.0	0.0008	0
HM-HMTS	-2.7253	0.0064	100.0	365.0	0.0008	0
HM-TMHS	-2.643	0.0082	104.0	361.0	0.0008	0
HM-FMTH	-1.4501	0.147	162.0	303.0	0.0008	0
IM-RIM	-3.4452	0.0006	65.0	400.0	0.0008	-0.629
IM-TM	-1.2444	0.2134	172.0	293.0	0.0008	0
IM-HMTS	-1.2855	0.1986	170.0	295.0	0.0008	0
IM-TMHS	-1.7175	0.0859	149.0	316.0	0.0008	0
IM-FMTH	-1.2032	0.2289	174.0	291.0	0.0008	0
RIM-TM	-2.6019	0.0093	359.0	106.0	0.0008	0
RIM-HMTS	-2.5402	0.0111	356.0	109.0	0.0008	0
RIM-TMHS	-2.1494	0.0316	337.0	128.0	0.0008	0
RIM-FMTH	-3.0338	0.0024	380.0	85.0	0.0008	0
TM-HMTS	-0.216	0.829	222.0	243.0	0.0008	0
TM-TMHS	-1.1004	0.2712	179.0	286.0	0.0008	0
TM-FMTH	-0.2365	0.813	244.0	221.0	0.0008	0
HMTS-TMHS	-0.7096	0.4779	198.0	267.0	0.0008	0
HMTS-FMTH	-0.7507	0.4528	269.0	196.0	0.0008	0
TMHS-FMTH	-0.9976	0.3185	281.0	184.0	0.0008	0

Table A.98: Wilcoxon Signed Ranks - Medium Scenario

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-2.4373	0.0148	351.0	114.0	0.0008	0
SGP-I	-1.9437	0.0519	327.0	138.0	0.0008	0
SGP-FM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-T	-4.0005	0.0001	427.0	38.0	0.0008	-0.7304
SGP-HM	-4.7616	0.0	1.0	464.0	0.0008	-0.8693
SGP-IM	-4.7204	0.0	3.0	462.0	0.0008	-0.8618
SGP-RIM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
SGP-TM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
SGP-HMTS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
SGP-TMHS	-4.7616	0.0	1.0	464.0	0.0008	-0.8693
SGP-FMTH	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
TH-I	-0.9359	0.3493	187.0	278.0	0.0008	0
TH-FM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
TH-T	-2.2522	0.0243	342.0	123.0	0.0008	0
TH-HM	-4.741	0.0	2.0	463.0	0.0008	-0.8656
TH-IM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
TH-RIM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
TH-TM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
TH-HMTS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
TH-TMHS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
TH-FMTH	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
I-FM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
I-T	-2.9721	0.003	377.0	88.0	0.0008	0
I-HM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
I-IM	-4.6793	0.0	5.0	460.0	0.0008	-0.8543
I-RIM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
I-TM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
I-HMTS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
I-TMHS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
I-FMTH	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
FM-T	-4.7821	0.0	465.0	0.0	0.0008	-0.8731
FM-HM	-1.594	0.1109	155.0	310.0	0.0008	0
FM-IM	-1.0798	0.2802	285.0	180.0	0.0008	0
FM-RIM	-1.3678	0.1714	166.0	299.0	0.0008	0
FM-TM	-1.0593	0.2895	181.0	284.0	0.0008	0
FM-HMTS	-4.4119	0.0	18.0	447.0	0.0008	-0.8055
FM-TMHS	-3.9183	0.0001	42.0	423.0	0.0008	-0.7154
FM-FMTH	-4.0211	0.0001	37.0	428.0	0.0008	-0.7342
T-HM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
T-IM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
T-RIM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
T-TM	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
T-HMTS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
T-TMHS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
T-FMTH	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
HM-IM	-2.9721	0.003	377.0	88.0	0.0008	0
HM-RIM	-0.7713	0.4405	270.0	195.0	0.0008	0
HM-TM	-0.5451	0.5857	259.0	206.0	0.0008	0
HM-HMTS	-3.548	0.0004	60.0	405.0	0.0008	-0.6478
HM-TMHS	-3.0133	0.0026	86.0	379.0	0.0008	0
HM-FMTH	-3.0955	0.002	82.0	383.0	0.0008	0
IM-RIM	-2.1083	0.035	130.0	335.0	0.0008	0

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
IM-TM	-2.1288	0.0333	129.0	336.0	0.0008	0
IM-HMTS	-4.453	0.0	16.0	449.0	0.0008	-0.813
IM-TMHS	-3.9594	0.0001	40.0	425.0	0.0008	-0.7229
IM-FMTH	-4.6999	0.0	4.0	461.0	0.0008	-0.8581
RIM-TM	-0.1954	0.8451	223.0	242.0	0.0008	0
RIM-HMTS	-3.8566	0.0001	45.0	420.0	0.0008	-0.7041
RIM-TMHS	-3.4246	0.0006	66.0	399.0	0.0008	-0.6252
RIM-FMTH	-4.3913	0.0	19.0	446.0	0.0008	-0.8017
TM-HMTS	-3.7332	0.0002	51.0	414.0	0.0008	-0.6816
TM-TMHS	-3.3835	0.0007	68.0	397.0	0.0008	-0.6177
TM-FMTH	-3.3218	0.0009	71.0	394.0	0.0008	0
HMTS-TMHS	-0.3394	0.7343	249.0	216.0	0.0008	0
HMTS-FMTH	-1.2855	0.1986	295.0	170.0	0.0008	0
TMHS-FMTH	-0.3394	0.7343	249.0	216.0	0.0008	0

Table A.99: Wilcoxon Signed Ranks - Hard Scenario

Best of Generation

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-1.1136	0.2655	166.0	269.0	0.0008	0
SGP-I	-1.2444	0.2134	172.0	293.0	0.0008	0
SGP-FM	-0.7507	0.4528	196.0	269.0	0.0008	0
SGP-T	-2.2934	0.0218	121.0	344.0	0.0008	0
SGP-HM	-0.7919	0.4284	194.0	271.0	0.0008	0
SGP-IM	-2.9516	0.0032	89.0	376.0	0.0008	0
SGP-RIM	-4.3502	0.0	21.0	444.0	0.0008	-0.7942
SGP-TM	-1.5323	0.1254	158.0	307.0	0.0008	0
SGP-HMTS	-3.6097	0.0003	57.0	408.0	0.0008	-0.659
SGP-TMHS	-3.0955	0.002	82.0	383.0	0.0008	0
SGP-FMTH	-2.7047	0.0068	101.0	364.0	0.0008	0
TH-I	-0.5245	0.5999	207.0	258.0	0.0008	0
TH-FM	-0.0514	0.959	230.0	235.0	0.0008	0
TH-T	-1.6352	0.102	153.0	312.0	0.0008	0
TH-HM	-0.2571	0.7971	245.0	220.0	0.0008	0
TH-IM	-2.0465	0.0407	133.0	332.0	0.0008	0
TH-RIM	-4.0417	0.0001	36.0	429.0	0.0008	-0.7379
TH-TM	-0.3599	0.7189	215.0	250.0	0.0008	0
TH-HMTS	-2.5813	0.0098	107.0	358.0	0.0008	0
TH-TMHS	-1.7175	0.0859	149.0	316.0	0.0008	0
TH-FMTH	-1.8614	0.0627	142.0	323.0	0.0008	0
I-FM	-0.9153	0.36	277.0	188.0	0.0008	0
I-T	-1.5118	0.1306	159.0	306.0	0.0008	0
I-HM	-0.8742	0.382	275.0	190.0	0.0008	0
I-IM	-0.977	0.3286	185.0	280.0	0.0008	0
I-RIM	-3.7126	0.0002	52.0	413.0	0.0008	-0.6778
I-TM	-0.1337	0.8936	226.0	239.0	0.0008	0
I-HMTS	-1.7175	0.0859	149.0	316.0	0.0008	0
I-TMHS	-0.7919	0.4284	194.0	271.0	0.0008	0

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
I-FMTH	-0.9976	0.3185	184.0	281.0	0.0008	0
FM-T	-1.6352	0.102	153.0	312.0	0.0008	0
FM-HM	-0.2777	0.7813	219.0	246.0	0.0008	0
FM-IM	-1.8614	0.0627	142.0	323.0	0.0008	0
FM-RIM	-4.2062	0.0	28.0	437.0	0.0008	-0.7679
FM-TM	-0.6273	0.5304	202.0	263.0	0.0008	0
FM-HMTS	-2.3756	0.0175	117.0	348.0	0.0008	0
FM-TMHS	-2.0671	0.0387	132.0	333.0	0.0008	0
FM-FMTH	-1.7586	0.0786	147.0	318.0	0.0008	0
T-HM	-1.6146	0.1064	311.0	154.0	0.0008	0
T-IM	-0.1337	0.8936	226.0	239.0	0.0008	0
T-RIM	-2.6636	0.0077	103.0	362.0	0.0008	0
T-TM	-1.1415	0.2536	288.0	177.0	0.0008	0
T-HMTS	-0.7302	0.4653	197.0	268.0	0.0008	0
T-TMHS	-0.1131	0.9099	238.0	227.0	0.0008	0
T-FMTH	-0.1131	0.9099	227.0	238.0	0.0008	0
HM-IM	-2.0877	0.0368	131.0	334.0	0.0008	0
HM-RIM	-4.3708	0.0	20.0	445.0	0.0008	-0.798
HM-TM	-0.3805	0.7036	214.0	251.0	0.0008	0
HM-HMTS	-2.4579	0.014	113.0	352.0	0.0008	0
HM-TMHS	-1.9848	0.0472	136.0	329.0	0.0008	0
HM-FMTH	-1.6352	0.102	153.0	312.0	0.0008	0
IM-RIM	-2.7664	0.0057	98.0	367.0	0.0008	0
IM-TM	-1.2444	0.2134	293.0	172.0	0.0008	0
IM-HMTS	-1.0387	0.2989	182.0	283.0	0.0008	0
IM-TMHS	-0.3394	0.7343	249.0	216.0	0.0008	0
IM-FMTH	-0.072	0.9426	229.0	236.0	0.0008	0
RIM-TM	-3.7537	0.0002	415.0	50.0	0.0008	-0.6853
RIM-HMTS	-2.5196	0.0117	355.0	110.0	0.0008	0
RIM-TMHS	-3.2807	0.001	392.0	73.0	0.0008	0
RIM-FMTH	-2.8076	0.005	369.0	96.0	0.0008	0
TM-HMTS	-2.9516	0.0032	89.0	376.0	0.0008	0
TM-TMHS	-1.4501	0.147	162.0	303.0	0.0008	0
TM-FMTH	-1.265	0.2059	171.0	294.0	0.0008	0
HMTS-TMHS	-0.6479	0.517	264.0	201.0	0.0008	0
HMTS-FMTH	-0.8947	0.3709	276.0	189.0	0.0008	0
TMHS-FMTH	-0.2365	0.813	221.0	244.0	0.0008	0

Table A.100: Wilcoxon Signed Ranks - Easy Scenario

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-0.0309	0.9754	234.0	231.0	0.0008	0
SGP-I	-0.4011	0.6884	252.0	213.0	0.0008	0
SGP-FM	-2.9927	0.0028	87.0	378.0	0.0008	0
SGP-T	-0.0103	0.9918	233.0	232.0	0.0008	0
SGP-HM	-2.4373	0.0148	114.0	351.0	0.0008	0
SGP-IM	-2.4579	0.014	113.0	352.0	0.0008	0
SGP-RIM	-4.7204	0.0	3.0	462.0	0.0008	-0.8618
SGP-TM	-3.7743	0.0002	49.0	416.0	0.0008	-0.6891

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-HMTS	-3.9594	0.0001	40.0	425.0	0.0008	-0.7229
SGP-TMHS	-4.2062	0.0	28.0	437.0	0.0008	-0.7679
SGP-FMTH	-3.8771	0.0001	44.0	421.0	0.0008	-0.7079
TH-I	-0.1337	0.8936	226.0	239.0	0.0008	0
TH-FM	-3.2807	0.001	73.0	392.0	0.0008	0
TH-T	-0.072	0.9426	236.0	229.0	0.0008	0
TH-HM	-3.5069	0.0005	62.0	403.0	0.0008	-0.6403
TH-IM	-3.1367	0.0017	80.0	385.0	0.0008	0
TH-RIM	-4.7204	0.0	3.0	462.0	0.0008	-0.8618
TH-TM	-4.124	0.0	32.0	433.0	0.0008	-0.7529
TH-HMTS	-4.5148	0.0	13.0	452.0	0.0008	-0.8243
TH-TMHS	-4.3502	0.0	21.0	444.0	0.0008	-0.7942
TH-FMTH	-4.2679	0.0	25.0	440.0	0.0008	-0.7792
I-FM	-2.8899	0.0039	92.0	373.0	0.0008	0
I-T	-0.5039	0.6143	208.0	257.0	0.0008	0
I-HM	-3.1161	0.0018	81.0	384.0	0.0008	0
I-IM	-2.5813	0.0098	107.0	358.0	0.0008	0
I-RIM	-4.3296	0.0	22.0	443.0	0.0008	-0.7905
I-TM	-4.0417	0.0001	36.0	429.0	0.0008	-0.7379
I-HMTS	-4.3091	0.0	23.0	442.0	0.0008	-0.7867
I-TMHS	-4.1445	0.0	31.0	434.0	0.0008	-0.7567
I-FMTH	-4.1034	0.0	33.0	432.0	0.0008	-0.7492
FM-T	-2.9927	0.0028	378.0	87.0	0.0008	0
FM-HM	-0.2571	0.7971	245.0	220.0	0.0008	0
FM-IM	-0.2777	0.7813	219.0	246.0	0.0008	0
FM-RIM	-3.6097	0.0003	57.0	408.0	0.0008	-0.659
FM-TM	-1.2444	0.2134	172.0	293.0	0.0008	0
FM-HMTS	-2.0054	0.0449	135.0	330.0	0.0008	0
FM-TMHS	-2.7047	0.0068	101.0	364.0	0.0008	0
FM-FMTH	-1.121	0.2623	178.0	287.0	0.0008	0
T-HM	-3.0133	0.0026	86.0	379.0	0.0008	0
T-IM	-2.9104	0.0036	91.0	374.0	0.0008	0
T-RIM	-4.7616	0.0	1.0	464.0	0.0008	-0.8693
T-TM	-4.1857	0.0	29.0	436.0	0.0008	-0.7642
T-HMTS	-4.5559	0.0	11.0	454.0	0.0008	-0.8318
T-TMHS	-4.6176	0.0	8.0	457.0	0.0008	-0.8431
T-FMTH	-4.2679	0.0	25.0	440.0	0.0008	-0.7792
HM-IM	-0.7507	0.4528	196.0	269.0	0.0008	0
HM-RIM	-4.1857	0.0	29.0	436.0	0.0008	-0.7642
HM-TM	-1.6763	0.0937	151.0	314.0	0.0008	0
HM-HMTS	-2.6636	0.0077	103.0	362.0	0.0008	0
HM-TMHS	-2.6842	0.0073	102.0	363.0	0.0008	0
HM-FMTH	-1.4912	0.1359	160.0	305.0	0.0008	0
IM-RIM	-3.4658	0.0005	64.0	401.0	0.0008	-0.6328
IM-TM	-1.121	0.2623	178.0	287.0	0.0008	0
IM-HMTS	-1.5529	0.1204	157.0	308.0	0.0008	0
IM-TMHS	-1.8409	0.0656	143.0	322.0	0.0008	0
IM-FMTH	-1.3061	0.1915	169.0	296.0	0.0008	0
RIM-TM	-2.7459	0.006	366.0	99.0	0.0008	0

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
RIM-HMTS	-2.6019	0.0093	359.0	106.0	0.0008	0
RIM-TMHS	-1.9437	0.0519	327.0	138.0	0.0008	0
RIM-FMTH	-3.1367	0.0017	385.0	80.0	0.0008	0
TM-HMTS	-0.5245	0.5999	207.0	258.0	0.0008	0
TM-TMHS	-1.4501	0.147	162.0	303.0	0.0008	0
TM-FMTH	-0.1748	0.8612	224.0	241.0	0.0008	0
HMTS-TMHS	-0.7302	0.4653	197.0	268.0	0.0008	0
HMTS-FMTH	-0.8124	0.4165	272.0	193.0	0.0008	0
TMHS-FMTH	-1.1827	0.2369	290.0	175.0	0.0008	0

Table A.101: Wilcoxon Signed Ranks - Medium Scenario

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
SGP-TH	-0.2571	0.7971	220.0	245.0	0.0008	0
SGP-I	-0.2777	0.7813	246.0	219.0	0.0008	0
SGP-FM	-4.1857	0.0	29.0	436.0	0.0008	-0.7642
SGP-T	-2.3139	0.0207	345.0	120.0	0.0008	0
SGP-HM	-4.2679	0.0	25.0	440.0	0.0008	-0.7792
SGP-IM	-3.548	0.0004	60.0	405.0	0.0008	-0.6478
SGP-RIM	-4.453	0.0	16.0	449.0	0.0008	-0.813
SGP-TM	-4.4736	0.0	15.0	450.0	0.0008	-0.8168
SGP-HMTS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
SGP-TMHS	-4.7204	0.0	3.0	462.0	0.0008	-0.8618
SGP-FMTH	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
TH-I	-0.0514	0.959	230.0	235.0	0.0008	0
TH-FM	-4.1857	0.0	29.0	436.0	0.0008	-0.7642
TH-T	-2.3756	0.0175	348.0	117.0	0.0008	0
TH-HM	-4.1651	0.0	30.0	435.0	0.0008	-0.7604
TH-IM	-3.7126	0.0002	52.0	413.0	0.0008	-0.6778
TH-RIM	-4.3708	0.0	20.0	445.0	0.0008	-0.798
TH-TM	-4.7204	0.0	3.0	462.0	0.0008	-0.8618
TH-HMTS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
TH-TMHS	-4.7616	0.0	1.0	464.0	0.0008	-0.8693
TH-FMTH	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
I-FM	-3.8977	0.0001	43.0	422.0	0.0008	-0.7116
I-T	-1.9026	0.0571	325.0	140.0	0.0008	0
I-HM	-4.5148	0.0	13.0	452.0	0.0008	-0.8243
I-IM	-3.3424	0.0008	70.0	395.0	0.0008	0
I-RIM	-4.7204	0.0	3.0	462.0	0.0008	-0.8618
I-TM	-4.1857	0.0	29.0	436.0	0.0008	-0.7642
I-HMTS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
I-TMHS	-4.7616	0.0	1.0	464.0	0.0008	-0.8693
I-FMTH	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
FM-T	-4.3913	0.0	446.0	19.0	0.0008	-0.8017
FM-HM	-1.738	0.0822	148.0	317.0	0.0008	0
FM-IM	-0.9976	0.3185	281.0	184.0	0.0008	0
FM-RIM	-1.5529	0.1204	157.0	308.0	0.0008	0
FM-TM	-1.1415	0.2536	177.0	288.0	0.0008	0
FM-HMTS	-4.5148	0.0	13.0	452.0	0.0008	-0.8243

Alg1-Alg2	Z	P-value	P rank	N rank	Alpha	R
FM-TMHS	-3.9594	0.0001	40.0	425.0	0.0008	-0.7229
FM-FMTH	-4.0828	0.0	34.0	431.0	0.0008	-0.7454
T-HM	-4.5148	0.0	13.0	452.0	0.0008	-0.8243
T-IM	-4.2885	0.0	24.0	441.0	0.0008	-0.783
T-RIM	-4.6793	0.0	5.0	460.0	0.0008	-0.8543
T-TM	-4.7616	0.0	1.0	464.0	0.0008	-0.8693
T-HMTS	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
T-TMHS	-4.7616	0.0	1.0	464.0	0.0008	-0.8693
T-FMTH	-4.7821	0.0	0.0	465.0	0.0008	-0.8731
HM-IM	-2.9516	0.0032	376.0	89.0	0.0008	0
HM-RIM	-0.6479	0.517	264.0	201.0	0.0008	0
HM-TM	-0.7096	0.4779	267.0	198.0	0.0008	0
HM-HMTS	-3.6714	0.0002	54.0	411.0	0.0008	-0.6703
HM-TMHS	-2.931	0.0034	90.0	375.0	0.0008	0
HM-FMTH	-3.1572	0.0016	79.0	386.0	0.0008	0
IM-RIM	-2.2111	0.027	125.0	340.0	0.0008	0
IM-TM	-2.1288	0.0333	129.0	336.0	0.0008	0
IM-HMTS	-4.4942	0.0	14.0	451.0	0.0008	-0.8205
IM-TMHS	-4.0005	0.0001	38.0	427.0	0.0008	-0.7304
IM-FMTH	-4.7204	0.0	3.0	462.0	0.0008	-0.8618
RIM-TM	-0.0926	0.9263	228.0	237.0	0.0008	0
RIM-HMTS	-3.8771	0.0001	44.0	421.0	0.0008	-0.7079
RIM-TMHS	-3.4246	0.0006	66.0	399.0	0.0008	-0.6252
RIM-FMTH	-4.3091	0.0	23.0	442.0	0.0008	-0.7867
TM-HMTS	-3.7332	0.0002	51.0	414.0	0.0008	-0.6816
TM-TMHS	-3.4246	0.0006	66.0	399.0	0.0008	-0.6252
TM-FMTH	-3.4452	0.0006	65.0	400.0	0.0008	-0.629
HMTS-TMHS	-0.4422	0.6583	254.0	211.0	0.0008	0
HMTS-FMTH	-1.4501	0.147	303.0	162.0	0.0008	0
TMHS-FMTH	-0.3188	0.7499	248.0	217.0	0.0008	0

Table A.102: Wilcoxon Signed Ranks - Hard Scenario