FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Designing an Extensible PBT Framework to Support Multiple Search Strategies

Rui Jorge Leão Guedes

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Hugo Sereno Ferreira, Assistant Professor

Second Supervisor: André Restivo, Assistant Professor

External Supervisor: Rui Gonçalves, M.Sc.

July 28, 2021

Designing an Extensible PBT Framework to Support Multiple Search Strategies

Rui Jorge Leão Guedes

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Prof. João Carlos Viegas Martins Bispo

External Examiner: Prof. Pedro Manuel Pinto Ribeiro

Supervisor: Prof. Hugo José Sereno Lopes Ferreira

July 28, 2021

Abstract

Testing, both static and dynamic, is a critical activity for detecting defects and ensuring software compliance according to a specification. One way to describe software specifications is by using formal methods; another one is by using tests. Tests are often written as a series of examples, but in Property-Based Testing (PBT), high-level properties — that a program must fulfill — are written instead. PBT is a software testing technique that allows us to transform such properties into a specific and valid representation, and ensure their validity. It combines program-specific properties with the concept of automatic test data generation to assure that a given program satisfies a collection of specified properties.

Currently, most PBT tools' search strategies are purely random. However, to increase the confidence over the properties under validation, some tools implement biased sampling strategies that, by considering a subset of potential edge-cases, allow testing in a more deterministic manner. Alternatively of being random-based, other tools simply implement exhaustive search strategies in order to cover, or at least try to cover, the entire search space. Nevertheless, this last approach is not widely implemented since the search space is typically quite large.

The random concept's heavy dominance in PBT tools' search strategies composes a critical problem in the PBT domain since it implies a total arbitrary search over the input space, thus not ensuring a fair distribution of test cases and a reliable coverage of the program's critical paths. Consequently, several types of defects can still elude current search strategies. Moreover, existing tools are also very limited in allowing the developer to fine-tune, tweak, or even extend the provided search strategies.

Previous works have shown that several solutions from multiple domains attempt to mitigate this problem. Nevertheless, such solution's results tend to be problem-dependent since they work particularly well only under certain conditions. Moreover, there is no evidence of a solution whose strategy is capable of outperforming all of its alternatives.

In this work, instead of developing an innovative and meta-search strategy, we propose the endowment of the FluentCheck framework with an easily extensible and scalable architecture capable of supporting several different search strategies from various domains. Striving to make out the most of these architectural characteristics, we also propose an entirely new coverage-guided search strategy and the implementation of the constants extraction and pairwise testing search strategy features.

The proposed solution was evaluated by comparing several evaluation metrics resulting from the execution of sixteen unique search strategies over six distinct benchmarks. According to the results achieved, we were able to conclude that the success of each search strategy cannot be generally determined since it heavily depends on the scenario in which the strategy is applied, which validated the hypothesis underlying the developed solution. We further identified a series of promising paths to be explored in the future.

Keywords: Property-Based Testing, Search-Based Testing, Coverage-Guided Fuzzing, Concolic Testing, Hybrid Fuzzing, Constants Extraction, Pairwise Testing

Resumo

Testes, tanto estáticos como dinâmicos, são uma atividade crítica para detectar defeitos e garantir a conformidade do software de acordo com uma especificação. Uma maneira de descrever as especificações de software é usando métodos formais; outra é usando testes. Os testes são geralmente descritos como uma série de exemplos, mas em Property-Based Testing (PBT), propriedades de alto nível — que um programa deve cumprir — são definidas. PBT é uma técnica de teste de software que nos permite transformar tais propriedades numa representação específica e válida, e garantir sua validade. Esta técnica combina propriedades específicas de um programa com o conceito de geração automática de casos de teste para garantir que um dado programa satisfaz um conjunto de propriedades específicas.

Atualmente, a maioria das estratégias de pesquisa das ferramentas de PBT são puramente aleatórias. Contudo, para aumentar a confiança sobre as propriedades perante validação, algumas ferramentas implementam estratégias de amostragem enviesada que, ao considerar um subconjunto de potenciais casos extremos, permitem testar de forma mais determinística. Alternativamente, outras ferramentas simplesmente implementam estratégias de pesquisa exaustivas para cobrir, ou pelo menos tentar cobrir, todo o espaço de pesquisa. No entanto, esta última abordagem não é extensamente implementada, uma vez que o espaço de pesquisa é tipicamente vasto.

O forte domínio do conceito aleatório nas estratégias de pesquisa das ferramentas PBT constitui um problema crítico no domínio de PBT, uma vez que implica uma pesquisa totalmente arbitrária sobre o espaço de pesquisa, não garantindo assim uma distribuição justa dos casos de teste e uma cobertura de confiança dos caminhos críticos do programa. Consequentemente, deste modo vários tipos de defeitos podem ainda iludir as estratégias de pesquisa atuais. Além disso, as ferramentas existentes também são muito limitadas no que diz respeito a permitir ao desenvolvedor ajustar, ou mesmo estender, as estratégias de pesquisa suportadas.

Trabalhos passados mostram que várias soluções de múltiplos domínios tentam mitigar este problema. No entanto, os resultados dessas soluções tendem a ser dependentes do problema a ser resolvido, uma vez que funcionam particularmente bem apenas sob certas condições. Além disso, não há evidência de uma solução cuja estratégia seja capaz de superar todas as suas alternativas.

Neste trabalho, em vez de desenvolvermos uma estratégia de pesquisa meta e inovadora, propomos dotar a framework FluentCheck de uma arquitetura facilmente extensível e escalável capaz de suportar diversas estratégias de pesquisa de diversos domínios. De forma a tirar o máximo proveito das características arquitetónicas mencionadas, também propomos uma estratégia de pesquisa guiada por cobertura totalmente nova e ainda a implementação das funcionalidades de extração de constantes e de testagem de pares.

A solução proposta foi avaliada através da comparação de várias métricas de avaliação resultantes da execução de dezasseis estratégias de pesquisa únicas em seis benchmarks distintas. De acordo com os resultados obtidos, pudemos concluir que o sucesso de cada estratégia de pesquisa não pode ser determinado de forma geral, pois depende vigorosamente do cenário em que a estratégia é aplicada, o que acaba por validar a hipótese subjacente à solução desenvolvida. Por fim, identificamos ainda uma série de caminhos promissores a serem explorados no futuro.

Acknowledgements

Even though I am not good with words, I would like to thank all the people that made this dissertation's development possible.

First and foremost, my sincere gratitude goes to my parents and brother for always being there for me, for supporting me throughout my entire life, and for making me the person I am today. A special thanks to my brother for all the support, including listening to all my problems and dilemmas, and for all the revisions made.

To Daniela Moreira, my girlfriend and a critical pillar of both this dissertation and my life, a huge thank you for the emotional support, for motivating me throughout this entire journey, and for constantly reminding me what I am capable of accomplishing.

Also, I would like to thank my friends, César Pinho and João Barbosa, for their friendship throughout the entire master's degree and for all the jokes and knowledge shared.

Lastly, but not least, I would also like to thank my supervisors, Hugo Sereno Ferreira and André Restivo, not only for the opportunity provided but also for the valuable knowledge supplied and for constantly pushing me to do better and better.

Rui Guedes

"It always seems impossible until it's done."

Nelson Mandela

Contents

List of Figures

List of Tables

Abbreviations

ISTQB	Internet Software Testing Qualifications Board
SBT	Search-Based Testing
MBT	Model-Based Testing
PBT	Property-Based Testing
SUT	System Under Test
SBST	Search-Based Software Testing
SBSE	Search-Based Software Engineering
CGF	Coverage-Guided Testing
AFL	American Fuzzy Lop
SLR	Systematic Literature Review
PDF	Portable Document Format
ART	Adaptive Random Testing
APSO	Adaptive Particle Swarm Optimization
PSO	Particle Swarm Optimization
CFG	Control-Flow Graph
MC/DC	Modified Condition Decision Coverage
MTTF	Mean Time to Failure

Chapter 1

Introduction

This chapter introduces the context and motivation of this dissertation, as well as the problem it aims to solve and the goals behind the proposed solution. Section 1.1 describes the context of this work. Section 1.2 explains the motivation behind the proposed work. Section 1.3 details the problem under study. Section 1.4 describes the general goal of this work's proposed solution. Finally, Section 1.5 details the structure of the remaining chapters of this document.

1.1 Context

Throughout software development history, testing activities, either static or dynamic, have been critical in finding failures and increasing confidence in the software's correctness and assessing its quality.

Software can be tested at several different levels using either white-box (structural) or black-box (functional) test case design techniques. Depending on the technique adopted, numerous quality attributes can be considered [22]. Security, robustness, performance, usability, reliability, and functionality compose the list of the most frequent quality attributes used in software specifications.

As of today, several software testing techniques have been proposed, and each technique has its unique way of describing software specifications. One possible way of describing such specifications is by using formal methods; another is by using tests, which are often written as a series of practical examples. In Property-Based Testing (PBT) [13, 14], a more recent software testing technique, high-level properties that a program must fulfill are defined instead. Such properties are responsible for representing partial specifications of the overall System Under Test (SUT) specification.

Besides the heavy reliance on the concept related to the definition of program-specific properties to assure that a given system satisfies a collection of specified properties, PBT also heavily relies on the automatic test data generation concept. Both of these concepts are crucial during the properties' validation process.

1.2 Motivation

One key component of PBT is the support of automatic test data generation [21], which has several benefits when compared to the manual approach. The main differentiating factor between these two types of test data generation processes is the significant increase in speed and accuracy for generating higher volume levels of data that former introduces over the latter.

Random, Goal Oriented, Intelligent, and Path Oriented test data generators compose some of the most used automatic test data generators by software testing techniques [21].

In PBT, most of the tools' generators are purely random. However, due to randomness implicit drawbacks, some PBT tools implement biased sampling strategies that, by considering a subset of potential edge-cases, allow the increase in confidence over the properties under validation. Alternatively of being random-based, some tools implement exhaustive generators that aim to cover the entire input space. Nevertheless, this exhaustive approach is not widely adopted because exhaustive testing is most of the times impossible since the input space is often quite vast.

Apart from the generators previously mentioned, some PBT tools (e.g., QuickTheories) have, more recently, started implementing generators that, by taking into account some form of feedback, guide the input generation process in an informed way. However, the implementation of such generators is still something experimental and not applicable to real-world scenarios.

1.3 Problem Definition

The heavy dominance of random generators in PBT tools composes a critical problem in the PBT domain since it implies a total arbitrary search over the input space.

Consequently, by being completely arbitrary, PBT generators do not ensure either a fair distribution of test-cases in the search space or a reliable coverage of the program's critical paths. Therefore, several types of defects can still elude current search strategies.

In order to better understand how such strategies can be eluded, let us consider the practical example represented in Listing 1.1, where two custom functions responsible for the sum of two numbers are defined.

```
myAdd(a,b) { return a == 13371337 ? 0xC0FF33 : a + b }
mySecretAdd(a,b) { return SHA1(a) == 13371337 ? 0xC0FF33 : a + b }
```

Listing 1.1: Random Search Strategies Limitations

By analyzing the defined functions, it is clear that both functions have the particularity of assuming two different behaviors depending on the satisfiability of the preconditions of `a`, or even more complicated `SHA1(a)`, being equal to `13371337`. As such, for all the cases where the preconditions are not met, the functions behave correctly; otherwise, the functions behave in a wholly different and erroneous manner.

Since current search strategies are mostly random-based, they have a lower probability of generating test cases satisfying the enumerated preconditions. Therefore, most of the time, such strategies will give false confidence over the SUT because they are not able to grant a reliable coverage of a system's possible behaviors.

Chapter 4 elaborates on the problem definition by describing it in greater detail and defining its underlying assumptions, scope, and hypothesis.

1.4 General Goal

This dissertation's primary goal is based on the absence of a search strategy capable of outperforming all the current alternatives in every possible scenario, and consists of complementing the current random-based search strategies, implemented by PBT tools, with new strategies that, by following some form of feedback, are capable of guiding the test data generation process in an informed and more efficient way.

Aiming to achieve the goal mentioned above, we present an unprecedented and enhanced version of the FluentCheck[1] [48] PBT framework, which given its easily extensible and scalable architecture, is capable of supporting multiple search strategies from various domains that, apart from the PBT domain itself, are also known to address the same problem this dissertation aims to solve. Furthermore, we also present an entirely new coverage-guided search strategy and two unique search strategy features based on the source code constants extraction and on pairwise testing. It is noteworthy that both the coverage-guided search strategy and the before-mentioned features are supported and implemented by the aforementioned framework.

1.5 Document Structure

Apart from this chapter, which introduced this dissertation's purpose by explaining its context, motivation, and the problem it aims to solve, this document is composed of seven more chapters, structured as follows:

[1] Available: https://github.com/fluent-check/fluent-check/tree/msc-ruiguedes-dev

- Chapter 2 (p. 5), **Background**, introduces the background's fundamental concepts needed to fully understand this dissertation.
- Chapter 3 (p. 15), **State of the Art**, describes the current state of the art regarding this dissertation's scope, using a Systematic Literature Review on PBT search strategies.
- Chapter 4 (p. 33), **Problem Statement**, presents the assumptions, scope, and hypothesis underlying the problem this dissertation aims to solve. It also briefly describes the proposed solution and the validation methodology.
- Chapter 5 (p. 37), **Designing an Extensible PBT Framework**, provides a detailed overview of all the objectives, requirements, and architectural design decisions that must be considered while implementing the proposed solution.
- Chapter 6 (p. 41), **Solution**, thoroughly details this dissertation's solution implementation process and all of its underlying decisions.
- Chapter 7 (p. 61), **Evaluation**, presents the evaluation process and analyzes to which extent the obtained results validate the developed solution.
- Chapter 8 (p. 85), **Conclusions**, concludes this dissertation by summarizing the developed work, and corresponding findings, and by detailing this dissertation's difficulties, main contributions, and future work.

Chapter 2

Background

Throughout this chapter, we provide the fundamental concepts surrounding this dissertation's scope. Section 2.1 presents an overview of Software Testing. Section 2.2 describes PBT's essence, while its sub-sections detail its essential components. Section 2.2.1 addresses the concept of properties and how they are specified. Section 2.2.2 describes how PBT tools generate test data. Section 2.2.3 explains how PBT counter-examples are generated and further processed. Lastly, Section 2.2.4 presents and analyses PBT tools implemented in some of the most popular programming languages. Section 2.3 explains the concept of Search-Based Testing, how it works, its objectives, and the algorithms used. Section 2.4 describes Coverage-Guided Fuzzing, including the current state of the art and the workflow of coverage-guided fuzzers. Section 2.5 presents an overview and an analysis of Concolic Testing. Finally, Section 2.6 explains the Hybrid Fuzzing concept and correlates it with the previous two sections.

2.1 Software Testing

The Software Testing paradigm is defined by the Internet Software Testing Qualifications Board (ISTQB) [23] as:

> *"The process consisting of all lifecycle activities, both static and dynamic, concerned with planning, preparation and evaluation of software products and related work products to determine that they satisfy specified requirements, to demonstrate that they are fit for purpose and to detect defects."*

Software Testing's primary goals consist of finding failures and increasing confidence in the software's correctness and assessing its quality. It can be performed at several different levels ranging all the way from unit, integration, and system testing to acceptance testing. Regardless of the test level, there are no restrictions on the test criteria used, being possible to perform non-functional, functional (black-box), and structural (white-box) testing. Depending on the test criteria selected, diverse quality attributes (e.g., security, robustness, performance, etc) can be considered. According to DeMillo [10], a typical black-box criterion is performance, while in white-box testing, structural and fault coverage are the most common criteria.

Unit Testing, Mutation Testing, Search-Based Testing (SBT), Concolic Testing, Model-Based Testing (MBT), and PBT are just some of all the numerous techniques developed to perform Software Testing.

Although all Software Testing techniques aim to detect defects and ensure software compliance according to a specification, each technique has its own way of describing software specifications. One possible way is by using formal methods; another one is by using tests (e.g., Unit Testing, Mutation Testing), which are often written as a series of examples. In PBT, high-level properties that a program must fulfill are written instead.

2.2 Property-Based Testing

PBT [13, 14] is a testing technique that allows us to transform high-level properties into a specific and valid executable representation. It combines program-specific properties with the concept of automatic test data generation to assure that a given program satisfies a collection of specified properties.

The way PBT validates such properties is relatively straightforward. Given a specific property and a valid executable representation, a PBT tool successively generates, often randomized, inputs and repeatedly executes the SUT in order to verify if the property is satisfiable. In the case of a property being falsified, the PBT tool generates a counter-example, which is further minimized through a process called shrinking.

2.2.1 Properties Specification

The properties used by PBT tools can be seen as partial specifications of the overall SUT specification. Properties should be compact, easy to understand, and written in the form of executable specifications so that PBT tools can adequately use them.

The definition of properties varies according to the used tool's notation. While defining such properties, developers should get the most out of the tool capabilities so that a wide variety of input-output relations can be accurately described.

In order to better understand how properties are specified, let us consider a practical example of a system in which basic mathematical operations (e.g., Addition, Subtraction, Multiplication, etc) are implemented. Even though we can use several properties to test our system, properties such as the Commutative, Associative, and the Identity properties immediately stand out. Listings 2.1 and 2.2 depict how the Commutative and Associative addition properties are specified in the FluentCheck framework, respectively.

```
it(’finds if addition is commutative’,
    () => {
expect(fc.scenario()
  .forall(’a’, fc.integer(-10, 10))
  .forall(’b’, fc.integer(-10, 10))
  .then(({a, b}) => a + b === b + a)
  .check()
).to.have.property(’satisfiable’, true)
})
```

Listing 2.1: FluentCheck's Addition Commutative Property Specification

```
it(’finds if additions is associative’,
     () => {
expect(fc.scenario()
  .forall(’a’, fc.integer(-10, 10))
  .forall(’b’, fc.integer(-10, 10))
  .forall(’c’, fc.integer(-10, 10))
  .then(({a, b, c}) => (a + b) + c ===
      a + (b + c))
  .check()
).to.have.property(’satisfiable’, true)
})
```

Listing 2.2: FluentCheck's Addition Associative Property Specification

2.2.2 Generators

PBT generators are responsible for generating the test data used to test the properties of a particular system. The way such data is generated varies according to the selected framework.

Most PBT tools' generators are purely random, which means that the generated test data is randomly distributed all over the search space. Additionally, such generators sometimes combine the randomly generated test data with edge cases to increase the confidence over the tested properties. Alternatively of being random, some generators simply implement an exhaustive and bounded test data generation approach in order to cover, or at least try to cover, the entire search space [1, p. 6].

Regardless of the approach taken to generate test data, generators also vary and depend on the type of test data to be generated.

2.2.2.1 Arbitrary Types

PBT tools have built-in generators that enable the automatic generation of test data types often required by property specifications. Such generators compose one of the critical components of what PBT designates as an `Arbitrary`.

In the PBT domain, an `Arbitrary` is an abstract class that enables the frameworks to support the generation of several basic test data types. Most PBT tools comprise several `Arbitrary Types` (e.g., Integers, Booleans, Arrays, Sets, Reals, etc) that allow the developers to test several properties without needing to supply a custom generator.

As an example, we illustrate the `ArbitraryArray` and `ArbitrarySet` from the FluentCheck framework in Listings 2.3 and 2.4, respectively.

```
pick(): FluentPick<A[]> | undefined {
 const size = Math.floor(Math.random()
     * (this.max - this.min + 1))
     + this.min
 const fpa = this.arbitrary
     .sample(size)

 const value = fpa.map(v => v.value)
 const original = fpa.map(v =>
     v.original)

 return {
    value,
    original: original.every(o =>
        o === undefined) ?
            value : original
 }
}
```

Listing 2.3: FluentCheck's ArbitraryArray Generator

```
pick(): FluentPick<A[]> | undefined {
 const size = Math.floor(Math.random()
     * (this.max - this.min + 1))
     + this.min
 const pick = new Set<A>()

 while (pick.size !== size)
    pick.add(this.elements[Math.floor(
        Math.random() * this.elements.
        length)])

 const value = Array.from(pick).sort()

 return {
    value,
    original: value
 }
}
```

Listing 2.4: FluentCheck's ArbitrarySet Generator

2.2.2.2 User-Defined Types

Despite some property specifications requiring only basic test data types, some properties involve sparse preconditions coupled with restrictive constraints that lead to more complex test data types. Such test data can still be automatically generated. However, there is a risk of hindering the testing process efficiency by testing the system with semantically invalid inputs.

In order to address these test data types, also known as user-defined types, PBT tools allow the definition of custom generators, which enable the developers to control the testing process's effectiveness by controlling how test data is generated. Nevertheless, defining such generators can be complex and error-prone.

2.2.3 Shrinking

Whenever a property is falsified, the PBT framework generates a counter-example that enables the developer to understand how and why the property does not hold in SUT. However, sometimes the generated counter-example is so large or complex that understanding it becomes almost, or even, impossible.

Through a process designated as Shrinking, PBT tools address this problem by minimizing a generated counter-example until a point where the counter-example itself becomes easy enough to be understood by the developer.

From a more technical perspective, Shrinking is usually accomplished by transforming all the generators used for the counter-example generation and carry them back towards their neutral point. For instance, a *"number tends to shrink from floating point values towards integers, and integers tend to shrink towards the number 0."* [19].

Considering that there is no objective measure for the minimal counter-example, most tools recognize the Shrinking process as a best-effort, nice-to-have feature of PBT. In fact, the way each framework implements the Shrinking process varies from tool to tool. However, and regardless of how the process is implemented, the goal is always the same.

2.2.4 Popular Tools

PBT was pioneered by QuickCheck, a framework developed in Haskell by Koen Claessen and John Hughes and designed for functional programming [9]. Since then, QuickCheck has been emulated in several programming languages. Table 2.1 presents an overview of QuickCheck's re-implementations in some of the most popular programming languages.

Table 2.1: QuickCheck's re-implementations in some of the most popular programming languages.

Language	**Tools**
C/C++	`Theft`, `QuickCheck++`, `CppQuickCheck`, `RapidCheck`
C#	`FsCheck`
Python	`Hypothesis`
Java	`Functional Java`, `junit-quickcheck`, `jqwik`, `QuickTheories`, `jetCheck`
JavaScript	`JSCheck`, `JSVerify`, `fast-check`
TypeScript	`FluentCheck`, `fast-check`

The tools presented in Table 2.1 are not at the same development stage. Therefore, and besides the difference in the programming language used, the enumerated tools also differ in terms of the implemented features.

Regarding the `Arbitraries` supported, there are tools (e.g., fast-check, jqwik) that support a wide range of `Arbitrary` types while others (e.g., Theft, QuickCheck++) only support the most primitive data types. In terms of the *Shrinking* process, apart from the QuickCheck++ tool, all the remaining tools support this process either manually or automatically. Concerning the identified tools' test data *Generators*, which compose the current dissertation's main focus, every tool presents purely random test data generators by default. Additionally, some tools (e.g., FluentCheck, fast-check) couple the random test data generation with potential edge-cases to allow testing in a more deterministic manner. QuickTheories stands out from the rest of the tools because it includes an experimental feature that allows the generation of test data based on exercising new code paths (coverage guidance). However, such feature effectiveness is not yet known in real-world scenarios.

2.3 Search-Based Testing

Search-Based Software Testing (SBST), or simply SBT, is a sub-area of Search-Based Software Engineering (SBSE), which ends up being a sub-area of Software Engineering. These two areas have origins dating back to the 1970s, but they were only formally defined in 2001. However, it was only in the more recent years that such areas started to gain more attraction [17].

The main difference between these areas resides in the type of problems addressed. While, in general, SBSE addresses software engineering problems using metaheuristic search techniques, SBT is concerned only with using these techniques in testing problems.

The use of metaheuristic search techniques has been, over the years, deeply associated with automatic test data generation giving the notably costly and inefficient practice of manually generating such data. Since *"test data generation is an undecidable problem"* [33], the application of these techniques allows addressing the limitations of other non-informed techniques (e.g., Random Testing), which often are limited by the size and complexity of the problem.

According to McMinn [33], *"Metaheuristic search techniques are high-level frameworks which utilize heuristics in order to find solutions to combinatorial problems at a reasonable computational cost."*. A vital aspect of these techniques is that they are problem independent, which means that they can be promptly adapted to specific problems. In test data generation, these techniques' search process is guided by a fitness function. *"Fitness function is a mathematical function that assigns a value to each solution in the search space to evaluate its quality"* [44].

The application of metaheuristic search techniques to test case generation comprises several areas. The most common area is *"the coverage of specific program structures, as part of a structural, or white-box testing strategy"* [33] since that, in test data generation, we want to minimize the number of test cases generated while maximizing coverage in the minimum time.

The most popular metaheuristic search algorithms are Hill Climbing, Simulated Annealing, Genetic Algorithms, Particle Swarm Optimization, Ant Colony Optimization, Tabu Search, Cuckoo Search, among others [44]. Despite these algorithms being well-known as well as their benefits, their success still varies according to the problem to be solved. Therefore, the limitations of SBT are bounded to the limitations of the algorithm used.

2.4 Coverage-Guided Fuzzing

Fuzzing is a widely effective and efficient state of the art technique used in vulnerability detection. Fuzzers work at the test case generation level, and they usually use two types of test case generation strategies: grammar-based or mutation-based [16]. The former achieves very appellative results, but building an input grammar from scratch requires substantial work. Due to not depending on an input grammar, the latter is more simple and scalable and, therefore, more widely used. However, due to the randomness involved, such solution does not present a consistent and satisfactory program's coverage. This poor coverage results led to the emergence of the coverage-guided fuzzing (CGF) concept.

CGF is one of the most popular fuzzing techniques and differentiates itself from traditional fuzzing [49] in the sense that, instead of blindly generating and mutating inputs hoping that a random input triggers a program vulnerability, it leverages the feedback from the program in the form of code coverage to generate and mutate inputs that, by exercising new program paths, have a higher probability of exhibiting a new program behavior.

American Fuzzy Lop (AFL), libFuzzer, and VUzzer are currently some state of the art coverage-guided fuzzers. Despite sharing similarities in terms of striving to maximize code coverage, the measures of code coverage used by these fuzzers are different. Currently, there are two types of measures used. One *"is to count execution basic blocks (BBLs)"* (e.g., VUzzer); another *"is to use the transition between two BBLs, which is also known as the edge coverage."* (e.g., AFL) [53]. Regardless of the fuzzers' internal architectural and design decisions, the typical workflow of standard coverage-guided fuzzers is relatively straightforward, as depicted in Figure 2.1.

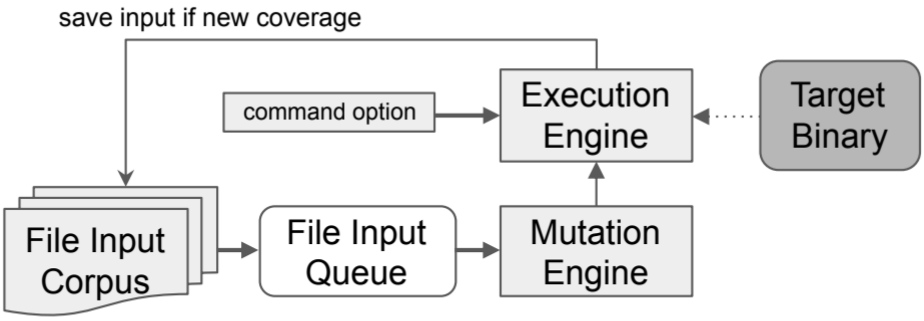


Figure 2.1: The workflow of standard coverage-guided fuzzers [49].

As represented in Figure 2.1, the workflow of a standard coverage-guided fuzzer consists on a cycle composed of five steps: *(i)* inserts the so-called seeds into the *File Input Queue*, *(ii)* selects one input from the *File Input Queue*, *(iii)* mutates the selected input, *(iv)* runs the target program with the mutated input, and *(v)* saves or discards the mutated input depending on whether it increases code coverage or not, respectively. According to Gan et al. [16] studies *"have shown that improvements to each step of this loop could promote the efficiency and effectiveness of fuzzers"*.

2.5 Concolic Testing

Over the years, coverage has been one of the most popular forms of feedback to indicate how well the SUT behaviors are being exercised. Random Testing and Symbolic Execution are just some of the proposed techniques used to automatically generate input values to improve the variety of behaviors exercised. Despite the intrinsic benefits of these methods, there are also some limitations [47]. One of Random Testing's significant limitations is the absence of control on the input generation, which often generates inputs that exercise the same program behaviors, hence being redundant. On the other hand, and even though Symbolic Execution addresses this problem of redundancy, it has the limitation of not scaling well *"for large software due to the limitations of underlying theorem provers and symbolic analyzers"* [46] due to the *path explosion problem*. Therefore, to address these techniques' limitations, Concolic Testing was proposed.

Concolic Testing [46], also known as Concolic Execution, is a popular and promising software testing technique that combines Concrete Execution and Symbolic Execution. Its main goal consists of achieving the highest code coverage values in the minimum time possible, assuming that it leads to higher fault detection rates.

Due to the fact that Concolic Testing uses Symbolic Execution and that real-world programs often exhibit an infinite variety of different paths, the *path explosion problem* is still a significant concern in Concolic Testing. Nevertheless, Concolic Testing addresses this problem by including search heuristics [5], a key component, in its process. The use of search heuristic allows Concolic Testing to *"preferentially explore particular classes of execution paths that they think are most effective to maximize code coverage within a given time limit."* [5].

Concolic Testing generates test cases through a cyclic and systematic process [5] composed of three steps: *(i) "it concolically executes the subject program to collect the path condition"*, *(ii) "it produces a new path condition by selecting and negating a branch of the current path condition"*, and *(iii) "it solves the resulting path condition to generate a new test-case that guides the next program execution towards the opposite of the selected branch."*. By being systematic, this process enables Concolic Testing to be used in numerous domains.

2.6 Hybrid Fuzzing

Hybrid Fuzzing [55, 51, 38] is a novel hybrid approach that combines fuzzing and concolic execution and aims to take advantage of both worlds. These techniques' benefits, limitations, and processes are described in Section 2.4 and Section 2.5, respectively.

Hybrid fuzzers, during runtime, take advantage of the combination of such techniques by continuously keep changing between them, depending on the complexity faced. Therefore, for more complex branches, concolic execution is used; otherwise, fuzzing, known for quickly exploring trivial input spaces, is used.

QSYM [55], Driller [51], and DigFuzz [56] are some of the current state of the art hybrid fuzzers. Even though each one of these fuzzers uses different implementation methods of applying

Hybrid Fuzzing [34], these fuzzers share the same goal: achieving more code coverage and finding more bugs in the minimum time possible.

Despite having proven results of achieving higher code coverage, and consequently finding more defects [52], Hybrid Fuzzing solutions' effectiveness is still bounded to the subsequent techniques' performance, namely to the concolic executor used.

Chapter 3

State of the Art

Throughout this chapter, we describe the state of the art of PBT search strategies. Section 3.1 presents the Systematic Literature Review conducted to assess the state of the art of PBT search strategies. Section 3.1.1 describes the defined Methodology, including the defined Research Questions and Databases used for the publications' retrieval. Section 3.1.2 details the Study Selection Criteria used for driving the Study Selection Process. Section 3.1.3 categorizes the selected publications while Section 3.1.4 descriptively synthesizes them. Section 3.1.5 terminates the Systematic Literature Review by presenting an analysis of the results' evolution and their contribution and the answers to the defined research questions. Finally, Section 3.2 contains a summary of findings of PBT search strategies' state of the art.

3.1 Systematic Literature Review

A Systematic Literature Review (SLR) was conducted to review the state of the art of PBT search strategies. The purpose of this review is to summarize the current evidence and evaluate its quality, along with the identification of any gaps in the current research for further investigation.

3.1.1 Methodology

This SLR followed the guidelines for performing Systematic Literature Reviews in Software Engineering [24]. Therefore, we started by defining the research questions and framing the research scope through the **PICOC** framework. Then, after selecting the data sources and defining the studies' selection criteria, we selected and assessed every publication resultant from the executed search.

3.1.1.1 Research Questions

In any SLR, the research questions' definition is critical to assess the state of the art and reveal pending research challenges of a particular subject. Therefore, we outline the following research questions to identify the latest practice, research, and studies related to PBT search strategies.

SRQ1 *Which are the most advanced PBT search strategies?*

SRQ2 *How can the current PBT search strategies be improved?*

SRQ3 *What are the most promising alternatives to the current adopted PBT search strategies?*

By answering these questions, we enable the practitioners to understand current PBT search strategies, their effectiveness, and how they can be improved. Regarding the researchers, we provide them valuable insights by stating the current challenges and issues that can be deeply explored.

Along with the research questions' specification, we structured the outlined questions by applying the PICOC (Population, Intervention, Comparison, Outcomes, and Context) framework to establish the research scope focus, as shown in Table 3.1.

Table 3.1: Research Questions structured by PICOC criteria.

PICOC	**Description**
Population	PBT search strategies.
Intervention	Metaheuristic search strategies.
Comparison	Current PBT adopted strategies.
Outcomes	Optimization of PBT search strategies.
Context	Focus on empirical studies within the domain of PBT and similar ones.

3.1.1.2 Databases

The publications obtained for this literature research were retrieved from the following digital libraries: (a) ACM Digital library, (b) Scopus, and (c) IEEExplore.

These three electronic sources were selected due to containing the most relevant digital literature for Computer Science studies, thus being the most relevant and reliable sources for Software Engineering.

3.1.2 Study Selection

The current chapter's section outlines the process of selecting the publication's representative of the state of the art, which enables us to answer the previously presented research questions. Therefore, we start by describing our study selection criteria, followed by describing the whole selection process itself and how both are correlated.

3.1.2.1 Study Selection Criteria

Our SLR followed the inclusion and exclusion criteria detailed in Table 3.2. This decision was made during the review's protocol definition to reduce the possibility of bias and to identify publications that provide direct evidence about the previously stated research questions.

Table 3.2: Inclusion and Exclusion Study Selection Criteria.

I/E	Identifier	Criterion
Exclusion	EC1	Publications not written in English.
	EC2	Publications outside the Computer's Science domain.
	EC3	Any other format than Portable Document Format (PDF).
	EC4	Duplicated publications.
	EC5	Publications that present just theoretical hypothesis (e.g., interviews, talks, lecture notes, etc).
Inclusion	IC1	Publications matching the query previously mentioned and within the scope of our analysis.
	IC2	Publications must be either Conferences, Journals or Articles.
	IC3	Publications ranging from 2013 to 2021.
	IC4	Publications that present promising and well-explained techniques, their contributions, challenges, and limitations.

3.1.2.2 Study Selection Process

The study selection criteria drove the entire study selection process of our SLR, performed in December 2020 and described in Figure 3.1. We initialized this process by searching in the aforementioned digital libraries using a query representative of the most probable keywords to be present in our target studies, namely `property-based testing` and `search-based testing`. Eventually, and after a preliminary analysis of the results, this query evolved to a more complex and fine-tuned one comprising several other keywords. In the end, we came up with the following query:

```
"Property-Based Testing" OR "Structure-Aware Fuzzing" OR
"Coverage-Guided Fuzzing" OR "Search-Based Testing" OR "Concolic
Execution" OR "Hybrid Fuzzing"
```

The process presented in Figure 3.1 comprises, in total, a sequence of 6 steps. Each step has its own definition and purpose, which are described below:

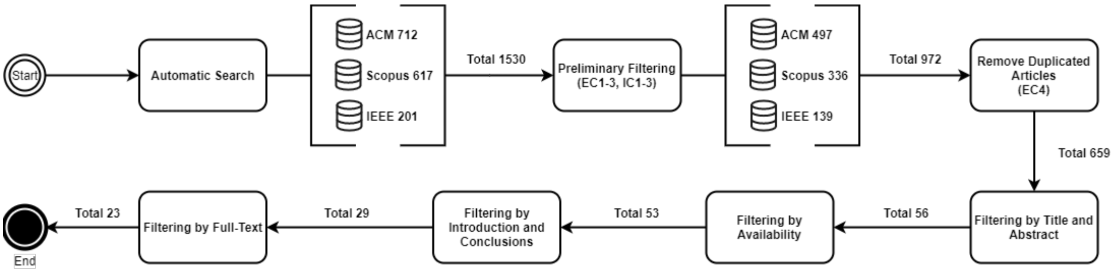

Start
Automatic Search
ACM 712
Scopus 617
IEEE 201
Total 1530
Preliminary Filtering (EC1-3, IC1-3)
ACM 497
Scopus 336
IEEE 139
Total 972
Remove Duplicated Articles (EC4)
Total 659
Filtering by Title and Abstract
Total 56
Filtering by Availability
Total 53
Filtering by Introduction and Conclusions
Total 29
Filtering by Full-Text
Total 23
End

Figure 3.1: Overview of the SLR Study Selection Process.

1. **Automatic Search**: Insertion of the elaborated search query in the aforementioned digital libraries without any type of filtering.

2. **Preliminary Filtering**: Before merging the results from the different sources into a single dataset, the initial results were filtered in order to exclude any publication not written in English (EC1), outside the Computer's Science domain (EC2), and in any other format than PDF (EC3). It was also only considered to be included all the publications that matched the search query, and that fell within the current dissertation's scope (IC1), from either Conferences, Journals, or Articles (IC2), ranging from 2013 to 2021 (IC3). The reason behind the considered time period is further explained in section 3.1.5.1.

3. **Remove Duplicated Articles**: Once all the results were merged into a single dataset, we then proceeded to remove duplicated publications (EC4).

4. **Filtering by Title and Abstract**: The selected papers were revised according to their *Title* and *Abstract*. For this particular step and the following steps, it was only considered publications that present promising and well-explained solutions, their contributions, challenges, and limitations (IC4), excluding all publications that just present theoretical hypothesis (EC5).

5. **Filtering by Introduction and Conclusions**: Potential papers considered in the previous step were then further analyzed regarding their *Introduction* and *Conclusions* section's content.

6. **Filtering by Full-Text**: Each publication was deeply analyzed at this point in the process to ensure that its content was, without any uncertainty, within the current dissertation's scope.

Initially, from a total of 659 publications and through a systematic and well-defined protocol, we selected 23 publications that we believe are the ones that compose the current dissertation's state of the art.

3.1.3 Data Categorization

The selected publications differ in many aspects, including their domain and proposed solution. Therefore, in this particular section, we categorize all 23 selected publications according to their domain. In total, we have identified five different domains.

An overview of each domain and associated publications is presented in Table 3.3, while their analysis is presented later in this publication, in section 3.1.5.

Table 3.3: Data Categorization according to the publication's domain.

Domain	Publication Title	Reference
Search-Based Testing	PSO based test case generation for critical path using improved combined fitness function	[44]
	Test Case Generation Based on Search-Based Testing	[45]
	Seeding strategies in search-based unit test generation	[43]
Property-Based Testing	Targeted Property-based testing	[28]
	Automating Targeted Property-Based Testing	[30]
	JQF: Coverage-guided property-based testing in Java	[36]
	Semantic fuzzing with ZEST	[37]
	Coverage guided, property based testing	[25]
	Quickly generating diverse valid test inputs with reinforcement learning	[42]
Coverage-Guided Fuzzing	PerfFuzz: Automatically generating pathological inputs	[26]
	CollAFL: Path Sensitive Fuzzing	[16]
	PathAFL: Path-Coverage Assisted Fuzzing	[53]
	CrFuzz: Fuzzing multi-purpose programs through input validation	[49]
	SLF: Fuzzing without Valid Seed Inputs	[54]
	Ankou: Guiding Grey-Box Fuzzing towards Combinatorial Difference	[32]
Concolic Testing	Enhancing Test Cases Generated by Concolic Testing	[12]
	Concolic Testing with Adaptively Changing Search Heuristics	[5]
Hybrid Fuzzing	Improving function coverage with munch: A hybrid fuzzing and directed symbolic execution approach	[35]
	QSYM: A practical concolic execution engine tailored for hybrid fuzzing	[55]
	EPfuzzer: Improving hybrid fuzzing with hardest-to-reach branch prioritization	[52]
	SHFuzz: Selective hybrid fuzzing with branch scheduling based on binary instrumentation	[34]
	Intriguer: Field-level constraint solving for hybrid fuzzing	[8]
	Sequence Directed Hybrid Fuzzing	[27]

3.1.4 Data Synthesis

As mentioned in the last section, the considered publications differ in many aspects, including their domain and proposed solution. While some papers propose a completely different and innovative solution, others introduce an enhancement over other proposed solutions.

Regardless of the publication's domain and proposed solution, the considered publications were selected because we believe that they are representative of the state of the art of this dissertation's scope.

In this particular section, instead of just presenting the selected publications in chronological or domain-based order, we decided to present them in a unique way that allows the reader to understand the publications' evolution and how they correlate with each other. Therefore, each publication is synthetically described in the following subsection.

3.1.4.1 Descriptive Synthesis

Löscher and Sagonas [28] introduce the concept of *targeted property-based testing*, an enhanced form of property-based testing that aims to replace the utterly random input generation by a guided strategy. In other words, it strives to combine the advantages of PBT with SBT. Hence, the authors developed the Target framework, a concrete implementation of targeted property-based testing, which is currently fully integrated into the PBT tool, PropEr [39]. This framework's motivation derives from the need to explore the input space more efficiently in order to increase confidence over the SUT. The framework achieves this by guiding the input generation with black-box search techniques that drive the input generation process through an algorithm responsible for trying to falsify the specified properties. Target comes with the implementation of **Hill Climbing** and **Simulated Annealing** as built-in search strategies. Both search strategies present better results than the purely random input generation approach. By better results, we mean fewer tests required to find defects and falsify the defined properties. However, these strategies' short-comings are directly related to themselves (e.g., local optima in Hill Climbing). The authors also mention that the framework is extensible to other algorithms, like Adaptive Random Testing (ART) [7], which keeps continuously evolving. However, ART focuses entirely on the input domain without taking into account feedback from the test execution.

Löscher and Sagonas [30] aim to enhance targeted property-based testing by introducing automated targeted property-based testing. This new concept is similar to what the authors have defined in their previous work [28]. The difference relies on the introduction of automation into the search strategies. In their previous work, the authors, despite the positive results, stated that the need for manually configuring each search strategy could compromise the adoption of targeted property-based testing. Therefore, they now introduce the concept of automation by focusing on automating the construction of the neighborhood function of Simulated Annealing, which is the default search strategy used by PropEr. This function is responsible for producing the next random input similar to its predecessor, and it is one of the configurations needed. With the introduction of automatic generation of the neighborhood function, the authors reduce the effort to adopt targeted property-based testing. However, and despite this automatic generation performing reasonably well, the generated neighborhood function still is a general one, which means that it is not as good as a carefully hand-written one.

Sahoo and Ray [44] support the fact that test case generation is a critical point in the testing process and that coverage criterion (e.g., statement coverage, branch coverage, path coverage, etc) is frequently used to conduct such activity. However, the authors endorse that focusing on the maximum percentage of coverage is not ideal since the critical path may be skipped. Therefore, the authors' focus is to cover the critical path, which they define as the path with a lower probability of being covered. In that sense, they propose an improved combined fitness function responsible for guiding a metaheuristic algorithm's search process. The proposed fitness function can be used with the Adaptive Particle Swarm Optimization (APSO) and Particle Swarm Optimization (PSO) metaheuristic algorithms. Compared with other metaheuristic algorithms and fitness functions, the authors' proposed solution achieves better results since it covers the critical path more efficiently. Also, most of the paths are covered by the time the critical one is covered.

Sahoo et al. [45] suggest a new and better alternative to how search-based testing can increase confidence over the testing process in terms of test case generation. The authors' main goal is to reduce the amount of time required to generate new test cases and maximize the test cases' path coverage. Thus, they propose a new approach using a metaheuristic search based algorithm, the cuckoo search algorithm, to generate test cases for path coverage. The authors' solution key point is to adapt the traditional cuckoo search algorithm to cover the critical path, which is the path with the lower probability of being covered. To assess the proposed solution's potential, the authors compared it with their previous work [44], which follows a similar approach and has the same goals. According to the authors, the cuckoo search achieves better results than the PSO based test case generation solution in terms of iterations needed to achieve coverage of the critical path. However, in terms of execution time, the proposed solution performs worse and requires more time than the alternatives. Nevertheless, the cuckoo search algorithm's implementation is more straightforward than PSO based one and needs fewer parameters. Therefore, the proposed solution is more appellative and easier to adopt.

Padhye et al. [36] present JQF[1], a platform for performing CGF, a modern technique for automatic test-input generation, which has gained particular attention in the last years due to the exponential increase in computation power. Instead of proposing a unique solution, JQF allows the implementation of multiple CGF based solutions. JQF designates these solutions as *Guidances*. At the time of this writing, JQF supports five different *Guidances*: *No Guidance*, *Zest Guidance*, *AFL Guidance*, *PerfFuzz Guidance*, and *Repro Guidance*. Each one of the listed solutions has its own advantages and drawbacks. The capability of being extensible to adopt new *Guidances*

[1] Available: https://github.com/rohanpadhye/jqf

together with the already supported *Guidances* is what composes JQF 's true potential.

Padhye et al. [37] propose Zest, a new technique for automatically guiding PBT frameworks' input generators to maximize code coverage in programs' semantic analysis stages. Zest is an alternative to the popular random input generation method. This new technique's novelty comes from driving the input generation by incorporating the SUT's feedback in the form of semantic validity. The proposed solution takes advantage of the popular concept of CGF. CGF is highly effective in finding defects while using code coverage as its criteria. However, most of the bugs found by this popular method are at the syntactical level. Zest readjust CGF to work at the semantic level by modifying how mutations are applied in each iteration. The authors evaluate Zest against QuickCheck [9] and AFL[2] tool, which is the most popular tool in the CGF domain. According to their evaluation, Zest outperforms both QuickCheck and AFL in terms of time required to find defects and in the number of bugs found. Hence, Zest can be considered a more reliable and effective approach.

Lemieux et al. [26] propose PerfFuzz, a method that automatically generates inputs via *feedback-directed mutational fuzzing*. PerfFuzz is a multi-objective coverage-guided approach, which combines total path length and edge execution count to generate pathological inputs without any domain knowledge. This method differentiates from the others so that inputs that exercise specific program locations more than other inputs are prioritized for posterior mutation. This enables identifying different program hot spots and finding inputs exercising the worst program behavior. PerfFuzz is built on top of AFL and inherits most of its implementation details. The authors evaluate PerfFuzz by comparing it with SlowFuzz [40], a similar and single-objective method, and with AFL itself. According to the authors, PerfFuzz outperforms both alternatives in terms of maximum path length, hot spots found, and execution counts across the control-flow graph (CFG) edges. Therefore, the results strengthen the potential of PerfFuzz and its multi-objective approach.

Ognawala et al. [35] introduce Munch[3], a new *hybrid fuzzing* framework, with the sole purpose of increasing coverage at all depths of C-programs. The authors consider coverage only at the component level, which means that a function is considered covered from the moment it is called, despite how its inner paths are exercised. The introduced framework combines symbolic execution with fuzzing to increase function coverage. Munch uses KLEE [4] for the former and AFL for the latter. The key idea is that combining both techniques enables each technique to address the other one's drawbacks. In that sense, Munch supports two different modes of operation: FS (Fuzzing with Symbolic Execution) and SF (Symbolic Execution with Fuzzing). This feature differentiates Munch from the rest of the hybrid fuzzing frameworks because it allows programs to choose which execution mode better suits their needs. Additionally, to the apparent different execution order of both modes, one significant difference is that FS needs to be provided with seed

[2] Available: https://github.com/google/AFL

[3] Available: https://github.com/tum-i4/munch

inputs while SF does not. The authors evaluated Munch by comparing it to KLEE and AFL alone on both modes. According to the results, Munch performs better than the alternatives since its results in terms of coverage and time required to cover deeper program paths were always equal or better than KLEE or AFL alone.

Yun et al. [55] propose QSYM[4], a practical concolic execution engine tailored for hybrid fuzzing. QSYM tackles and overcomes the state of the art fuzzers' performance bottlenecks such as Driller [51], which still suffers from scaling to find real defects in non-trivial, real-world applications. *Slow Symbolic Emulation* [55, p. 3], *Ineffective Snapshot* [55, p. 5], and *Slow and Inflexible Sound Analysis* [55, p. 5] are the targeted performance bottlenecks that drove QSYM architectural design and implementation decisions. From these decisions, *Fast Symbolic Emulation* [55, p. 11], *Optimistic Solving* [55, p. 8], and *Pruning Basic Block* [55, p. 8] are the ones that enable QSYM to achieve better performance and code coverage than other state of the art alternatives. QSYM was implemented from scratch and combined with AFL so that hybrid fuzzing could be integrated into the concolic execution engine. According to the authors' evaluation, QSYM scales well to real-world programs and achieves higher code coverage more efficiently than AFL, Driller, and VUzzer [41]. Additionally to the positive results, one key point of QSYM's implementation is that it can be used beyond fuzzing or even with other fuzzers. However, QSYM still is a concolic executor, so its performance is still bound to theoretical limits like constraint solving.

Wang et al. [52] propose EPfuzzer, a tool that implements a hardest-to-reach branch prioritization strategy to improve hybrid fuzzing. This strategy involves selecting the hardest-to-reach branches for the fuzzer for concolic execution instead of solving all branches blindly. The proposed tool aims to tackle and overcome QSYM limitations, and as such, it can be seen as an enhancement over QSYM. Besides overcoming QSYM weaknesses, EPfuzzer's motivation came from the necessity of tackling the following concerns: *"When should the concolic executor be launched?"* [52, p. 4], *"What paths should be solved by concolic execution?"* [52, p. 5], and *"How to perform symbolic emulation and constraint solving efficiently?"* [52, p. 7]. Hence, EPfuzzer was architecturally designed to address these problems and has the following key components: *Hardest-to-reach Branch Prioritization* [52, p. 9], *Relevant Input Bytes Identification* [52, p. 12], and *Target Guided Concolic Execution* [52, p. 13]. To assess EPfuzzer potential, the authors evaluated it against AFL and QSYM. According to the authors' evaluation, EPfuzzer outperforms both alternatives since it achieves more code coverage and finds more bugs in a shorter period of time and with fewer inputs.

Gan et al. [16] endorse that the state of the art CGF solutions are limited by coverage inaccuracy, which hinders their performance. Additionally, coverage inaccuracy also includes compromising seed selection policies. Thus, the authors propose CollAFL, a path sensitive fuzzing approach that accurately measures coverage and shows its potential. CollAFL is built on top of AFL

[4] Available: https://github.com/sslab-gatech/qsym

and aims to solve AFL's most significant problem: the hash collision problem[5]. Besides solving this problem completely and ensuring that each edge in the CFG has a unique hash, the proposed solution suggests three new seed selection policies: *untouched-neighbor-branch guided* [16, p. 7], *untouched-neighbor-descendant guided* [16, p. 8], and *memory-access guided* [16, p. 8]. According to the authors, the proposed solution performs better than AFL in terms of covering more code and finding more bugs. All the proposed seed selection policies achieve higher code coverage results than those adopted by AFL. The authors also verified that even in the worst-case scenario, CollAFL could outperform AFL in terms of path discovery.

Yan et al. [53] propose PathAFL[6], a path coverage-assisted fuzzing approach. Path coverage is proven to be the most accurate way of measuring code coverage. However, in the real world, it is not practical due to the path explosion problem. To address this problem, the proposed solution designed a path filtering algorithm that assesses each program's path and evaluates if it is worthy of being added to the seed queue or not. There is an associated weight for each path, which determines the path's priority of being further selected (a bigger weight implies a higher priority). Despite this algorithm's efficiency in reducing the total number of paths, it incurs additional performance overhead. However, the authors found out that it is possible to trace critical paths with very little overhead. Therefore, PathAFL makes a trade-off between path coverage granularity and fuzzing performance. PathAFL is built on top of AFL and CollAFL. It combines both approaches' features, especially the one presented in CollAFL that is responsible for solving the AFL's hash collision problem. According to the authors' evaluation, the proposed solution outperforms both AFL and CollAFL by achieving higher code coverage values and finding more unique defects in a shorter period of time.

Song et al. [49] introduce CrFuzz, a fuzzing technique directed to programs that take both file input and command-line options as their inputs. The authors denominate these types of programs as multi-purpose programs. CrFuzz development was motivated by the lack of capability of other fuzzers to fuzz multi-purpose programs, more specific programs whose inputs can take several command-line options. Thus, the proposed technique addresses this concern in order to allow the efficient exploration of all these programs' possible behaviors. The most significant aspect of CrFuzz is its validity checker that, through machine learning, determines whether a particular input is valid or not. The key idea behind the validity checker component is that it enables saving fuzzing time. This is possible because by knowing the valid and invalid inputs, we also know when to fuzz. Using multi-purpose programs, the authors evaluated CrFuzz against AFL, QSYM, and MOpt [29] and verified that the proposed solution covers more paths than the alternatives in the same period of time. The authors also assessed the validity checker's accuracy results, and despite not being entirely accurate, its results are promising and adequate to be used in fuzzing activities.

[5] https://lcamtuf.coredump.cx/afl/technical_details.txt

[6] Available: https://github.com/yanxxd/PathAFL

Mi et al. [34] propose SHFuzz, a Selective Hybrid Fuzzing (SHF) solution, which aims to enhance hybrid fuzzing performance. The proposed solution implements synchronization as its form of applying hybrid fuzzing. This type of implementation consists of running fuzzing and concolic execution concurrently, and it is the same as the one implemented by QSYM. Hybrid fuzzing with synchronization has already been proven to be very efficient, as shown by QSYM results. However, it still has some unsolved problems like (1) the problem of having too many candidate inputs for concolic execution, (2) the problem of having duplicated inputs leading to the same coverage, and (3) the problem of having several branches with different complexities. SHFuzz proposes the critical branch selection [34, p. 9] and priority score calculation [34, p. 11] algorithms for solving these problems in order to enhance hybrid fuzzing performance. The authors evaluated SHFuzz against QSYM to assess its potential. According to their evaluation, SHFuzz covered more paths and found more crashes than QSYM in the same period of time. A critical aspect of SHFuzz is that it does not try to replace either QSYM or even AFL. Instead, it aims to combine both to achieve better results. That said, the proposed solution can be used in conjunction with other state of the art fuzzers.

You et al. [54] introduce SLF, a Seedless Fuzzer that performs fuzzing without the need for specifying valid seed inputs. As a matter of fact, such capability is what differentiates SLF from other fuzzers. SLF's goal is the same as other fuzzers: achieving the highest code coverage possible in the shortest period of time, which leads to finding more bugs efficiently. The proposed solution is built on top of AFL, and despite its complex implementation, its design is relatively straightforward. SLF starts with an elementary and short input (4 bytes), which most likely fails the input validation process. Then that same input is adequately mutated until a more complex and valid one is produced. The process repeats itself, and the valid inputs generated are fed into the input queue. According to the authors' evaluation, SLF achieves overall better coverage and performance than AFL, AFLFast [3], and Driller. Considering that seed selection is one of the fuzzing concerns, the results reinforce SLF potential.

Cho et al. [8] propose Intriguer[7], a hybrid fuzzer designed to improve hybrid fuzzing performance. QSYM, along with its problems, was the tool that motivated the development of Intriguer. From QSYM's problems, the authors highlight the following ones: *slow symbolic emulation* [8, p. 2], *unnecessary constraints dominate solving* [8, p. 2], *resources overly allocated* [8, p. 3], and *missing hard-to-trigger bugs* [8, p. 3]. Intriguer's design was carefully conceived to address all of these problems. Its core idea is the optimization of symbolic execution through *field-level constraint solving* [8, p. 7]. LAVA-M [11] dataset was the one used to evaluate Intrigues against QSYM, VUzzer, and AFL. According to the authors' results, Intriguer outperforms its alternatives since it detects more defects and covers more paths in less time.

[7] Available: https://github.com/seclab-yonsei/intriguer

Manès et al. [32] state that the current widely used code coverage strategy by coverage-guided fuzzers has some limitations, mainly regarding the loss of information. This loss of information results from the fact that most fuzzers, regardless of their implementation, prioritize test cases that achieve new code coverage. However, higher code coverage values do not translate to a higher number of bugs found. In fact, the authors endorse the idea that bugs are usually triggered when a specific execution path is exercised and not when a particular code section is covered. To address this problem, the authors propose Ankou[8], a fuzzing technique that the authors refer to as *distance-based fuzzing*. Solving the previously mentioned problem includes tackling the following challenges: Ankou's fitness function should be informative and computationally fast, and can not accept to many seeds in its seed pool in order to handle such seeds practically. To address these challenges, Ankou leverages from a distance-based fitness function [32, p. 3], a dynamic Principal Component Analysis (PCA) [32, p. 4], and an adaptive seed pool update [32, p. 6] concept, which compose Ankou's core components. Ankou is built on top of AFL, and it only includes its most significant and required features. The authors evaluate Ankou against AFL and Angora [6]. Their evaluation shows that all three fuzzers achieve similar code coverage values. However, Ankou outperforms the other fuzzers in terms of finding more unique bugs and also in the time required to find such bugs. Hence, the results contribute to reinforce the authors' core idea that "software bugs often manifest when we exercise a particular execution path, but not when we reach a node" [32, p. 10].

Rojas et al. [43] address the impact of seeding strategies in search-based techniques. The authors state that seeding is a critical phase in the process of generating test cases. Therefore they propose four different seeding strategies: *seeding of constants extracted from source code or bytecode* [43, p. 4], *dynamic seeding* [43, p. 5] (seeding numerical values and strings), *type seeding* [43, p. 7], and *seeding by incorporating previous solutions* [43, p. 8]. These strategies were implemented in the EVOSUITE [15] tool and were evaluated according to their impact on branch coverage. Regardless of the strategy, the results show that each one led to higher values of coverage. However, the increase in coverage was different from project to project. This was expected since it is correlated with the fact that each strategy works differently. Thus each strategy is more suited for certain projects.

Dutta et al. [12] introduce a new technique, SCOTP, which stands for Strengthening COncolic Test using Pair-Wise testing. Pair-wise testing enables the enhancement of concolic testing since it allows extending the concolic tester's test suite with additional test cases uncovered initially due to the concolic tester's limitations. SCOTP aims to increase both test coverage and mutation score of concolic testing. The proposed approach was evaluated using two distinct modes. In *Mode-I*, the authors evaluated the original test suite using coverage criteria and mutation ratio. In *Mode-II*, they improved the test suite using pair-wise testing and measured their effectiveness against the first mode. According to the authors' results, *Mode-II* presents better results than the alternative in

[8] Available: https://github.com/SoftSec-KAIST/Ankou

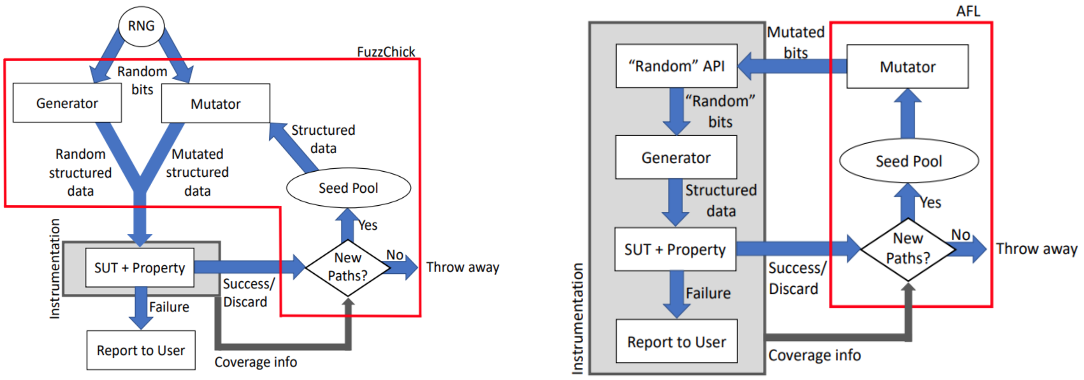
terms of achieving higher decision coverage, Modified Condition / Decision Coverage (MC/DC), and mutation ratio, which reinforces pair-wise testing potential.

Cha and Oh [5] propose Chameleon[9], a concolic testing tool that adaptively changes its search heuristics in order to maximize the final performance. Concolic testing is similar to Symbolic Execution in a way that it also faces the *path explosion problem.* Traditional concolic testing tools address this problem by using search heuristics responsible for selecting paths that maximize code coverage. Hence, search heuristics play a vital role in concolic testing. Instead of proposing a new and more efficient search heuristic, Chameleon proposes an algorithm that automatically learns and switches search heuristics during the concolic testing process. To achieve this, the proposed tool keeps a set of search heuristics and dynamically alternates between them. Therefore, this solution allows the tool to maximize performance and be more efficient because, at each step of the concolic testing process, the tool uses the most suited search heuristic. Chameleon is built on top of Crest and was evaluated on several projects against other popular search heuristics over 24 hours. According to the results, the proposed tool achieves higher branch coverage values and finds more bugs than the alternatives. The authors also evaluated Chameleon's learning algorithm. For this purpose, they compared it with a naive algorithm that randomly alternates between search heuristics. The results show that the Chameleon's algorithm performs way better and covers much more branches than the naive one. Despite the overall positive results, Chameleon must be manually tuned, which means that its success is program dependent.

Liang et al. [27] introduce sequence directed hybrid fuzzing (SDHF) technique. This technique can be seen as a combination of directed grey-box fuzzing with hybrid fuzzing because, on one hand, it guides the fuzzing and concolic execution processes with the program's enhanced target statement sequences, and on the other hand, it combines both of these processes. By combining directed grey-box fuzzing with hybrid fuzzing, the proposed technique aims to make use of the best of both worlds while enhancing the overall process. Thus, the authors propose a customized concolic execution method, an energy scheduling algorithm, and a seed priority mechanism. SDHF was implemented in a tool called Berry, and according to the results, Berry performed 1.28x faster than the popular hybrid fuzzer, QSYM. Berry was also able to reproduce more crashes than QSYM, which reinforces the authors' proposed technique potential.

Lampropoulos et al. [25] propose coverage-guided, property-based testing (CGPT) technique that combines PBT with CGF. This proposal's motivation was PBT's incapacity of generating inputs capable of satisfying sparse conditions. This is due to PBT's reliance on a pure test case random generator. Hence, the authors propose fuzzing, more specifically CGF, as an alternative to enhance PBT. This way, test cases are generated according to the program's feedback (coverage) instead of being randomly generated. The authors' proposed technique is implemented in FuzzChick, an extension of the property-based random tester for Coq, QuickChick. FuzzChick was

[9] Available: https://github.com/kupl/Chameleon



(a) FuzzChick Workflow

(b) Crowbar Workflow

Figure 3.2: FuzzChick vs Crowbar workflows, highlighting how test cases are generated and consequently accepted or discarded. [25]

evaluated against QuickChick and an adaption of Crowbar[10]. Crowbar is similar to FuzzChick in a way that it also integrates CGF with PBT. The main difference between them is at the mutation level because while Crowbar uses its fuzzer to fuzz the source of the PBT framework's randomness, FuzzChick uses it to fuzz the inputs directly. Both tools' workflows are depicted in Figure 3.2. According to the authors' evaluation, apart from QuickChick with carefully handcrafted inputs, FuzzChick performs better than the other alternatives having a lower Mean Time to Failure (MTTF).

Reddy et al. [42] propose RLCheck[11], a novel method that quickly generates several valid inputs through reinforcement learning. RLCheck's is a black-box approach that aims to improve PBT while addressing the drawbacks of other alternatives such as white-box and grey-box techniques, that hinder the test cases' generation process performance due to their complexity. By improving PBT, the authors mean generating several valid inputs in the shortest time possible. To achieve this, the authors faced the problem that they have designated as the *diversifying guidance problem*, a problem similar to the ones solved by reinforcement learning. Therefore, the RLCheck approach is based on reinforcement learning, which adapts its test cases' selections on-the-fly during testing. The authors evaluate RLCheck against other tools like QuickCheck and Zest in several metrics: *Generating Diverse Valid Inputs* [42, p. 7], *Covering Different Valid Behaviors* [42, p. 8], and *Greybox Information* [42, p. 9]. According to the results, the number of diverse valid inputs generated by RLCheck is far superior to the other approaches. However, the authors found out that no method achieves the highest branch coverage on all used benchmarks. In fact, *"RLCheck's plateauing branch coverage suggests that it may be learning to generate diverse inputs with similar features rather than discovering new behavior"* [42, p. 9]. Additionally, the authors also evaluated that adding grey box feedback to RLCheck hinders its performance and has no significant gains, reinforcing the idea that RLCheck performs better as a black-box technique.

[10] Available: https://github.com/stedolan/crowbar
[11] Available: https://github.com/sameerreddy13/rlcheck

3.1.5 Data Analysis

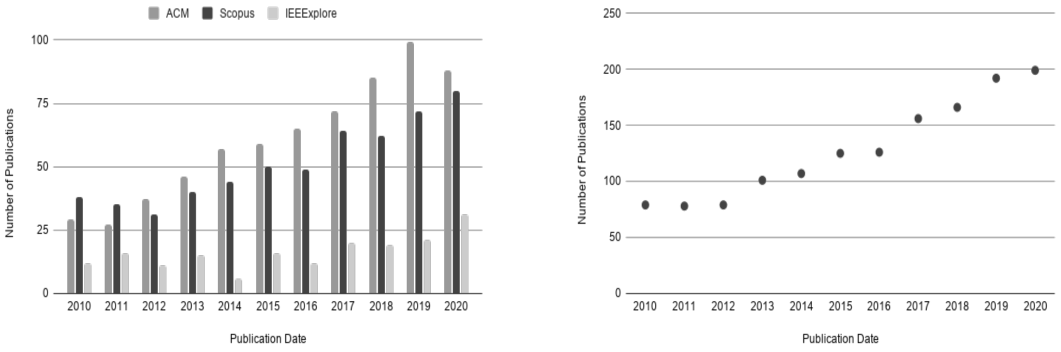
In this section, we analyze the extracted data from the literature according to its categorization and synthesis made in sections 3.1.3 and 3.1.4, respectively. Therefore we start by analyzing how the data evolved over the years, followed by the analysis on how the selected publications contribute to the problem we are trying to solve. Finally, we revisit the research questions defined in section 3.1.1.1 and answer them according to our findings during the SLR process.

3.1.5.1 Evolution

An essential aspect while assessing the state of the art is understanding how data is evolving over the years and how it correlates with the results found. Therefore, we built two evolution charts, depicted in Figure 3.3 a and 3.3 b. The former portrays the number of publications present in each of the digital libraries considered and described in section 3.1.1.2. The latter aggregates all of this information and depicts the total number of publications over the years. Both charts present data ranging from 2010 until 2020.

According to the charts' results, we can verify that the number of publications is growing linearly year over year. This publications' linear relation is easily verified for the ACM Digital Library and Scopus. However, such relation is more complicated to observe in the IEEExplore library when considering the entire time period. Nevertheless, if we only consider the last five years, such relation becomes more apparent for this particular library.

Also, by attentively observing the chart depicted in Figure 3.3 b, we can notice that the number of publications is relatively stable in the first three years (2010, 2011, and 2012). However, from 2013 until 2020, the number of publications increases almost linearly. Such a trend motivated us to consider the range of 2013 to 2020 as an inclusion criterion, as previously mentioned in section 3.1.2.1.



(a) Publications Evolution Over Time per Library.

(b) Publications Evolution Over Time in Total.

Figure 3.3: Distribution of publications available over time for the ACM Digital Library, Scopus, and IEEExplore libraries.

We believe that the reason behind the aforementioned linear increase in the results is due to the nature of the current dissertation's primary goal, which consists of creating an extensible PBT framework capable of supporting multiple search strategies. Such strategies include those in which the testing process is driven by following some form of feedback. As we could verify from the *Descriptive Synthesis*, in section 3.1.4.1, most of the selected publications propose strategies based on white-box and grey-box techniques. These type of techniques are very complex and demand a lot of computation power to be viable. Since the computation power increases year by year and these techniques continuously get improved, more and more solutions are proposed, leading to a higher number of publications available over the years.

3.1.5.2 Analysis

In section 3.1.3, we could verify that the selected publications can be grouped according to their domain. In total, we have identified five different domains directly related to this dissertation's scope. We were also able to verify that each solution within its respective domain has a unique contribution and that, regardless of the solutions' domain, they all contribute positively in terms of using unique strategies — essentially informed — to drive the testing process. However, and as expected, there are also some drawbacks associated with each domain. An overview of each domain, its benefits and drawbacks, is made in the following paragraphs.

In the SBT domain, the associated solutions use metaheuristics algorithms to drive the testing process while aiming to increase path coverage by focusing on specific paths. The principal benefit of these solutions' domain is that the algorithms used are well studied, making their implementation and adaptation easier. However, such algorithms are complex and may require a lot of configuration. Therefore the results are bounded to the algorithms' limitations.

Regarding CGF, Concolic Testing, and Hybrid fuzzing, and despite them being considered different domains, their goals and methodologies are very similar. In fact, most of the solutions associated with these domains aim at increasing code coverage and bug detection in the most efficient way. However, and despite being successful in meeting their objectives, these solutions still require a lot of time to achieve good coverage and bug detection results. A particularly interesting fact about these domains' solutions is that while some of them blindly aim at increasing code coverage, others focus on exercising new program behaviors by targeting specific code regions. The exciting part about this is that it introduces a conflict on whether we should maximize coverage or focus on specific code sections.

The publications related to the PBT domain propose different solutions to enhance the current state of PBT test case generators, which are predominantly random. The proposed solutions are mainly based on the solutions already proposed in the other domains. In fact, some of them are even the integration of those same solutions into PBT itself. According to the respective publications, the results achieved by these solutions are better than the currently available test case generators since they achieve higher code coverage levels and are able to find more bugs.

3.1.5.3 Research Questions

In this last section, we address the research questions that drove the entire SLR process. Therefore, an answer to each one of the research questions is presented here.

SRQ1 *Which are the most advanced PBT search strategies?* Most PBT tools only support purely random test cases generators or an enhanced form of this type of generator that considers exceptional corner cases. However, new search strategies have been proposed and adopted by some PBT frameworks. Such strategies involve the integration and adaptation of solutions from Fuzzing, mainly CGF (Padhye et al. [36, 37], and Lampropoulos et al. [25]), Reinforcement Learning (Reddy et al. [42]), and SBT (Löscher and Sagonas [30, 28]) into PBT.

SRQ2 *How can the current PBT search strategies be improved?* Current PBT search strategies can be improved in several ways. On one hand, such strategies can be improved by simply integrating ART [7] into PBT. On the other hand, and in a more informed perspective, the PBT search strategies can be improved by adapting other solutions from other domains, such as SBT, CGF, Concolic Testing, and Hybrid Fuzzing, which also address the test data generation problem.

SRQ3 *What are the most promising alternatives to the current adopted PBT search strategies?* The Fuzzing domain's solutions, mainly those from Hybrid Fuzzing, which combines CGF with Concolic Testing, are the most promising alternatives to the current PBT search strategies. This is because such solutions, despite being very time-consuming, are continuously getting improved and are the ones that achieve the best results in terms of combining higher coverage and bug detection values in an efficient way.

3.2 Summary

A SLR was conducted to review the state of the art of PBT search strategies. As a result of the conveyed literature review, we were able to identify 23 key publications from a total of 659, which we believe to be representative of the state of the art of this dissertation's scope.

From the selected publications, and apart from the PBT domain, we have identified four other domains that pursue the same goal as ours. SBT, CGF, Concolic Testing, and Hybrid Fuzzing compose the additional identified domains.

The domains' diversity found led us to conclude that the test data generation problem we are trying to solve is a problem that has been tackled over the years, not only in PBT but also in other domains. As a matter of fact, such a problem has gained a lot more attraction in the last years due to the increase in computation power and in the algorithms' efficiency.

Despite such diversity and the fact of each proposed solution being unique within its respective domain, it was also possible to conclude that the considered domains share similar goals. Their main goals consist of generating test cases using some form of feedback from the program under evaluation in order to test a system appropriately. The most popular forms of feedback used by the considered solutions are code coverage and bug detection.

The difference between the identified solutions' domains relies on the approach taken, which consequently leads to different results. While some solutions prioritize higher coverage over performance and vice-versa, others try to balance both evaluation metrics.

According to the literature and regarding PBT, our dissertation's primary focus, we were able to conclude that a few informed search strategies have already been proposed. Such strategies involve the integration and adaptation of concepts from the other identified domains, mainly from CGF, SBT, and also from Reinforcement Learning.

Chapter 4

Problem Statement

Throughout this chapter, we describe the problem that this dissertation aims to solve. Section 4.1 presents the current issues that led to the definition of the problem under study. Section 4.2 contains the defined hypothesis that this dissertation aims to validate. Section 4.3 details the scope of the problem we are trying to solve. Section 4.4 describes our proposed solution in order to solve the defined problem. Section 4.5 details how the proposed solution will be validated and evaluated. Finally, Section 4.6 closes this chapter by presenting the conclusions concerning the problem under study.

4.1 Current Issues

The heavy dominance of the random concept generally associated with PBT generators composes a critical problem in PBT since it implies a total arbitrary search over the input space. Therefore, the need for integrating other types of strategies in PBT, namely those which are informed and have already proved their value, is currently something worthy of exploring.

Chapter 3 contains several solutions from various domains that focus on generating test data while incorporating some form of the SUT's feedback to drive the search process. SBT, CGF, Concolic Testing, and Hybrid Fuzzing compose all the domains identified and related to the identified problem, apart from the PBT domain itself. An overview of each one of these domains is presented in Chapter 2.

Regardless of the domain, each identified solution either has a unique way of generating test data or enhances an already existent solution. Either way, in the end, they all share the same goal of generating test data that, by following some form of feedback, allow tests to be more predisposed to exercising several different behaviors and consequently be more prone to find failures. Coverage and fault detection are usually the most popular forms of feedback used to drive the search process.

According to the analysis made in Section 3.1.5.2, the identified solutions tend to be problem dependent. This means that even though some solutions perform better in some scenarios, that does not mean that they always perform better than the alternatives. In this context, by *better*, we mean achieving higher coverage values or fault detection rates in the minimum time possible.

All things considered, it is reasonable to assume that the development of a meta-search strategy capable of ensuring the absence of errors in a system is not something trivial, or even possible. Therefore, providing a PBT framework with the ability to use different and already existent search strategies seems more logical and the most promising approach.

As far as our knowledge goes, JQF [36] is the only identified solution that allows selecting different informed search strategies. Nevertheless, JQF is no *silver bullet* since it is endowed with an architecture not easily extensible and whose strategies are mostly coverage-guided based. These aspects compose JQF's most critical limitations because, as we could conclude from the last chapter, solutions from other domains also present promising, if not even better, results.

4.2 Hypothesis

As we could conclude from the previous chapters, the test data generation problem is a well-known problem in the software testing world that multiple domains have addressed over time. At the time of this writing and despite the efforts made, there is no evidence of a proposal capable of outperforming the current alternatives in every possible scenario. Given the nature of the aforementioned problem, which can be classified as a scenario-dependent problem, we decided to build this dissertation around the following hypothesis:

> *"Empowering a PBT framework with the ability to extend its search strategies with multiple and custom algorithms (either informed or not) to guide the test data generation process would allow one to (a) choose the most suitable search strategy for a particular problem, or (b) implement meta-strategies to automate that selection. This ability would provide higher control over the testing strategy, thus increasing both robustness and confidence in the outcomes."*

The considered hypothesis induces an architectural design that would allow PBT frameworks to select not only the most suitable search strategy to a specific problem but also the possibility of integrating state of the art techniques, thus being always up-to-date with most advanced strategies.

Such design would also enable the implementation of meta strategies focused on automating the search strategy selection process. It would be expected from these meta strategies the ability to, on the fly, alternate between search strategies depending on the most suitable strategy to the problem in hands.

4.3 Scope

Even though PBT can be explored at different levels, the ineffective exploration of the input space by the current, and typically random, PBT search strategies is still the most concerning problem in the PBT domain. According to the analysis made in Section 3.1.5.2, several solutions from different and specific domains can be used to address such a problem. Therefore, this dissertation's scope includes not only PBT search strategies but also search strategies pertaining to the domains of the solutions identified in Chapter 3.

4.4 Proposed Solution

Aiming to become closer to solving the test data generation problem and considering the hypothesis defined in Section 4.2, we propose in this dissertation the endowment of a PBT framework with an easily extensible and scalable architecture capable of supporting several different search strategies. By supporting multiple search strategies, we mean supporting search strategies from various domains that, apart from the PBT domain itself, have already proven to achieve promising results. All the details concerning our proposed solution and its implementation will be thoroughly described in Chapter 5 and 6.

4.5 Validation and Evaluation

In the interest of assessing and validating our central hypothesis, a set of several different strategies will be executed over multiple benchmarks. Even though each of these strategies will be different, they will share the same base configuration to enable an appropriate comparison of the extracted metrics within benchmarks and between strategies. Concerning the benchmarks, they will be selected according to their uniqueness and individual characteristics so that we can better understand how the different strategies will behave under different circumstances.

By performing the aforementioned evaluation, we aim to prove that there is no search strategy capable of outperforming all the current alternatives by showing that the ideal search strategy for each benchmark is different. Consequently, by doing so, we also prove the need for a framework capable of supporting multiple search strategies to allow us to select the most adequate search strategy for a particular scenario in order to increase the robustness and confidence in the outcomes.

The strategies' and benchmarks' definitions will be presented and detailed further in this dissertation, in Chapter 7. Moreover, and with the sole purpose of guiding the evaluation and validation phase, a set of research questions will be defined in the same chapter, in section 7.2. In the end, and after carefully analyzing the results, it is expected to be possible to answer, with confidence, those same questions.

4.6 Conclusions

Currently, the total arbitrary search over the input space, implied by the typically random-based generators, composes the most concerning problem in PBT. This problem is a well-known problem that other domains, apart from the PBT domain itself, have addressed over time with the proposal of unique solutions, whose results are positively promising. However, such results tend to be problem-dependent, meaning that the given solutions are no *silver bullet*.

Therefore, and having no evidence of a proposal capable of performing better than the current alternatives in every possible scenario, we decided to build this dissertation around the hypothesis that by having an extensible framework capable of supporting multiple search strategies we *would allow one to (a) choose the most suitable search strategy for a particular problem, or (b) implement meta-strategies to automate that selection*.

Once a fully extensible framework is built and multiple search strategies pertaining to various domains are fully supported, the considered hypothesis can be appropriately assessed and validated by executing several different strategies over multiple test scenarios in order to understand their behavior under different circumstances.

Chapter 5

Designing an Extensible PBT Framework

This chapter describes all the aspects that need to considered when designing an extensible PBT framework that supports multiple search strategies. Section 5.1 provides context for the whole chapter. Section 5.2 outlines the objectives and requirements underlying the design of an extensible PBT framework. Finally, Section 5.3 details all the factors that must be considered at the architectural level. By the end of this chapter, the reader should have all the needed knowledge to replicate this dissertation's proposal.

5.1 Context

In the previous chapter, Chapter 4, we have discussed the problem we are trying to solve and what we consider to be the most promising solution to solve it. According to our proposal, briefly defined in Section 4.4 of the same chapter, such a solution consists of endowing a PBT framework with the ability to be easily extendable and support multiple search strategies.

Based on this proposed solution, we can promptly assume that its development presupposes a set of fundamental aspects that must be considered, regardless of whether we are creating an entirely new framework or selecting one already available.

Therefore, in the following sections, we will cover all the objectives, requirements, and architectural design decisions that one must consider when designing an extensible PBT framework or any kind of framework, which purpose is similar to the one we are trying to achieve in this dissertation.

5.2 Objectives and Requirements

The main objective of developing an extensible PBT framework that supports multiple search strategies is to grant the developer with the possibility to choose or customize a given search strategy according to his needs. An apparent issue with this objective is the necessity to define how deeply we want the developer to be able to customize a given search strategy.

Concerning such issue, and based on the hypothesis defined earlier in Section 4.2, we believe that a developer should be able to fully control or even implement its own search strategy. However, for this to be possible, a global and standard interface must be defined, and every single search strategy must implement it. Listing 5.1 depicts an example of a possible search strategy interface definition.

```
interface SearchStrategyInterface {
  hasInput: () => boolean
  getInput: () => any
  handleResult: (data: any[]) => void
}
```

Listing 5.1: Search Strategy Interface Definition Example

Although the interface definition depicted above can slightly differ from tool to tool, essentially in terms of naming conventions and required parameters, the three interface methods defined are mandatory.

The mandatory character underlying such methods is directly related to the PBT's framework cyclic validation process, which demands that a new input must be tested and then validated on each iteration. Being such methods responsibility of the search strategy, it is up to that same strategy to control which input will be tested at each iteration — `getInput()` —, when to stop testing — `hasInput()` —, and what to do with the result of every single input — `handleResult()` — . It is also worth mentioning that, apart from the stop testing criteria defined by a given search strategy, the framework's testing process can also be halted by the framework's satisfiability checker, whose implementation also varies from tool to tool. An overview of the PBT cyclic validation process is depicted in Algorithm 1.

Algorithm 1 PBT Cyclic Validation Process

```
Input strategy used for driving the testing process
Output An object containing the property satisfiability information
satisfiabilityData ← satisfiabilityChecker()
while strategy.hasInput() do
    inputData ← strategy.getInput()
    result ← assertions.forall(assertion => assertion(inputData) == true)
    satisfiabilityData ← satisfiabilityChecker(result)
    strategy.handleResult(satisfiabilityData)
    if satisfiabilityData.satisfiable then
        return satisfiabilityData
    end if
end while
return satisfiabilityData
```

A search strategy that exclusively follows the interface illustrated in Listing 5.1 can be considered on its default state. To complement it, we need to extend the search strategy in order to be able to customize it with new features. How we extend the search strategy depends on the selected architectural design.

5.3 Architectural Design

Based on the objectives and requirements defined in the last section, we have identified the need to design a PBT framework with an architecture capable of supporting not only several search strategies but also a high level of customization. Therefore, we are looking for architectural designs capable of simultaneously scaling and be easily extendable.

While considering these architectural requirements, we can immediately acknowledge the existence of a set of multiple design patterns [31] that stand out as candidate solutions for our architectural problem. Factory Method, Builder, Composite, Decorator, Strategy, and Template Method compose the list of candidate patterns. However, given the fact that most of these patterns are substantially based on inheritance principles, creating a solution based on composition does not seem easy to implement and maintain. In fact, one of the most challenging problems when dealing with inheritance, specifically with multiple inheritance, is the diamond problem [2], which is when two classes B and C inherit from A, and class D inherits from both B and C, as depicted in Figure 5.1.

With the ambiguity underlying this problem, other problems [2] also emerge: *the encapsulation of inheritance*, *the common ancestor duplication problem*, *the common ancestor name conflicts problem*, and *the duplicate parent operation invocation problem*.

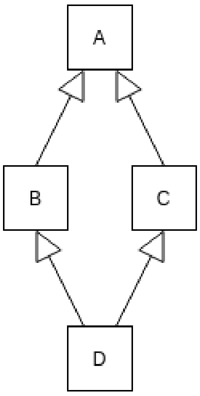


Figure 5.1: Multiple inheritance - Diamond Problem.

Given these problems, we are required to change our approach to a more robust one. As such, we can use a mixin-based approach [2] that, despite not being considered a design pattern, is something that solves our architectural problem and that most programming languages support. Mixins are a flexible way to provide methods that implement a particular behavior to other classes without establishing a complex inheritance relation. Its application is ideal when either we want to provide a lot of optional features for a class or when we want to use one particular feature in multiple different classes. Additionally, the mixins usage encourages code reuse and can avoid the multiple inheritance implicit problems. An overview of a mixin-based approach is depicted in Figure 5.2, which represents the creation of a new object whose behaviors are not intrinsically defined but rather defined by the associated mixins.

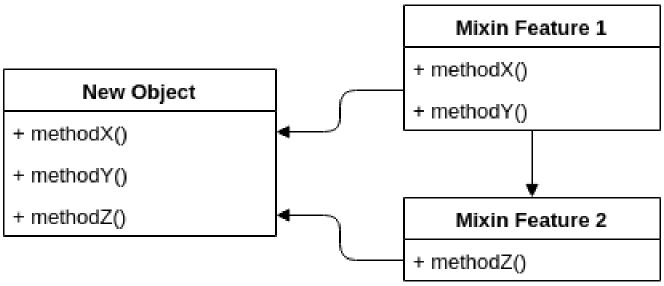


Figure 5.2: Mixin-Based approach.

Despite being supported by most programming languages, mixins' implementation in each language differs, and their potential may not be considered the same. Hence, an architecture purely mixin-based may not be possible for every single programming language. However, to address this problem, we can use a hybrid approach that combines both mixins and some of the aforementioned design patterns to overcome possible implementation gaps. In fact, this hybrid approach, due to its potential, seems the most promising one when designing an extensible PBT framework that supports multiple search strategies.

Chapter 6

Solution

Throughout this chapter, we describe how the problem presented in Chapter 4 was tackled and solved. Section 6.1 explains the process of selecting the most adequate framework for implementing our proposed solution. Section 6.2 provides all the needed information about the selected framework, mainly regarding its property validation process (Section 6.2.2) and supported search strategies (Section 6.2.3). Section 6.3 presents all the details concerning the implementation of our proposed solution, ranging from all the decisions made at the architectural level to the definition of new search strategies. Finally, Section 6.4 summarizes this chapter by presenting an overview of all the topics mentioned before.

6.1 Framework Selection Process

Considering the proposed solution defined in Section 4.4, there are two ways we can use to proceed with the implementation of such a proposal: one is to create, from scratch, an entirely new PBT framework; another is to make use of one of the numerous PBT frameworks already available. While the former may seem ideal in terms of enabling total control over the framework's architecture, it also involves several complex features that go beyond this dissertation's defined scope, as described in Section 4.3. Thus, and despite not being the ideal solution, the latter seems the more reasonable approach.

Therefore, to select the ideal framework to use as the basis for this dissertation's development, we have decided to consider not only all the tools specified in Table 2.1 of Chapter 2 but also the JQF framework that, as we previously discussed in Section 4.1, is the only tool that offers the possibility of selecting multiple search strategies. However, JQF, which is built on top of junit-quickcheck, was excluded right away due to its inflexible architecture and the fact that it uses an abstraction of property-based testing. With the exclusion of JQF, the framework selection process was reduced to all the PBT tools defined in Table 2.1 of Chapter 2.

Aiming at finding the perfect candidate among such tools, we have decided to consider two main factors: the tool's repository activity and the tool's feature-rich level. As such, and based on the activity-based factor, we decided to discard all the tools whose repository has not been updated at least once in the current year. After this process, we found out that only the following seven frameworks matched this criterion: FsCheck, Hypothesis, junit-quickcheck, jqwik, JSCheck, FluentCheck, and fast-check. From these remaining tools, fast-check immediately stood out from the rest. The reason for this was due to fast-check having the most up-to-date repository and being the most feature-rich framework found by allowing the integration with popular test frameworks (e.g., jasmine, jest, mocha, etc) and supporting not only a wide range of `Arbitraries` types but also the ability to shrink and drive the test data generation process in a smarter way using a `biased` search.

The selection of fast-check could not be made without considering the FluentCheck framework, which is a property-based testing framework inspired by fast-check itself and focused on more innovative search strategies and statistical confidence calculation. Nevertheless, both tools were at completely different development stages at the moment of this selection process, being fast-check in a more advanced phase and FluentCheck in a more early phase.

All things considered, it would seem more logical to select the fast-check framework due to its development stage. However, and considering this dissertation's scope and hypothesis, both defined in the previous chapter, we decided to select the FluentCheck framework. The selection of this framework made more sense to us because not only could it be seen as a newer and improved version of fast-check, but it also revealed the perfect opportunity in terms of enabling greater control over the entire framework architecture due to its embryonic development stage.

6.2 FluentCheck

The current chapter’s section provides an overview of the FluentCheck framework and its unique characteristics, followed by a detailed description of both FluentCheck’s property validation process and currently supported search strategies. It is also worthy of mentioning that only the details that fall within this dissertation’s scope will be covered in this section.

6.2.1 Overview

FluentCheck is a property-based testing framework inspired by fast-check and focused on more innovative search strategies and statistical confidence calculation. Apart from these characteristics, FluentCheck also distinguishes itself through its API and the type of quantifiers supported.

Concerning its API and as the name itself suggests, FluentCheck aims to provide a `fluent` way to define properties. In that sense, the framework is provided with an API designed to allow the properties’ definition to be very compact and behavior-driven. To have a visual representation of FluentCheck’s API, let us consider the example depicted in Listing 6.1, which represents a complex property specification.

```
it(’should find if two different stacks behave the same’, () => {
  expect(fc.scenario()
    .forall(’es’, fc.array(fc.integer()))
    .given(’s1’, () => new Stack<number>())
    .and(’s2’, () => new Stack<number>())
    .when(({es, s1}) => s1.push(...es))
    .and(({es, s2}) => s2.push(...es))
    .then(({s1, s2}) => s1.size() === s2.size())
    .and(({es, s1}) => s1.size() === es.length)
    .and(({es, s2}) => s2.size() === es.length)
    .check()).to.have.property(’satisfiable’, true)
})
```

Listing 6.1: FluentCheck Property Specification Example

The support of both universal and existential quantifiers composes another unique feature of the FluentCheck framework. In fact, and as far as our knowledge goes, FluentCheck is the only framework capable of supporting both types of quantifiers, whereas the current alternatives can only support universal quantifiers.

6.2.2 Property Validation Process

In order to understand how the property validation process is defined in the FluentCheck framework, we first need to understand how the framework assembles its scenarios. For that, let us consider the example depicted in Listing 6.1, which purpose was to outline FluentCheck’s API.

By analyzing the considered example we can see that a given scenario is assembled through a series of API calls. Consequently, each of these calls is responsible for creating a new class that extends the main one used to build the default scenario on line 2. Thus, we can conclude that the FluentCheck framework assembles its scenarios through a composition-based process.

Based on this process and through the implementation of a runnable method in every class that composes a given scenario, FluentCheck is able to validate properties through a series of callbacks that are made between those same classes. However, and as we previously mentioned in Chapter 5, for this process to be complete, there must be some sort of cyclicity introduced.

Listing 6.2 portrays the FluentCheck framework classes responsible for introducing the aforementioned cyclicity into the property validation process. Furthermore, all the previously mentioned details concerning the property validation process are also noticeable in this listing.

```
class FluentCheckUniversal<K extends string, A, P extends TestCase, G extends P & Record<K, A>> extends FluentCheck<G, P> {
  private cache: Array<FluentPick<A>>
  private dedup: Arbitrary<A>

  constructor(protected readonly parent: FluentCheck<P, any>, public readonly name: K, public readonly a: Arbitrary<A>) {
    super(parent)
    this.dedup = a.unique()
    this.cache = this.dedup.sampleWithBias(1000)
  }

  protected run(testCase: TestCase, callback: (arg: TestCase) => FluentResult, partial: FluentResult | undefined = undefined): FluentResult {
    const example = partial || new FluentResult(true)
    const collection = partial === undefined ? this.cache : this.dedup.shrink(partial.example[this.name]).sampleWithBias(1000)

    for (const tp of collection) {
      testCase[this.name] = tp
      const result = callback(testCase)
      if (!result.satisfiable) {
        result.addExample(this.name, tp)
        return this.run(testCase, callback, result)
      }
    }

    return example
  }
}

class FluentCheckExistential<K extends string, A, P extends TestCase, G extends P & Record<K, A>> extends FluentCheck<G, P> {
  private cache: Array<FluentPick<A>>
30    private dedup: Arbitrary<A>
31
32    constructor(protected readonly parent: FluentCheck<P, any>, public readonly name:
           K, public readonly a: Arbitrary<A>) {
33      super(parent)
34      this.dedup = a.unique()
35      this.cache = this.dedup.sampleWithBias(1000)
36    }
37
38    protected run(testCase: TestCase, callback: (arg: TestCase) => FluentResult,
          partial: FluentResult | undefined = undefined): FluentResult {
39      const example = partial || new FluentResult(false)
40      const collection = partial === undefined ? this.cache : this.dedup.shrink(
            partial.example[this.name]).sampleWithBias(1000)
41
42      for (const tp of collection) {
43        testCase[this.name] = tp
44        const result = callback(testCase)
45        if (result.satisfiable) {
46          result.addExample(this.name, tp)
47          return this.run(testCase, callback, result)
48        }
49      }
50
51      return example
52    }
53  }
```

Listing 6.2: FluentCheckUniversal and FluentCheckExistential Classes

By carefully analyzing the classes depicted above, we can infer that both classes are fundamentally the same, differing only on the satisfiability checker, which is implicitly implemented on lines 18 and 45. The reason for this difference is because both classes represent different quantifiers with opposite satisfiability criteria.

Moreover, it is also possible to conclude that both classes are not only responsible for introducing the cyclicity mentioned above but also for containing all the information related to the arbitraries and search strategy used by the framework, ranging all the way from the use of bias to the utilization of shrinking. All the details concerning the supported search strategies will be discussed in the following section.

6.2.3 Supported Search Strategies

As we previously mentioned in Section 6.2.2, all the logic concerning FluentCheck's supported search strategies is self-contained in the `FluentCheckUniversal` and `FluentCheckExistential` classes.

According to the definition of such classes, we can assume that the FluentCheck framework only supports one single random-based search strategy endowed with several hard-coded features. An overview of those features is given in the following subsections.

6.2.3.1 Sampling Without Replacement

By generating unique input collections without duplicated values, the support of sampling without replacement allows the search strategy to cover a broader range of input values. Consequently, this feature also increases the overall strategy efficiency because no input is tested more than once.

6.2.3.2 Caching Mechanism

By caching the input collections generated by each arbitrary, the caching mechanism allows the reuse of such collections by the search strategy on each callback of the FluentCheck property validation process, previously described in Section 6.2.2.

6.2.3.3 Biased Sampling

Biased sampling means that the generated input collections will not be entirely random. Instead, they will be partially filled with corner cases and then complemented with random values. This adjustment on the generated collections implies that input values that are considered to have a higher probability of triggering an unusual behavior will be tested first.

6.2.3.4 Shrinking Process

This process can be seen as a best-effort, nice-to-have feature of PBT by allowing a given test case to be repeatedly minimized to its most concise form. More details on this process were already provided in Section 2.2.3.

6.3 Implementation

The current chapter's section contains all the details concerning the implementation of our proposed solution. Such details will be thoroughly described in the following subsections in the same order as they were implemented.

6.3.1 Abstracting FluentCheck's Search Strategy

The initial phase of implementing our proposed solution consists of abstracting all the information associated with FluentCheck's search strategy, which, as we previously specified in Section 6.2.2, is self-contained in the `FluentCheckUniversal` and `FluentCheckExistential` classes.

Striving to abstract the aforementioned information, which includes references to multiple features ranging from the arbitraries used to the search strategy configuration itself, our first challenge was to determine how we could redesign the FluentCheck framework architecture so that such features could be optionally integrated, or not, into the search strategy itself. Given this challenge, and taking into account all the architectural considerations specified in Section 5.3, we decided to begin our implementation with a purely mixin-based approach.

Considering the selected approach, we started by defining a new class, the `FluentStrategy` class, which purpose is to contain all the information concerning the arbitraries and the strategy configuration used in a given scenario. Such strategy configuration includes details concerning the strategy general settings such as the `sample size` used on the initial collections and those generated by shrunk arbitraries.

Aiming to endow the `FluentStrategy` class with the ability to be easily extendable, we also defined a global and generic interface that the newly defined class must follow. This interface definition, illustrated in Listing 6.3, was based on the objectives and requirements defined in Section 5.2.

```
interface FluentStrategyInterface {
  hasInput: <K extends string>(arbitraryName: K) => boolean
  getInput: <K extends string, A>(arbitraryName: K) => FluentPick<A>
  handleResult: (inputData: any[]) => void
}
```

Listing 6.3: FluentCheck's FluentStrategy Interface

Apart from the methods defined in the interface illustrated in Listing 6.3, we also saw fit implementing two additional methods, the `addArbitrary()` and the `configArbitrary()` methods, whose purpose is to add and configure, respectively, all the information concerning the arbitraries used by the `FluentStrategy` class. Since this information includes all the details concerning the test data generation process and consequently references multiple features (e.g., biased sampling, shrinking, etc), these methods' implementation would not be possible without evolving our architecture to a hybrid one that integrates the Template Method design pattern into our previously defined mixin-based approach. With the integration of this design pattern, we allowed the `FluentStrategy` class associated mixins to either entirely or partially redefine this class methods' behaviors.

A particular detail concerning our implementation, precisely our interface definition, is that the interface methods implemented directly by the `FluentStrategy` class have no associated behavior. Such behavior is instead defined by the `Random` mixin class, which acts as the base strategy and is responsible for defining the random and default behavior of such methods. Apart from the `Random` mixin class definition, several other mixins were also defined. In fact, a total of four additional mixins were defined, one for each of the features mentioned earlier in the subsections of section 6.2.3. Figure 6.1 provides an overview of the architecture defined in the current section implementation phase.

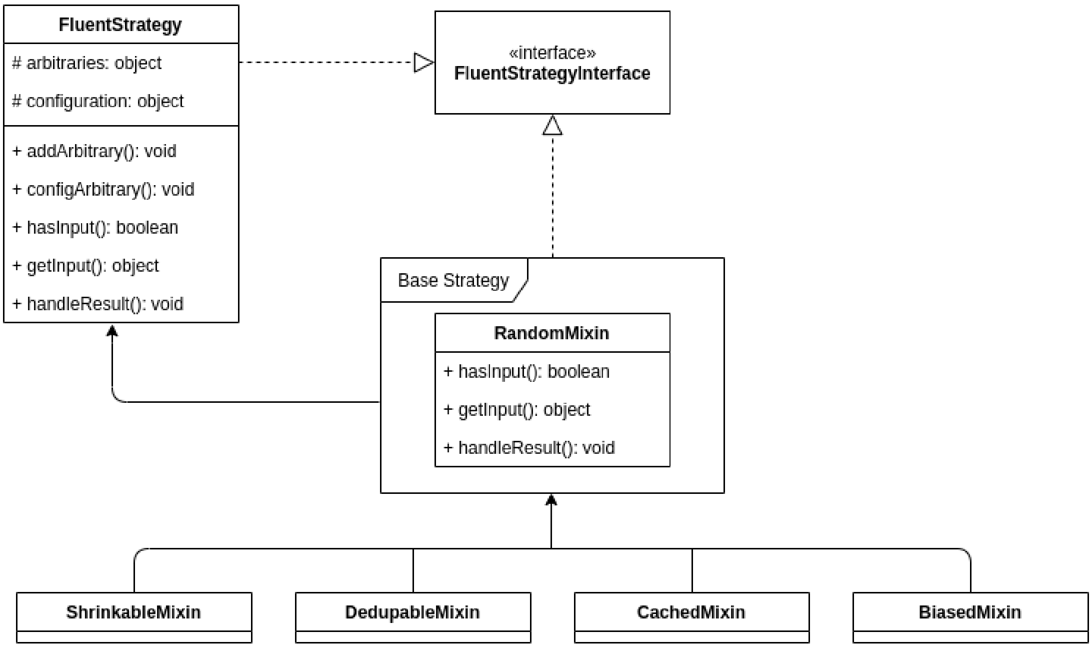


Figure 6.1: FluentCheck's search strategy architecture.

After implementing all the components defined above, we proceeded with their integration into the FluentCheck property validation process. Such integration was achieved by refactoring the `FluentCheckUniversal` and `FluentCheckExistential` classes, which led to the definition of the `FluentCheckQuantifier` abstract class. Listing 6.4 exhibits the final result of the refactoring process.

```
abstract class FluentCheckQuantifier<K extends string, A, Rec extends ParentRec & Record<K, A>, ParentRec extends {}>
  extends FluentCheck<Rec, ParentRec> {

  constructor(
    protected readonly parent: FluentCheck<ParentRec, any>,
    public readonly name: K,
    public readonly a: Arbitrary<A>,
    strategy: FluentStrategy) {

    super(strategy, parent)
    this.strategy.addArbitrary(this.name, a)
  }

  protected run(
    testCase: WrapFluentPick<Rec>,
    callback: (arg: WrapFluentPick<Rec>) => FluentResult,
    partial: FluentResult | undefined = undefined,
    depth = 0): FluentResult {
```

```

    this.strategy.configArbitrary(this.name, partial, depth)

    while (this.strategy.hasInput(this.name)) {
      testCase[this.name] = this.strategy.getInput(this.name)
      const result = callback(testCase)
      if (result.satisfiable === this.breakValue) {
        result.addExample(this.name, testCase[this.name])
        return this.run(testCase, callback, result, depth + 1)
      }
    }

    return partial ?? new FluentResult(!this.breakValue)
  }

  abstract breakValue: boolean
}

class FluentCheckUniversal<K extends string, A, Rec extends ParentRec & Record<K, A>, ParentRec extends {}>
  extends FluentCheckQuantifier<K, A, Rec, ParentRec> {
  breakValue = false
}

class FluentCheckExistential<K extends string, A, Rec extends ParentRec & Record<K, A>, ParentRec extends {}>
  extends FluentCheckQuantifier<K, A, Rec, ParentRec> {
  breakValue = true
}
```

Listing 6.4: Refactoring Process Final Result

Lastly, to complete the initial phase of our implementation, we proceeded with the definition of an API in order to allow a developer to benefit from the extensibility introduced, with the aforementioned components, by being able to customize the search strategy to be used in a given scenario. This API definition is implicitly embodied in the `FluentStrategyFactory` class that is also responsible for creating the search strategy object. Moreover, we decided to define such API in a fluent style, to maintain coherence with FluentCheck's already available API.

Listing 6.5 illustrates the defined API usage. For explanatory purposes, we decided not only to consider the same property as the one defined previously in Listing 6.1 but also to include all the possible API calls.

```
it('should find if two different stacks behave the same', () => {
  expect(fc.scenario()
    .config(fc.strategy()
      .withRandomSampling()
      .withBias()
```

```
        .usingCache()
        .withShrinking()
        .withoutReplacement()
      )
      .forall('es', fc.array(fc.integer()))
      .given('s1', () => new Stack<number>())
      ...
})
```

Listing 6.5: FluentCheck's Search Strategy API

With the abstraction of FluentCheck's search strategy finished, which was the whole purpose of this section's implementation, we shifted our focus towards the implementation of multiple new search strategies. As such, a detailed description of the newly implemented search strategies will be provided in the following sections.

6.3.2 Source Code Constants Extraction

Similar to the already available biased sampling feature, and based on the Rojas et al. [43] proposal, previously described in Section 3.1.4, we decided to drive the next phase of our implementation towards the belief that, by extracting constants from multiple sources, we would provide greater confidence over test specifications and an increase on the property validation's process efficiency.

Aiming to fulfill our belief, and based on the promising results achieved by Rojas et al.'s proposal, coupled with the common presence of constant variables in source code, we decided to define and implement a strategy responsible for extracting constants from source code, which in turn happens to partially compose one of the four strategies proposed by Rojas et al.

Furthermore, and given the fact that the extracted constants can be associated with several different types, we also decided to define the scope of our strategy in terms of the type of constants to be extracted. Such scope was defined by considering all the possible primitive data types, which we reduced to three major groups: `numerics`, `strings`, and `booleans`. From these three groups, we decided to reduce our scope to `numeric` and `string` constants only since `booleans` can only assume one of two possible values, which makes their extraction not worthy.

All things considered, we ended up with a new strategy proposal that consists of extracting `numeric` and `string` constants from source code, and whose implementation was accomplished by breaking its logic into two distinct phases: one that immediately occurs once the entire search strategy is built and another one that takes place during the test data generation process.

Although this section's strategy implementation was achieved by breaking its logic into two separate phases, its usage was abstracted into a single API call — `withConstantExtraction()` —, which is responsible for enabling the extraction of constants from source code, and consequently their use.

A detailed overview of each of the phases that compose this section's strategy is presented in the following subsections.

6.3.2.1 Phase I - Constants Extraction Process

The first phase of this section's strategy is directly related to the implementation of the process responsible for extracting constants. This process was implemented in the `Constant-ExtractionBased` mixin and was designed to extract constants from multiple sources, ranging from the test specification methods to all the source code files that can compose a given program. Moreover, and given the complexity underlying such process, we also decided to break its implementation into a series of sequential stages.

Therefore, and striving to perform the constant extraction process, we begin by performing a lexical analysis — `tokenization` — of the considered sources. Alternatively, a syntactic analysis could also be performed, but we considered the `tokenization` process to be more suitable given our proposed strategy's scope. Moreover, and considering the complexity underlying the `tokenization` process, we decided to use a third-party tool, the espree[1] parser.

The `tokenization` process, once completed, is followed by a `filtering` stage that excludes all tokens apart from the `numeric` and `string-based` ones. The tokens resulting from this filtering stage are then used to generate new tokens. This was achieved by introducing a new stage responsible for concatenating and mutating all the extracted `string` constants and for applying basic arithmetic operations (e.g., addition, subtraction, division, multiplication) over all the extracted `numeric` constants.

With the introduction of the before-mentioned last stage, we aimed at generating new inputs that, despite not being present in the considered sources, are more likely to impact the program than others generated entirely randomly. A graphical representation of the entire constant extraction process just described is shown in Figure 6.2.

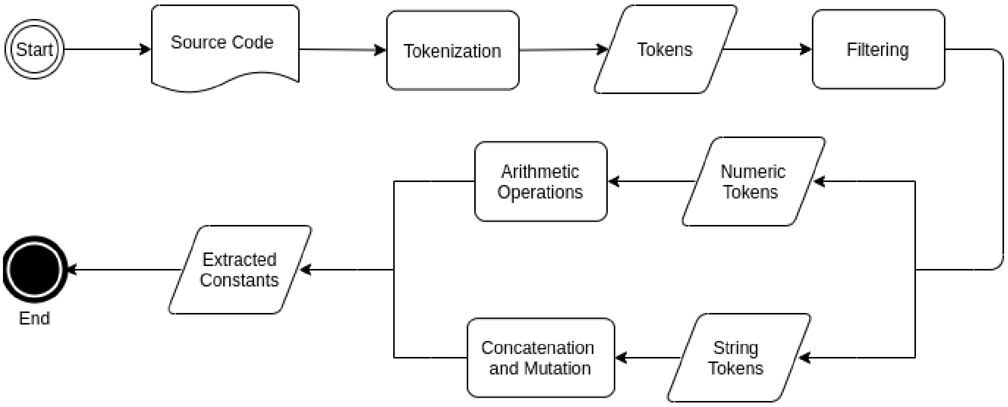


Figure 6.2: Constant extraction process workflow.

[1] Available: https://www.npmjs.com/package/espree

6.3.2.2 Phase II - Constants Extraction Usage

The second and last phase of this section's strategy implementation consists of endowing FluentCheck's arbitraries with the ability to use the extracted constants.

Aiming to provide such ability, we decided to follow a similar approach to the one used for considering corner cases, and which consisted of implementing an abstract method responsible for handling all the logic concerning the arbitrary ability to select a set of extracted constants. Such method was implemented on the `Arbitrary` abstract class, and therefore, was further implemented and overridden by all the arbitraries classes, each with its own logic.

A particular detail worthy of mentioning is that any other implementation, apart from the one considered, would only allow the usage of the extracted constants by numeric and string-based arbitraries (e.g., `ArbitraryInteger`, `ArbitraryString`, etc). Our implementation, by assigning each arbitrary with the responsibility to define which extracted constants to use, also allows the usage of such constants by non-basic arbitraries (e.g., `ArbitraryArray`, `ArbitrarySet`, `ArbitraryTuple`, etc).

6.3.3 Pairwise Testing

While abstracting FluentCheck's search strategy, we noticed that the test case generation process was being entirely accomplished and controlled by the recursivity implicitly introduced by the cyclic property validation process. Furthermore, we also noticed that such process, by being purely based on recursivity, was forced to exhaustively generate all possible test cases.

Given the exhaustive nature of such process, and being such test cases the result of combining all the inputs generated by the arbitraries involved in the testing process, we immediately concluded that the FluentCheck framework had no control over the total number of test cases generated and that such number exponentially increased with the number of arbitraries involved.

By considering the aforementioned exponential increase, we could easily predict a series of performance problems in test specifications involving either a considerable sample size or a significant number of arbitraries. In fact, such performance problems were unveiled by simply considering the example of a generic test specification involving three different arbitraries and FluentCheck's default sample size (1000). Even though it seems simple, such test specification, if valid, will generate and test a total of one billion test cases ($1000 \times 1000 \times 1000 = 1000^3$), which will end up taking a considerable amount of time.

Based on the generic test specification mentioned above, we were curious about how FluentCheck handled the performance problems underlying similar test specifications. However, after a quick overview of FluentCheck's internal test suite, we rapidly identified test specifications similar to the one just provided. Listing 6.6 describes one of those test specifications.

```
it('finds if additions are associative', () => {
    expect(fc.scenario()
      .forall('a', fc.integer(-10, 10))
      .forall('b', fc.integer(-10, 10))
      .forall('c', fc.integer(-10, 10))
      .then(({a, b, c}) => a + b + c === a + (b + c))
      .check()
    ).to.have.property('satisfiable', true)
})
```

Listing 6.6: FluentCheck's Addition Associative Property Specification

By carefully analyzing the listing depicted above, we could verify that the authors of the test specification addressed the before-mentioned performance problems by directly manipulating the domain of the arbitraries involved in the testing process. Such manipulation was performed by restraining the arbitraries integer domain to the interval of $[-10, 10]$, which only includes 21 unique values. By doing so, the authors bounded the maximum number of test cases of the depicted test specification to 9261 test cases $(21 \times 21 \times 21 = 21^3)$. Apart from FluentCheck's sample size default value, if no restrictions were applied, we would have a total of one billion test cases, which is approximately 110 times more than the actual value used.

Considering all the details discussed in the last paragraphs, we clearly identified the need for FluentCheck to establish a trade-off between performance and the sample size value applied in the test case generation process. Therefore, and being such a trade-off, the consequence of the exhaustive combinatorial process inherent to the FluentCheck framework, we decided to redesign the whole process of generating test cases by introducing a new concept: pairwise testing.

Pairwise testing is a combinatorial process designed to execute all possible discrete combinations of each pair of input parameters. By doing so, the number of test cases generated through pairwise testing is given by the function $T = Max(X) \times Max(X \backslash Max(X))$, where $Max(X)$ and $Max(X \backslash Max(X))$ correspond, respectively, to the first and second input parameters with the most extensive input range.

With the introduction of pairwise testing into the framework, we eliminated the aforementioned mentioned trade-off by bounding the number of test cases to a maximum of $SampleSize^2$. Such a bounding was possible given the function underlying the pairwise testing process and the fact that, in the FluentCheck framework, the following property is always satisfied: $\forall X$, $Max(X) <= SampleSize$.

Striving to implement pairwise testing into the framework, we had to consider several complex factors. First of all, we had to completely redesign FluentCheck's framework architecture in order to eliminate the recursivity associated with the test case generation process. Aiming to do so, we proceeded with the definition of the `FluentCheckRunner` class, in which we abstracted the property validation process, previously described in Section 6.2.2. Moreover, the definition of such a class also allowed us to transfer all the logic concerning the test case generation process to the `FluentStrategy` class. It worthy of mentioning that the before-mentioned abstraction did not include the abstraction of the `satisfiability checker`, which logic was kept on the `FluentCheckQuantifier` class.

Nevertheless, the complete elimination of the recursivity associated with the property validation process was not straightforward as just creating the before-mentioned class with an iterative-based process. The reason for this is because not only was the recursivity responsible for generating test cases but also for generating them in a particular order that allowed the existence of a satisfiability checker capable of solving the **Boolean Satisfiability Problem** for both universal ($\forall x$) and existential ($\exists x$) quantifiers (e.g., $\forall x \in X\ \forall y \in Y\ \exists z \in Z\ (x \vee y \vee z)\ \wedge\ (\neg x \vee \neg y \vee \neg z)$).

Therefore, and in order to properly eliminate such recursivity while maintaining the framework's core functionalities, we also enhanced the `FluentCheckQuantifier` satisfiability checker with the capacity to store context from each test iteration. By doing so, we allowed such a class to properly determine whether the testing process should be halted or not by having knowledge about all the test iterations and their associated test cases. Coupled with such improvement, we also ensured that the test cases were generated in the same order as they were when recursively generated. This last improvement was achieved by defining a complex algorithm that is not only responsible for generating all possible pairs of combinations but also for ensuring that they follow the order just described.

With the implementation of all the details mentioned above, we were able to endow the framework to either generate test cases exhaustively or through a pairwise testing combinatorial process that can be enabled through the `withPairWiseTesting()` API method. Listing 6.7 represents the before-mentioned method API usage in the same test specification defined in Listing 6.6, but now without any restrictions at the arbitraries' domain level.

```
it('finds if additions is associative', () => {
    expect(fc.scenario()
      .config(fc.strategy()
        .withRandomSampling()
        ...
        .withPairWiseTesting()
      )
      .forall('a', fc.integer())
      .forall('b', fc.integer())
      .forall('c', fc.integer())
      .then(({a, b, c}) => a + b + c === a + (b + c))
      .check()
    ).to.have.property('satisfiable', true)
})
```

Listing 6.7: FluentCheck's Addition Associative Property Specification with Pairwise Testing

6.3.4 Coverage-Guided Strategy

Following the implementation of the strategies previously described in the last sections, and based on the promising results achieved by coverage-guided solutions, we decided to finish our implementation phase with the implementation of an entirely new strategy: a coverage-guided strategy.

Furthermore, and given the before-mentioned strategy coverage-based character, we also decided to break its implementation into two separate phases: one responsible for tracking coverage and another responsible for introducing the coverage-guided logic into the FluentCheck framework. Both phases, detailed in the following subsections, were accomplished with the definition and implementation of the `CoverageTracker` and `CoverageGuidance` mixins, respectively.

6.3.4.1 CoverageTracker Mixin

The `CoverageTracker` mixin was the mixin defined and implemented in the first phase of this section's strategy. The purpose of this mixin was to endow the FluentCheck framework with the ability to track coverage on each iteration of the property validation process. To accomplish such purpose, we had to define two processes: one responsible for tracking coverage and another one responsible for determining the source code where coverage should be tracked.

Concerning the aforementioned first process, and considering source code instrumentation as a requirement for tracking coverage, we decided to use Istanbul[2], a JavaScript-based coverage tracker, to fulfill such process's purpose. Moreover, given that such tool is designed to measure coverage of complete test suites, we had to use it programmatically in order to take advantage of its numerous features.

With the definition of the process responsible for tracking coverage achieved, we proceeded with the definition of the second process that, as we previously mentioned, is responsible for determining the source code where coverage should be tracked. Given such process, we ended up concluding that tracking coverage of all test specification methods, apart from the ones responsible for the arbitraries' definition, was the most logical approach. Therefore, we developed a complex algorithm that, apart from being capable of extracting all the methods from a test specification, is also capable to recursively detect those methods' dependencies and, include them in the tracking coverage process.

By the end of this section's strategy first phase, we had fully defined and implemented mixin capable of tracking coverage over all the methods, and corresponding dependencies, involved in a given test specification.

[2] Available: https://github.com/istanbuljs/istanbuljs

6.3.4.2 CoverageGuidance Mixin

The second and last phase of the current section's strategy implementation consisted of defining and implementing the `CoverageGuidance` mixin, which, similar to the `Random` mixin, acts as a base strategy and is responsible for implementing the `FluentStrategyInterface`, previously specified in Listing 6.3.

However, given the nature of the newly implemented mixin, we decided to update such interface with the integration of the `FluentStrategy` class `configArbitrary()` method, whose internal logic is different for both base strategy mixins. Moreover, the before-mentioned method was also renamed to `configArbitraries()`. An overview of the final version of the `FluentStrategyInterface` is illustrated in Listing 6.8.

```
interface FluentStrategyInterface {
  configArbitraries: () => void
  hasInput: () => boolean
  getInput: () => WrapFluentPick<any>
  handleResult: (inputData: any[]) => void
}
```

Listing 6.8: FluentCheck's FluentStrategy Final Interface

Apart from implementing the `FluentStrategyInterface`, the before-mentioned mixin is also responsible for implementing the coverage-guided strategy workflow into the FluentCheck framework.

The aforementioned workflow implementation was achieved by considering the workflow implemented by FuzzChick [25], which we already described in Section 3.1.4. However, our implemented workflow differentiates itself from the one implemented by FuzzChick, and any other known implementation, in the sense that it includes unique innovations at the seed collection generation and mutation levels. While traditional coverage-guided strategies often randomly generate their seed collection, our implementation allows such collection to be generated in numerous ways by taking advantage of the mixins already implemented. By default, our implementation generates the seed collection through a pairwise testing process that considers both corner cases and extracted constants. Concerning the mutation level, our tool differentiates itself from the traditional ones in the sense that instead of mutating test cases randomly at the byte level, it delegates such task to the arbitraries involved in the testing process, which are responsible for implementing their own mutation logic. By doing so, we allow mutations to be always syntactically and semantically valid, which is something that does not happen in any other known coverage-guided solution.

A graphical representation of the coverage-guided workflow implemented by the `CoverageGuidance` mixin is illustrated in Figure 6.3.

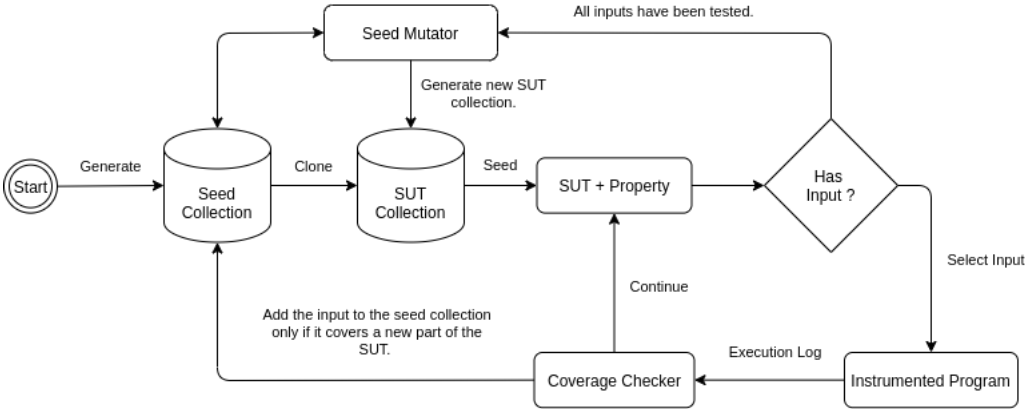


Figure 6.3: FluentCheck's coverage-guided strategy workflow.

With the implementation of the last section's mixin and the one just described in this section, we were able to extend even further the FluentCheck architecture, which now includes two new mixins and an entirely new collection of search strategies. An overview of our final architecture is presented in Figure 6.4.

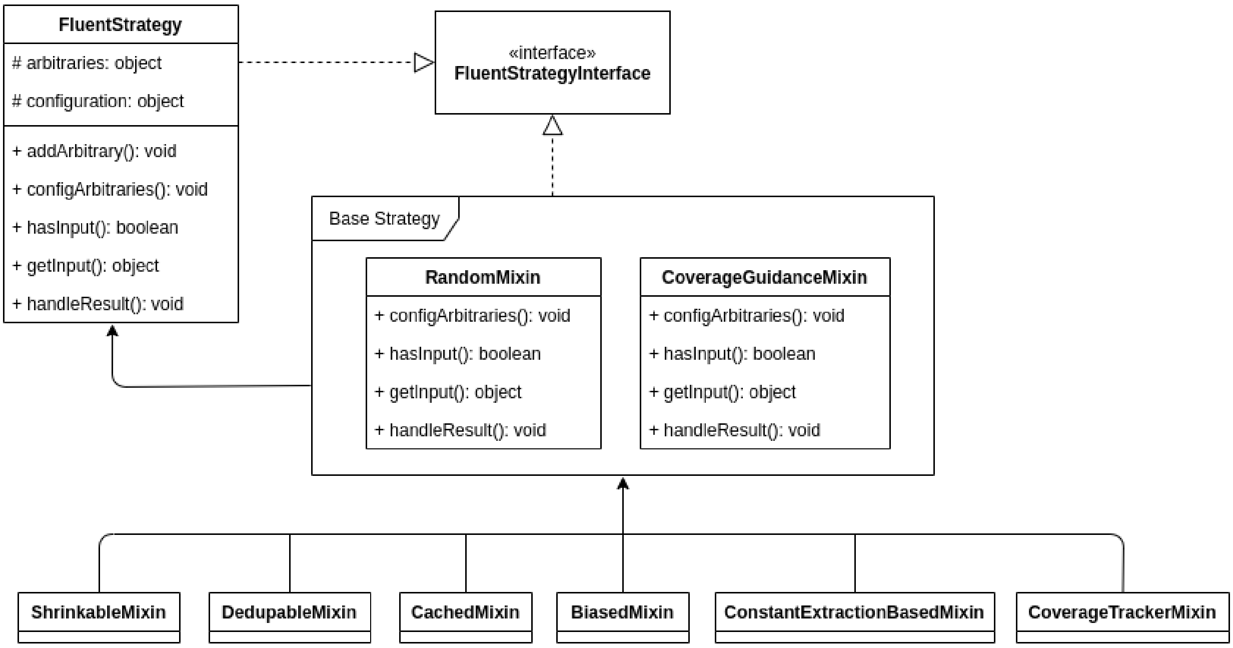


Figure 6.4: FluentCheck's search strategy final architecture.

6.3.5 Complementary Features

Apart from the implementation of the features described in the last sections, we also saw adequate to implement a set of complementary features, whose purpose, as the name itself suggests, is to complement FluentCheck's main features.

An important aspect concerning the implementation of the aforementioned complementary features is that most of them were not considered a mixin due to their lack of complexity. Instead, they were considered to be part of the `FluentStrategy` class configuration object, previously referenced in Section 6.3.1.

An overview of each of the implemented complementary features is presented in the following subsections.

6.3.5.1 Timeout

Given the possibility of having very time-consuming test specifications, we decided to implement a stop testing criterion based on a timeout. Therefore, and by calling the `withTimeout()` API method, a developer is able to specify, in milliseconds, the maximum amount of time for a given test specification.

6.3.5.2 Maximum Number of Test Cases

Based on the specification of the timeout, previously described in the last section, we also decided to implement another stop testing criterion based on the maximum number of test cases to be used. Therefore, and by calling the `withMaxNumberOfTestCases()` API method, a developer is able to specify the maximum number of test cases for a given test specification.

6.3.5.3 Minimum Coverage

With the implementation of the coverage-guided strategy, we also saw adequate to implement a stop testing criterion based on coverage. Therefore, and by calling the `withMinimumCoverage()` API method, a developer is able to specify the minimum coverage percentage that needs to be achieved before terminating the testing process.

6.3.5.4 Maximum Number of Mutations Per Arbitrary

Different from the previously mentioned complementary features, we also implemented a feature exclusively reserved to coverage-guided based strategies. Such feature is represented by the `withMaxNumMutationsPerArbitrary()` API method and can be used by a developer to specify how many mutations he desires to be performed on each input present in the seed collection.

6.3.5.5 Dynamic Sample Sizing

Implemented on the `DynamicSampleSizing` mixin class and explicitly directed for evaluation purposes, the dynamic sample sizing feature is responsible for enabling a righteous comparison between a search strategy with pairwise testing and another without, by ensuring that, regardless of the search strategy considered, the total number of test cases used by each strategy is always bounded by the number of test cases that would be used if pairwise testing was considered.

In order to better understand the dynamic sample sizing feature's behavior, let us consider a generic test specification involving three distinct arbitraries and FluentCheck's default sample size. On such test specification, a search strategy without pairwise testing will consider a total of 1 billion test cases $(1000 \times 1000 \times 1000 = 1000^3)$, while one with pairwise testing will consider a total of 1 million test cases $(1000 \times 1000 = 1000^2)$. With dynamic sample sizing, by directly manipulating the sample size value to 100, we enable the first search strategy to use the same number of test cases as the second one $(100 \times 100 \times 100 = 100^3 = 1000^2)$. Moreover, it worthy of mentioning that such manipulation is done based on the exponential increase associated with the number of test cases and on the formula underlying pairwise testing.

6.4 Summary

Aiming to implement this dissertation's proposed solution, we began with the process of selecting one of the numerous available PBT testing frameworks. FluentCheck was the framework selected.

The selection of such a tool was mainly based on the tool's embryonic development stage, which in turn was seen as the perfect opportunity in terms of enabling greater control over the entire framework architecture while implementing our solution. The support for both universal and existential quantifiers, coupled with the definition of a fluent API, compose the selected framework's most unique features.

According to the analysis made over FluentCheck's property validation process, it was found that such a process was heavily based on the recursivity implicitly implemented by the framework's architecture.

Concerning the strategies supported by the FluentCheck framework, we found out that such a tool was only capable of supporting one single random-based search strategy endowed with several hard-coded features. Sampling without replacement, biased sampling, caching, and shrinking compose all the before-mentioned features.

Striving to implement this chapter's proposed solution, an abstraction of the FluentCheck framework search strategy architecture was performed in order to endow such a tool with the ability to be easily extendable, and therefore, capable of supporting multiple search strategies. Such abstraction was made by following a hybrid approach that integrates the *Template Method* design pattern with a mixin-based approach.

Lastly, and to complement the aforementioned abstraction, three new strategies were implemented: one based on the extraction of numeric and string constants from source code; another responsible enabling pairwise testing; and ultimately, and entirely new, a coverage-guided strategy. Such strategies were complemented with several other features, which usage is attainable through the API precisely defined for configuring the search strategy to be used in a given test specification.

In the end, and with all the aforementioned details implemented, we were able to endow the FluentCheck framework with an easily extensible architecture capable of supporting multiple search strategies with a high degree of configuration (e.g., sampling without replacement, biased sampling, shrinking, etc). Moreover, we also ended up implementing three innovative and unprecedented search strategies, which were further complemented with various features.

Chapter 7

Evaluation

This chapter evaluates how the solution developed provides evidence towards the validity of this dissertation's main hypothesis. Section 7.1 presents an overview of the performed evaluation. Section 7.2 outlines the defined research questions that drove the evaluation and data collection processes. Moreover, this section also presents the search strategies and the evaluation metrics used in the evaluation process. Section 7.3 presents the benchmarks selection process and an overview of the selected benchmarks. Section 7.4 provides additional information about our experimental evaluation, including the benchmarks test specifications, the statistical analysis, and the setup used. Section 7.5 presents and discusses all the results achieved for each benchmark. Section 7.6 addresses the defined research questions and evaluates this dissertation's main hypothesis according to the results achieved. Section 7.7 reflects on the lessons learned during the evaluation process. Finally, Section 7.8 presents this chapter's validation threats.

7.1 Overview

Aiming to check to which extent we were able to validate this dissertation's central hypothesis (Section 4.2), we proceeded with the evaluation of our proposed solution, previously described in Chapter 6.

Accordingly, and given the nature of the aforementioned hypothesis, we decided to conduct such evaluation based on the statistical analysis of all the data extracted from the execution of several different search strategies over multiple benchmarks. Moreover, it is worthy of mentioning that such evaluation was conducted while only using the framework in which our proposed solution was implemented.

7.2 Research Questions

In any evaluation process, the research questions' definition is critical to guide the data collection process and its corresponding analysis. Therefore, we outlined the following quantitative research questions in order to identify whether our proposed solution validates, or not, the central hypothesis underlying this dissertation's development.

RQ1 *What is the impact, on **time**, **number of test cases**, and **satisfiability**, of using a coverage-guided search strategy in alternative to a random-based search strategy?*

RQ2 *What is the impact, on **time**, **number of test cases**, and **satisfiability**, of using both a coverage-guided and a random-based search strategy with biased sampling?*

RQ3 *What is the impact, on **time**, **number of test cases**, and **satisfiability**, of using both a coverage-guided and a random-based search strategy with pairwise testing?*

RQ4 *What is the impact, on **time**, **number of test cases**, and **satisfiability**, of using both a coverage-guided and a random-based search strategy with constants extraction?*

RQ5 *What is the impact, on **coverage**, of using a coverage-guided search strategy with biased sampling?*

RQ6 *What is the impact, on **coverage**, of using a coverage-guided search strategy with pairwise testing?*

RQ7 *What is the impact, on **coverage**, of using a coverage-guided search strategy with constants extraction?*

By considering the research questions mentioned above, we can easily assume that its definition would not be possible without first defining which search strategies to use and which metrics to extract during the evaluation process. As such, in the following subsections, we provide all the details concerning both the process of determining which search strategies to use and which metrics to extract.

7.2.1 Search Strategies Definition Process

While defining the search strategies to be used during the evaluation process, the main goal was to ensure that such strategies were not only taking advantage of the extensibility introduced by our proposed solution but also taking into account the newly implemented search strategies.

Therefore, and striving to define such search strategies, we decided to consider two types of base strategies: one that only includes random-based search strategies and another that only includes coverage-guided search strategies. Furthermore, we also decided to endow the beforementioned base strategies with a common default configuration that includes sampling without replacement, the caching mechanism, and dynamic sample sizing. It is worth mentioning that shrinking was not considered part of such configuration because this behavior was seen as a best-effort, nice-to-have feature of PBT, which does not actively contribute to our evaluation phase.

Apart from the behaviors specified above, three more possible behaviors could also be considered. Biased sampling, pairwise testing, and constants extraction compose the remaining possible behaviors. In the interest of taking full advantage of the extensibility introduced by our proposed solution, we decided to consider all the possible combinations of such behaviors, which led us to the definition of a total of 16 different search strategies.

Table 7.1 outlines all the defined search strategies used in our evaluation phase and their corresponding configuration.

Table 7.1: Evaluation phase search strategies' configuration. Check marks (✓) mean that the respective feature **is** included, hyphens (-) mean that the respective feature **is not** included.

I/E	Identifier	Biased Sampling	Constants Extraction	Pairwise Testing
Random	**PBT_R_S1**	-	-	-
	PBT_R_S2	✓	-	-
	PBT_R_S3	-	✓	-
	PBT_R_S4	-	-	✓
	PBT_R_S5	✓	✓	-
	PBT_R_S6	✓	-	✓
	PBT_R_S7	-	✓	✓
	PBT_R_S8	✓	✓	✓
Coverage-Guided	**PBT_CG_S1**	-	-	-
	PBT_CG_S2	✓	-	-
	PBT_CG_S3	-	✓	-
	PBT_CG_S4	-	-	✓
	PBT_CG_S5	✓	✓	-
	PBT_CG_S6	✓	-	✓
	PBT_CG_S7	-	✓	✓
	PBT_CG_S8	✓	✓	✓

7.2.2 Evaluation Metrics Definition Process

Based on the similarity of our proposed solution with the ones identified in Chapter 3, we decided to define the evaluation metrics by following a systematic approach that consisted of analyzing the most frequently extracted metrics among all the publications that compose this dissertation's state of the art. Figure 7.1 presents a graphical representation of the results of the before-mentioned systematic approach.

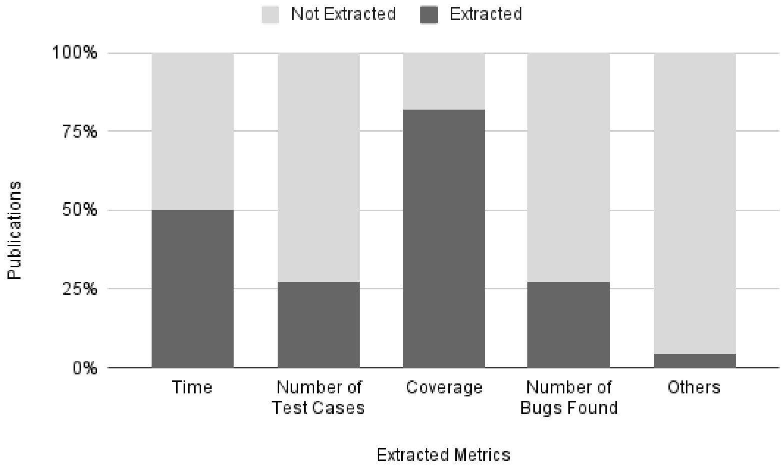


Figure 7.1: Most frequent extracted evaluation metrics among all the publications considered in this dissertation's state of the art.

By analyzing the graph's results, we could conclude that the *Time*, *Number of Test Cases*, *Coverage*, and *Number of Bugs Found* compose the set of the most frequently extracted metrics among all the aforementioned publications. Moreover, we could also notice that other metrics were also extracted; nevertheless, each of those metrics was only associated with a specific publication, hence the lower percentage exhibited by the *Others* column.

Thus, and based not only on the graph's results just described but also on the nature of our proposed solution and correlated hypothesis, we decided to consider the following metrics in our evaluation phase:

- **Time**: The amount of time spent while checking a given property's satisfiability.
- **Number of Test Cases**: The number of test cases used while checking a given property's satisfiability.
- **Coverage**: The final code coverage percentage achieved while checking a given property's satisfiability.
- **Satisfiability**: Binary metric that indicates whether a given property is satisfied or not. Chosen alternatively to the *Number of Bugs Found* metric by considering the nature of our experimental evaluation (Section 7.4) and corresponding benchmarks (Section 7.3).

7.3 Benchmarks

Having defined the research questions outlined in Section 7.2, we proceeded with the definition of the benchmarks to be used as the foundation of our evaluation experiments. As such, in the following subsections, we provide a detailed description of the benchmark selection process and an overview of each of the selected benchmarks.

7.3.1 Benchmarks Selection Process

While selecting the benchmarks to be used during the evaluation process, the primary goal was to ensure that, through the selected benchmarks, we could collect trustworthy and representative information of our solution's true potential.

Aiming to fulfill the before-mentioned goal, and based on the similarity of our proposed solution with the ones identified in Chapter 3, we decided to conduct a systematic process that consisted of selecting the most promising benchmarks used by the publications that compose this dissertation's state of start. Such a process is depicted in Figure 7.2.

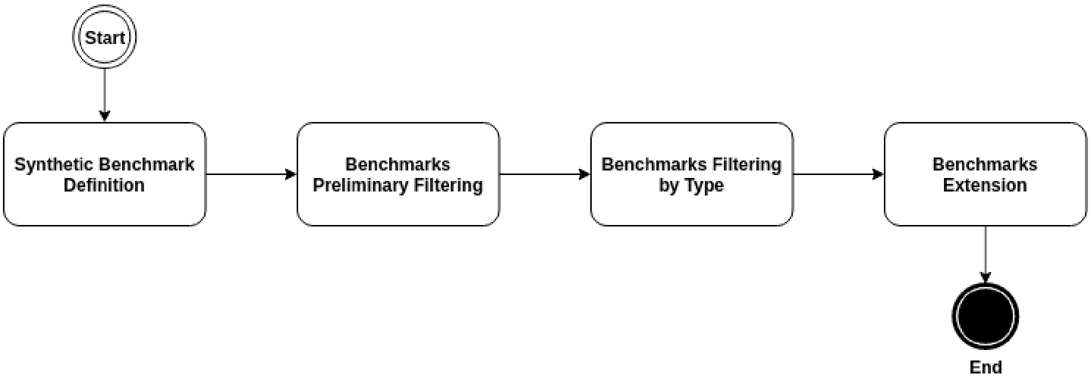

Start
Synthetic Benchmark Definition
Benchmarks Preliminary Filtering
Benchmarks Filtering by Type
Benchmarks Extension
End

Figure 7.2: Overview of the Benchmark Selection Process.

The process presented in Figure 7.2 comprises, in total, a sequence of 4 steps. Each step has its own definition and purpose, which are described below:

1. **Synthetic Benchmark Definition**: Given the absence of control underlying the systematic process, in terms of the selected benchmarks being able to make the most out of the search strategies defined in Section 7.2.1, we decided to initialize such process by specifically designing a synthetic benchmark capable of exercising all the behaviors implicitly and explicitly incorporated by the before-mentioned search strategies.

2. **Benchmarks Preliminary Filtering**: With the last step fully achieved, we shifted our focus to the selection of new benchmarks while considering all the publications that compose this dissertation's state of start. After a preliminary overview of all publications' benchmarks, we immediately filtered the most promising ones by excluding those whose replication on the FluentCheck framework would be nearly impossible given their complexity and base

programming language. In the end, from a total of 23 publications, only the benchmarks of 3 of them were considered for further analysis. Table A.1 comprises the data concerning this preliminary phase.

3. **Benchmarks Filtering by Type**: By further analyzing the most promising benchmarks resulting from the last step, we identified several different types of benchmarks. *Research*, *Artificial*, *Open-Source*, *Online Program*, *Real-World*, *Challenge*, and *Academic* compose the set of types identified. Striving to select at least one benchmark from all the beforementioned types, and based on the characteristics of each benchmark, we ended up selecting a total of 4 benchmarks. Table A.2 portrays all the details concerning this step's filtering process.

4. **Benchmarks Extension**: Given the implicit high rate of exclusion mentioned in the *Benchmarks Preliminary Filtering* step, we decided to extend the benchmark selection process even further by considering the benchmarks used by the **fast-check** and **JQF** frameworks, which compose the most viable alternatives to FluentCheck. From the set of benchmarks found, we only decided to consider one benchmark, which corresponds to an implementation of the *Stack* data structure in JavaScript; however, many others could also be selected, but we considered to have, at this point, an adequate number of distinct benchmarks.

By the end of the aforementioned systematic process, a total of 6 benchmarks were selected. Such benchmarks are presented in Table 7.2.

Table 7.2: Evaluation phase selected benchmarks.

Benchmark	**Type**
`Defective Calculator`	Artificial / Synthetic
`JS Algorithms - Stack`	Open-Source
`Multiplication Sign`	Academic
`RERS Challenge 2012`	Challenge
`Triangle Classification Program`	Research
`Zodiac`	Online Program

7.3.2 Benchmarks Overview

This section aims to provide an overview of each of the selected benchmarks. As such, each benchmark is synthetically described in the following subsections.

7.3.2.1 Defective Calculator

The `Defective Calculator` benchmark composes the synthetic benchmark specifically designed to exercise all the behaviors incorporated by the evaluation phase's search strategies. Moreover, such benchmark consists of a calculator capable of performing the following basic arithmetic operations: *Addition*, *Subtraction*, *Multiplication*, and *Division*. However, this section's benchmark endows each of those operations with a series of predicates responsible for modifying those operations' default behaviors.

7.3.2.2 JS Algorithms - Stack

This section's benchmark consists of an implementation of the `Stack` data structure in JavaScript. Such implementation is based on the implementation of the `Linked List`[1] data structure and is composed of six methods: `isEmpty()`, `peek()`, `push()`, `pop()`, `toArray()`, and `toString()`.

7.3.2.3 Multiplication Sign

The `Multiplication Sign` benchmark consists of a simple academic program responsible for returning the sign resulting from the multiplication of three different numbers.

7.3.2.4 RERS Challenge 2012

This section's benchmark consists of a complex problem in which a set of underlying reachability and behavioral properties must be satisfied. Moreover, such benchmark composes the most complex benchmark used during the entire evaluation phase, given its type — *Challenge* — and the fact that it includes a considerable amount of complex predicates.

7.3.2.5 Triangle Classification Program

The `Triangle Classification Program` benchmark consists of a program responsible for classifying a triangle according to its sides' length. Additionally, this section's benchmark composes one of most used benchmarks in the software testing domain due to its interesting control flow.

7.3.2.6 Zodiac

The `Zodiac` benchmark consists of a simple online program responsible for determining the zodiac sign from a given month and day. Even though such a benchmark can be considered simple, it includes an interesting series of predicates.

7.4 Experimental Evaluation

The current section aims to provide additional information about our experimental evaluation by presenting, in the following subsections, a detailed description of the benchmarks' test specifications, the evaluation metrics' statistical analysis, and the setup used for performing such evaluation.

[1] Available: https://github.com/trekhleb/javascript-algorithms

7.4.1 Benchmarks Test Specifications

Considering the benchmarks selected in Section 7.3, and given the nature of our proposed solution's domain, we had to define for each benchmark a set of test specifications in order to be able to apply the defined search strategies (Section 7.2.1) and consequently extract the specified evaluation metrics (Section 7.2.2). However, given the nature of the selected benchmarks, we had two define two different types of test specifications. As such, the following subsections provide a detailed description of each type of the test specifications defined.

7.4.1.1 Satisfiability-Oriented Test Specifications

The current section's test specifications aim to prove that a set of specific properties inherent to a given benchmark are indeed satisfied. It is worthy of mentioning that such properties do not describe a given benchmark as a whole. Instead, they express specific reachability and behavioral properties. `Defective Calculator` and `RERS Challenge 2012` compose the benchmarks that motivated this section's test specifications and in which they are employed.

7.4.1.2 Integrity-Oriented Test Specifications

Contrarily to the test specifications described in the previous section, this section's test specifications aim to verify the integrity of a given benchmark as a whole. By doing so, such test specifications can be used to determine whether an entire system behaves as expected or not.

However, for this section's test specifications to actively contribute to our evaluation phase, the benchmarks underlying such specifications must be endowed with any sort of bug capable of disrupting the typical behavior of a given benchmark. Only by doing so can we appropriately compare the considered search strategies and their corresponding evaluation metrics.

In that sense, and considering the witnessed absence of bugs in the benchmarks in which this section's test specifications are employed, we saw ourselves forced to insert random bugs into those same benchmarks. Aiming to introduce such bugs while not compromising the whole evaluation purpose's primary goal, which is to be the most systematic and trustworthy as possible, we recurred to the StrykerJS[2] mutation testing tool.

By using the aforementioned mutation testing tool, we were able to randomly generate mutations for each of the benchmarks that were not considered in the last section's test specifications. Furthermore, such mutations were further properly filtered by excluding all the equivalent mutants. From the resulting filtered mutations — each representing a unique bug — we randomly selected five different mutations in order to create five unique variants of each benchmark. Then, by running the test specifications over each benchmark variant instead of the benchmark itself, we could adequately compare the selected search strategies and their impact on the considered evaluation metrics.

[2] Available: https://stryker-mutator.io/docs/stryker-js/introduction

7.4.2 Statistical Analysis

Striving to determine the impact of the defined search strategies (Section 7.2.1) on the considered evaluation metrics (Section 7.2.2), we decided to conduct a statistical analysis of all the evaluation metrics extracted from each of the selected benchmarks (Section 7.3) and corresponding variants. As such, in the following subsections we present an overview of the ***Analysis of Variance Test*** [50] and ***Tukey's Test*** [18], which compose the foundation of our statistical analysis.

7.4.2.1 Analysis of Variance Test

Based on the nature of this chapter's underlying research questions, we can infer that the answer to each of the defined research questions consists of determining the impact of two dependent variables on a single independent variable, being the dependent variables the presence and absence of a particular behavior (e.g., biased sampling), and the independent variable an evaluation metric (e.g., time).

Therefore, in order to determine the aforementioned impact for each of the defined research questions, we decided to execute the Analysis of Variance (ANOVA) test, more precisely the One-way ANOVA, which consists of a statistical test used to determine whether or not there is a statistically significant difference between the means of two different groups. Aiming to perform the before-mentioned test, we defined the following hypotheses:

- **Null Hypothesis** (H_0) – There is no significant difference among the groups. Rejected if p-value < 0.05.
- **Alternate Hypothesis** (H_a) – There is a significant difference among the groups.

Even though the ANOVA test provides information about whether or not there is a significant statistical difference between two groups, it does not provide deeper insights concerning which groups are different from each other. In that sense, we complemented this test with the post hoc test described in the following section.

7.4.2.2 Tukey's Test

Tukey's test is a post hoc test, also known as a multiple comparison test, that, complementary to the one defined in the last section, provides information on which groups are different from each other by making pairwise comparisons between the means of two or more groups.

Considering this section's test potential, we decided to apply it throughout all the extracted metrics in order to be able to answer the defined research questions. In fact, all the results presented in Section 7.5 correspond to the results provided by the execution of this section's test.

Aiming to allow a better understanding of the results presented in Section 7.5, let us consider the meaning behind each of the columns represented in Table 7.3, which depicts a possible Tukey test result.

Table 7.3: Example of Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-0.0166	0.9000	-5.0733	5.0401	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-0.0652	0.9000	-5.1215	4.9911	False
Random withConstantExtraction	Random withoutConstantExtraction	-5.0612	0.0010	-5.1558	-4.9666	True
Coverage-Guided withBias	Coverage-Guided withoutBias	-4.2194	0.5943	-22.0413	13.6026	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	11.0656	0.1131	-3.5357	25.6669	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-12.3356	0.0675	-25.8833	1.2121	False
Coverage-Guided	Random	-186.8168	0.0010	-194.5256	-179.1079	True

- ***Group1***: This column corresponds to a given search strategy that **contains** a particular feature.

- ***Group2***: This column corresponds to a given search strategy that does **not contain** a particular feature.

- ***Mean Diff***: Determines which group performs better for a given evaluation metric. $MeanDiff < 0$ means that *Group1* has higher absolute values than *Group2*; $MeanDiff > 0$ means the opposite. Therefore, by analyzing this column's associated values in the table above, it is possible to conclude that the usage of any of the developed search strategy features has a negative impact on the *Time* evaluation metric since it implies higher time values.

- ***P-Adj***: Corresponds to the ***p-value*** [20] used for evaluating the underlying test hypotheses.

- ***Lower and Upper***: Define, respectively, the lower and upper confidence interval limits for pairwise mean differences.

- ***Reject***: Indicates whether the **null hypothesis** has been rejected (*True*) or not (*False*). If *True*, then there is a significant statistical difference between *Group1* and *Group2*.

7.4.3 Setup

The experimental evaluation was performed on Ubuntu 20.04 64-bit operating system with 16.0 GB RAM and Intel Core i7-8565U CPU processor @ 4.6GHz speed. The Typescript version was 4.2.3, Mocha at version 8.2.2, and Python at version 3.9.5.

Moreover, the experimental evaluation was also performed by following a 4x3 approach that consisted of testing each benchmark variant four different times while executing three times each benchmark's associated test specifications for each considered search strategy. Only by doing so could we collect reliable data while considering the lack of precision of the CPU while measuring times in the order of milliseconds. It is also worthy of mentioning that, during the experimental evaluation, the FluentCheck default configuration sample size (1000) and number of mutations per arbitrary (5) values were used, which, coupled with the use of dynamic sample sizing, limited the number of test cases used by each search strategy to a maximum of one million test cases.

7.5 Results and Discussion

The current section aims to provide a detailed overview of our evaluation phase's experimental results. As such, the following subsections present and discuss the experimental results for each of the selected benchmarks. Furthermore, it is worth mentioning that only the results that revealed significant statistical differences — Tukey's test *Reject* column's values set to *True* — are presented in this section.

7.5.1 Defective Calculator

The `Defective Calculator` benchmark's results — Appendix B — revealed significant statistical differences in the following evaluation metrics: ***Time***, ***Number of Test Cases***, and ***Satisfiability***. The following subsections present and discuss the results concerning each of those metrics.

7.5.1.1 Time

Table 7.4 presents the Tukey test's results associated with the impact of the considered search strategies on the *Time* evaluation metric. By analyzing the before-mentioned results, it is possible to conclude the following:

- From all the possible features, only the *constants extraction* feature revealed significant statistical differences. Such differences are exhibited by all the considered properties.

- The *constants extraction* feature usage on both random-based and coverage-guided search strategies decreases the time required to satisfy all the considered properties.

Table 7.4: Tukey's test results concerning the *Time* evaluation metric for the *Defective Calculator* benchmark.

Mutation	Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
P1	Random withConstantExtraction	Random withoutConstantExtraction	1867.6894	0.001	1834.6321	1900.7467	True
P1	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	3112.9687	0.001	3076.0032	3149.9343	True
P2	Random withConstantExtraction	Random withoutConstantExtraction	1998.1144	0.001	1975.3796	2020.8492	True
P2	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	3688.9906	0.001	3408.9145	3969.0668	True
P3	Random withConstantExtraction	Random withoutConstantExtraction	1993.6062	0.001	1959.2709	2027.9416	True
P3	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	3222.9987	0.001	3170.1929	3275.8046	True
P4	Random withConstantExtraction	Random withoutConstantExtraction	2052.4800	0.001	2011.3730	2093.5870	True
P4	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	3243.4550	0.001	3155.6907	3331.2193	True
P5	Random withConstantExtraction	Random withoutConstantExtraction	2065.1700	0.001	2014.4966	2115.8434	True
P5	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	3304.3556	0.001	3227.2990	3381.4123	True
P6	Random withConstantExtraction	Random withoutConstantExtraction	2012.0731	0.001	1977.0830	2047.0632	True
P6	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	3308.0156	0.001	3223.5476	3392.4836	True
P7	Random withConstantExtraction	Random withoutConstantExtraction	1953.5125	0.001	1914.7681	1992.2569	True
P7	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	3532.9463	0.001	3200.4516	3865.4409	True
P8	Random withConstantExtraction	Random withoutConstantExtraction	2046.5638	0.001	2030.9810	2062.1465	True
P8	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	3177.9694	0.001	3168.8306	3187.1081	True

7.5.1.2 Number of Test Cases

Table 7.5 presents the Tukey test's results associated with the impact of the considered search strategies on the *Number of Test Cases* evaluation metric. By analyzing the before-mentioned results, it is possible to conclude the following:

- The coverage-guided search strategies require fewer test cases than the random-based search strategies to satisfy this section benchmark's properties. Such a conclusion is verified for all the considered properties.
- From all the possible features, only the *constants extraction* feature revealed significant statistical differences. Such differences are exhibited by all the considered properties but only by coverage-guided search strategies.
- The *constants extraction* feature usage on coverage-guided search strategies decreases the number of test cases required to satisfy all the considered properties.

Table 7.5: Tukey's test results concerning the *Number of Test Cases* evaluation metric for the *Defective Calculator* benchmark.

Mutation	**Group1**	**Group2**	**Mean Diff**	**P-Adj**	**Lower**	**Upper**	**Reject**
P1	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	127920.750	0.0010	126055.6345	129785.8655	True
P1	Coverage-Guided	Random	436038.625	0.0382	27409.2752	844667.9748	True
P2	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	146478.250	0.0010	135946.3844	157010.1156	True
P2	Coverage-Guided	Random	427257.375	0.0417	18388.1005	836126.6495	True
P3	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	126809.000	0.0010	125679.4650	127938.5350	True
P3	Coverage-Guided	Random	436594.000	0.0379	28022.4296	845165.5704	True
P4	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	126529.500	0.0010	124721.7083	128337.2917	True
P4	Coverage-Guided	Random	436733.750	0.0379	28176.0146	845291.4854	True
P5	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	129058.000	0.0010	125972.4307	132143.5693	True
P5	Coverage-Guided	Random	435448.000	0.0384	26771.5516	844124.4484	True
P6	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	129832.000	0.0010	126965.7959	132698.2041	True
P6	Coverage-Guided	Random	435061.000	0.0386	26344.5396	843777.4604	True
P7	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	141404.250	0.0010	126585.0127	156223.4873	True
P7	Coverage-Guided	Random	437797.375	0.0318	44038.8563	831555.8937	True
P8	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	124784.250	0.0010	124765.8202	124802.6798	True
P8	Coverage-Guided	Random	437372.375	0.0376	28914.7557	845829.9943	True

7.5.1.3 Satisfiability

Table 7.6 presents the Tukey test's results associated with the impact of the considered search strategies on the *Satisfiability* evaluation metric. By analyzing the before-mentioned results, it is possible to conclude the following:

- The coverage-guided search strategies present higher satisfiability rates than the random-based search strategies. Such a conclusion is verified for all the considered properties.

7.5.1.4 Conclusions

According to this section benchmark's results, we can conclude that a coverage-guided search strategy endowed with the *constants extraction* feature composes the ideal search strategy for the

Table 7.6: Tukey's test results concerning the *Satisfiability* evaluation metric for the *Defective Calculator* benchmark.

Mutation	**Group1**	**Group2**	**Mean Diff**	**P-Adj**	**Lower**	**Upper**	**Reject**
P1	Coverage-Guided	Random	-50.0	0.0192	-90.5327	-9.4673	True
P2	Coverage-Guided	Random	-50.0	0.0192	-90.5327	-9.4673	True
P3	Coverage-Guided	Random	-50.0	0.0192	-90.5327	-9.4673	True
P4	Coverage-Guided	Random	-50.0	0.0192	-90.5327	-9.4673	True
P5	Coverage-Guided	Random	-50.0	0.0192	-90.5327	-9.4673	True
P6	Coverage-Guided	Random	-50.0	0.0192	-90.5327	-9.4673	True
P7	Coverage-Guided	Random	-50.0	0.0192	-90.5327	-9.4673	True
P8	Coverage-Guided	Random	-50.0	0.0192	-90.5327	-9.4673	True

`Defective Calculator` benchmark. Such a conclusion is based on the higher satisfiability rates achieved by coverage-guided search strategies and on the decrease in time and number of test cases caused by the *constants extraction* feature usage.

Moreover, it is worth mentioning that the aforementioned conclusion aligns with what was already expected, considering the heavy presence of constants, and their direct influence, on the satisfiability-oriented test specifications associated with this section's benchmark.

7.5.2 JS Algorithms - Stack

The `JS Algorithms - Stack` benchmark's results — Appendix C — revealed significant statistical differences only on the ***Time*** evaluation metric. The following subsection presents and discusses the results concerning the before-mentioned metric.

7.5.2.1 Time

Table 7.7 presents the Tukey test's results associated with the impact of the considered search strategies on the *Time* evaluation metric. By analyzing the before-mentioned results, it is possible to conclude the following:

- The coverage-guided search strategies require more time than the random-based search strategies to find bugs in this section benchmark's variants. Such a conclusion is verified for all the considered benchmark variants.

- From all the possible features, only the *constants extraction* feature revealed significant statistical differences. Such differences are exhibited by all benchmark variants for the random-based search strategies and only by the benchmark variant **M4** for the coverage-guided search strategies.

- The *constants extraction* feature usage on both coverage-guided and random-based search strategies increases the time required to find bugs on the corresponding benchmark variants.

Table 7.7: Tukey's test results concerning the *Time* evaluation metric for the *JS Algorithms - Stack* benchmark.

Mutation	Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
M0	Random withConstantExtraction	Random withoutConstantExtraction	-1.0036	0.0010	-1.2063	-0.8009	True
M0	Coverage-Guided	Random	-205.7254	0.0010	-207.5752	-203.8756	True
M1	Random withConstantExtraction	Random withoutConstantExtraction	-1.1504	0.0016	-1.6657	-0.6352	True
M1	Coverage-Guided	Random	-205.5731	0.0010	-208.0666	-203.0796	True
M2	Random withConstantExtraction	Random withoutConstantExtraction	-1.1759	0.0010	-1.4621	-0.8898	True
M2	Coverage-Guided	Random	-203.6985	0.0010	-204.6794	-202.7177	True
M3	Random withConstantExtraction	Random withoutConstantExtraction	-1.0439	0.0010	-1.2597	-0.8282	True
M3	Coverage-Guided	Random	-206.7135	0.0010	-207.8506	-205.5765	True
M4	Random withConstantExtraction	Random withoutConstantExtraction	-1.1731	0.0037	-1.7990	-0.5473	True
M4	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-2.7138	0.0263	-4.9795	-0.4480	True
M4	Coverage-Guided	Random	-206.8249	0.0010	-208.3565	-205.2934	True

7.5.2.2 Conclusions

According to this section benchmark's results, we can conclude that a random-based search strategy not provided with the *constants extraction* feature composes the ideal search strategy for the `JS Algorithms - Stack` benchmark. Such a conclusion is based on the decrease in time revealed by the random-based search strategies usage, and on the increase in time caused by the *constants extraction* feature usage.

Furthermore, it is noteworthy that the before-mentioned conclusion aligns with what was already expected, considering that the constants present in the benchmark's source code do not directly influence the integrity-oriented test specifications associated with this section's benchmark. In fact, their extraction only introduces an overhead in terms of the time required to find bugs.

7.5.3 Multiplication Sign

The `Multiplication Sign` benchmark's results — Appendix D — revealed significant statistical differences in the following evaluation metrics: ***Time***, ***Number of Test Cases***, and ***Coverage***. The following subsections present and discuss the results concerning each of those metrics.

7.5.3.1 Time

Table 7.8 presents the Tukey test's results associated with the impact of the considered search strategies on the *Time* evaluation metric. By analyzing the before-mentioned results, it is possible to conclude the following:

- The coverage-guided search strategies require less time than the random-based search strategies to find bugs in this section benchmark's **M0** and **M3** variants. The opposite is verified for the remaining benchmark variants (**M1**, **M2**, and **M4**).

- From all the possible features, only the *pairwise testing* and the *constants extraction* features revealed significant statistical differences. Such differences are exhibited by the benchmark variants **M0**, **M3**, and **M4** for the *pairwise testing* feature and only by the benchmark variant **M1** for the *constants extraction* feature.

- The *pairwise feature* usage on both coverage-guided and random-based search strategies increases the time required to find bugs on the corresponding benchmark variants.

- The *constants extraction* feature usage on random-based search strategies increases the time required to find bugs on the corresponding benchmark variant.

Table 7.8: Tukey's test results concerning the *Time* evaluation metric for the *Multiplication Sign* benchmark.

Mutation	Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
M0	Random withPairWiseTesting	Random withoutPairWiseTesting	-1021.4963	0.0010	-1067.0433	-975.9492	True
M0	Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-95.7550	0.0010	-104.0085	-87.5015	True
M0	Coverage-Guided	Random	811.0581	0.0010	394.7795	1227.3368	True
M1	Random withConstantExtraction	Random withoutConstantExtraction	-4.3917	0.0441	-8.6217	-0.1617	True
M1	Coverage-Guided	Random	-131.2696	0.0010	-135.3519	-127.1873	True
M2	Coverage-Guided	Random	-131.5273	0.0010	-136.9074	-126.1472	True
M3	Random withPairWiseTesting	Random withoutPairWiseTesting	-1064.0544	0.0010	-1106.5401	-1021.5687	True
M3	Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-88.9181	0.0010	-99.6804	-78.1558	True
M3	Coverage-Guided	Random	849.9438	0.0010	416.7858	1283.1017	True
M4	Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-9.0044	0.0403	-17.4587	-0.5500	True
M4	Coverage-Guided	Random	-136.8012	0.0010	-142.3077	-131.2946	True

7.5.3.2 Number of Test Cases

Table 7.9 presents the Tukey test's results associated with the impact of the considered search strategies on the *Number of Test Cases* evaluation metric. By analyzing the before-mentioned results, it is possible to conclude the following:

- From all the possible features, only the *constants extraction* feature revealed significant statistical differences. However, such differences are only exhibited by the benchmark variant **M4**.

- The *constants extraction* feature usage on both coverage-guided and random-based search strategies increases the number of test cases required to find bugs on the corresponding benchmark variant.

Table 7.9: Tukey's test results concerning the *Number of Test Cases* evaluation metric for the *Multiplication Sign* benchmark.

Mutation	Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
M4	Random withConstantExtraction	Random withoutConstantExtraction	-1.25	0.0025	-1.8617	-0.6383	True
M4	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-1.50	0.0020	-2.2064	-0.7936	True

7.5.3.3 Coverage

Table 7.10 presents the Tukey test's results associated with the impact of the considered search strategies on the *Coverage* evaluation metric. By analyzing the before-mentioned results, it is possible to conclude the following:

- From all the possible features, only the *constants extraction* feature revealed significant statistical differences. However, such differences are only exhibited by the benchmark variant **M4** for the coverage-guided search strategies.
- The *constants extraction* feature usage on coverage-guided search strategies increases the coverage achieved while finding bugs on the corresponding benchmark variant.

Table 7.10: Tukey's test results concerning the *Coverage* evaluation metric for the *Multiplication Sign* benchmark.

Mutation	Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
M4	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-4.5925	0.001	-5.671	-3.514	True

7.5.3.4 Conclusions

According to this section benchmark's results, we can conclude that a base search strategy not provided with the *pairwise testing* feature composes the ideal search strategy for the `Multiplication Sign` benchmark. Such a conclusion is based on the increase in time and the number of test cases caused by the *pairwise testing* feature usage.

Moreover, it is worth mentioning that the aforementioned conclusion partially aligns with what was already expected, considering that, while using the *pairwise testing* feature, we were not only expecting the introduction of an overhead in time, given the complex operations underlying such feature, but we were also expecting a positive impact on the number of test cases used to find bugs, which based on the results just presented did not happen. However, after further analysis, we concluded that the only reason our expectations were not fully met was due to the lack of complexity underlying this section's benchmark and corresponding variants.

7.5.4 RERS Challenge 2012

The `RERS Challenge 2012` benchmark's results — Appendix E — revealed significant statistical differences in the following evaluation metrics: ***Time***, ***Number of Test Cases***, and ***Satisfiability***. The following subsections present and discuss the results concerning each of those metrics.

7.5.4.1 Time

Table 7.11 presents the Tukey test's results associated with the impact of the considered search strategies on the *Time* evaluation metric. By analyzing the before-mentioned results, it is possible to conclude the following:

- The coverage-guided search strategies require more time than the random-based search strategies to satisfy this section benchmark's properties. Such a conclusion is verified for all the considered properties.

- From all the possible features, only the *constants extraction* and the *pairwise testing* features revealed significant statistical differences. Such differences are exhibited by almost all the considered properties for the *constants extraction* feature, but only by the property **P13** for the *pairwise testing* feature.

- The *constants extraction* feature usage on both base search strategies, but mainly on the random-based search strategies, increases the time required to satisfy all the considered properties.

- The *pairwise testing* feature usage on random-based search strategies increases the time required to satisfy the corresponding property.

Table 7.11: Tukey's test results concerning the *Time* evaluation metric for the *RERS Challenge 2012* benchmark.

Mutation	**Group1**	**Group2**	**Mean Diff**	**P-Adj**	**Lower**	**Upper**	**Reject**
P1	Random withConstantExtraction	Random withoutConstantExtraction	-78.7656	0.0010	-90.6664	-66.8649	True
P1	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-83.7562	0.0050	-131.2807	-36.2318	True
P1	Coverage-Guided	Random	-1604.3791	0.0010	-1655.0439	-1553.7142	True
P2	Random withConstantExtraction	Random withoutConstantExtraction	-80.5312	0.0010	-99.7665	-61.2960	True
P2	Coverage-Guided	Random	-1627.9050	0.0010	-1708.8960	-1546.9140	True
P3	Random withConstantExtraction	Random withoutConstantExtraction	-55.1800	0.0010	-65.1018	-45.2582	True
P3	Coverage-Guided	Random	-1416.7050	0.0010	-1542.2831	-1291.1269	True
P4	Random withConstantExtraction	Random withoutConstantExtraction	-67.2206	0.0010	-80.9274	-53.5139	True
P4	Coverage-Guided	Random	-1565.6866	0.0010	-1829.5061	-1301.8670	True
P5	Random withConstantExtraction	Random withoutConstantExtraction	-58.1419	0.0010	-74.8952	-41.3886	True
P5	Coverage-Guided	Random	-1449.9447	0.0010	-1554.3206	-1345.5688	True
P6	Random withConstantExtraction	Random withoutConstantExtraction	-94.6006	0.0010	-108.6612	-80.5401	True
P6	Coverage-Guided	Random	-1445.0522	0.0010	-1614.0465	-1276.0579	True
P7	Random withConstantExtraction	Random withoutConstantExtraction	-66.2225	0.0074	-107.1340	-25.3110	True
P7	Coverage-Guided	Random	-5694.5256	0.0010	-7410.7365	-3978.3148	True
P8	Random withConstantExtraction	Random withoutConstantExtraction	-100.4294	0.0010	-137.8617	-62.9971	True
P8	Coverage-Guided	Random	-6963.0553	0.0010	-9057.6169	-4868.4937	True
P9	Random withConstantExtraction	Random withoutConstantExtraction	-87.7475	0.0116	-147.6120	-27.8830	True
P9	Coverage-Guided	Random	-4976.8125	0.0010	-6539.9368	-3413.6882	True
P10	Coverage-Guided	Random	-5170.0772	0.0010	-7091.6992	-3248.4552	True
P11	Random withConstantExtraction	Random withoutConstantExtraction	-103.7900	0.0101	-172.5208	-35.0592	True
P11	Coverage-Guided	Random	-20515.9656	0.0010	-22941.6497	-18090.2816	True
P12	Coverage-Guided	Random	-16512.6703	0.0010	-20261.4399	-12763.9007	True
P13	Random withPairWiseTesting	Random withoutPairWiseTesting	-72.8106	0.0391	-140.5811	-5.0401	True
P13	Coverage-Guided	Random	-20091.9997	0.0010	-22473.3934	-17710.6060	True
P14	Coverage-Guided	Random	-26838.5022	0.0010	-26935.6240	-26741.3804	True

7.5.4.2 Number of Test Cases

Table 7.12 presents the Tukey test's results associated with the impact of the considered search strategies on the *Number of Test Cases* evaluation metric. By analyzing the before-mentioned results, it is possible to conclude the following:

- The coverage-guided search strategies require more test cases than the random-based search strategies to satisfy this section benchmark's properties. Such a conclusion is verified for the following properties: **P8**, **P11**, and **P13**.

- From all the possible features, only the *constants extraction* and the *pairwise testing* features revealed significant statistical differences. Such differences are exhibited by the property **P6** for the *constants extraction* feature, and by the property **P3** for the *pairwise testing* feature.

- The *constants extraction* feature usage on coverage-guided search strategies decreases the number of test cases required to satisfy the corresponding property.
- The *pairwise testing* feature usage on random-based search strategies decreases the number of test cases required to satisfy the corresponding property.

Table 7.12: Tukey's test results concerning the *Number of Test Cases* evaluation metric for the *RERS Challenge 2012* benchmark.

Mutation	Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
P3	Random withPairWiseTesting	Random withoutPairWiseTesting	792.000	0.0473	13.1957	1570.8043	True
P6	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	1188.500	0.0490	7.3017	2369.6983	True
P8	Coverage-Guided	Random	-15711.250	0.0024	-24853.4916	-6569.0084	True
P11	Coverage-Guided	Random	-30691.875	0.0010	-43102.1009	-18281.6491	True
P13	Coverage-Guided	Random	-26427.750	0.0094	-45262.2614	-7593.2386	True

7.5.4.3 Satisfiability

Table 7.13 presents the Tukey test's results associated with the impact of the considered search strategies on the *Satisfiability* evaluation metric. By analyzing the before-mentioned results, it is possible to conclude the following:

- The coverage-guided search strategies present lower satisfiability rates than the random-based search strategies. Such a conclusion is verified for the following properties: **P11** and **P12**.
- From all the possible features, only the *constants extraction* feature revealed significant statistical differences. Such differences are only exhibited by the property **P13**.
- The *constants extraction* feature usage on coverage-guided search strategies increases the satisfiability rates of the corresponding property.

Table 7.13: Tukey's test results concerning the *Satisfiability* evaluation metric for the *RERS Challenge 2012* benchmark.

Mutation	Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
P11	Coverage-Guided	Random	56.25	0.0010	39.4460	73.0540	True
P12	Coverage-Guided	Random	18.75	0.0096	5.3451	32.1549	True
P13	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-31.25	0.0170	-54.6107	-7.8893	True

7.5.4.4 Conclusions

According to this section benchmark's results, we can conclude that a random-based search strategy not provided with either the *pairwise testing* feature and the *constants extraction* feature composes the ideal search strategy for the `RERS Challenge 2012` benchmark. Such a conclusion is based on the higher satisfiability rates presented by the random-based search strategies and by the significant increase in time caused by the before-mentioned features.

Furthermore, it is noteworthy that the before-mentioned conclusion aligns with what was already expected, considering the heavy presence of constants in the source code and the fact that such constants do not influence the satisfiability-oriented test specifications associated with this section's benchmark.

7.5.5 Triangle Classification Program

The `Triangle Classification Program` benchmark's results — Appendix F — revealed significant statistical differences only on the ***Time*** evaluation metric. The following subsection presents and discusses the results concerning the before-mentioned metric.

7.5.5.1 Time

Table 7.14 presents the Tukey test's results associated with the impact of the considered search strategies on the *Time* evaluation metric. By analyzing the before-mentioned results, it is possible to conclude the following:

- The coverage-guided search strategies require less time than the random-based search strategies to find bugs in this section benchmark's variants. Such a conclusion is verified for all the considered benchmark variants, except for the benchmark variant **M2**, which reveals the opposite.

- From all the possible features, only the *pairwise testing* feature revealed significant statistical differences. Such differences are exhibited by the benchmark variants **M0**, **M3**, and **M4** for the random-based search strategies and by the benchmark variants **M0**, **M1**, **M3** for the coverage-guided search strategies.

- The *pairwise testing* feature usage on random-based search strategies decreases the time required to find bugs on the corresponding benchmark variants.

- The *pairwise testing* feature usage on coverage-guided search strategies increases the time required to find bugs on the corresponding benchmark variants, except for the benchmark variant **M4**, which reveals the opposite.

7.5.5.2 Conclusions

According to this section benchmark's results, we can conclude that a coverage-guided search strategy not provided with the *pairwise testing* feature composes the ideal search strategy for the `Triangle Classification Program` benchmark. Such a conclusion is based on the decrease in time implied by the coverage-guided search strategies and by the increase in time caused by the *pairwise testing* feature on such base search strategies.

Table 7.14: Tukey's test results concerning the *Time* evaluation metric for the *Triangle Classification Program* benchmark.

Mutation	Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
M0	Random withPairWiseTesting	Random withoutPairWiseTesting	15.2225	0.0090	5.4024	25.0426	True
M0	Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-21.8481	0.0216	-39.1835	-4.5128	True
M0	Coverage-Guided	Random	158.9534	0.0010	145.4685	172.4384	True
M1	Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-15.5350	0.0180	-27.3225	-3.7475	True
M1	Coverage-Guided	Random	165.1200	0.0010	142.7157	187.5243	True
M2	Coverage-Guided	Random	-111.4216	0.0010	-115.2162	-107.6270	True
M3	Random withPairWiseTesting	Random withoutPairWiseTesting	38.6675	0.0026	19.5267	57.8083	True
M3	Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-18.8069	0.0329	-35.4844	-2.1293	True
M3	Coverage-Guided	Random	152.3066	0.0010	132.0616	172.5516	True
M4	Random withPairWiseTesting	Random withoutPairWiseTesting	22.4462	0.0380	1.7239	43.1686	True
M4	Coverage-Guided	Random	153.6322	0.0010	140.7367	166.5276	True

Moreover, it is worth mentioning that the aforementioned conclusion partially aligns with what was already expected, considering that, given this section's benchmark underlying test specifications, it was expected for the random-based search strategies to be associated with faster times, especially when endowed with the *pairwise testing* feature. However, and even though the *pairwise testing* feature positively impacts the random-based search strategies, it was verified that such strategies were still slower than the coverage-guided ones. After a deeper analysis, we concluded that our expectations were not fully met because the benchmark's bugs were being found with the first input generated, which did not allow us to verify the search strategies' behavior in the long run.

7.5.6 Zodiac

The `Zodiac` benchmark's results — Appendix G — revealed significant statistical differences in the following evaluation metrics: ***Time***, ***Number of Test Cases***, and ***Coverage***. The following subsections present and discuss the results concerning each of those metrics.

7.5.6.1 Time

Table 7.15 presents the Tukey test's results associated with the impact of the considered search strategies on the *Time* evaluation metric. By analyzing the before-mentioned results, it is possible to conclude the following:

- The coverage-guided search strategies require more time than the random-based search strategies to find bugs in this section benchmark's variants. Such a conclusion is verified for all the considered benchmark variants.

- From all the possible features, only the *constants extraction* feature revealed significant statistical differences. Such differences are exhibited by all the considered benchmark variants for both coverage-guided (except the benchmark variant **M4**) and random-based search strategies.

- The *constants extraction* feature usage on both coverage-guided and random-based search strategies increases the time required to find bugs on the corresponding benchmark variants.

Table 7.15: Tukey's test results concerning the *Time* evaluation metric for the *Zodiac* benchmark.

Mutation	Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
M0	Random withConstantExtraction	Random withoutConstantExtraction	-5.0599	0.0010	-5.2142	-4.9056	True
M0	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-15.8719	0.0276	-29.2997	-2.4440	True
M0	Coverage-Guided	Random	-172.7412	0.0010	-181.4178	-164.0647	True
M1	Random withConstantExtraction	Random withoutConstantExtraction	-5.0403	0.0010	-5.1418	-4.9387	True
M1	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-4.8869	0.0010	-6.7012	-3.0725	True
M1	Coverage-Guided	Random	-128.1747	0.0010	-131.1142	-125.2351	True
M2	Random withConstantExtraction	Random withoutConstantExtraction	-27.7868	0.0010	-34.0829	-21.4906	True
M2	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-64.7456	0.0028	-97.2424	-32.2488	True
M2	Coverage-Guided	Random	-246.3419	0.0010	-277.9004	-214.7834	True
M3	Random withConstantExtraction	Random withoutConstantExtraction	-4.9934	0.0010	-5.1355	-4.8513	True
M3	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-6.5881	0.0010	-7.8741	-5.3022	True
M3	Coverage-Guided	Random	-149.3024	0.0010	-152.6940	-145.9109	True
M4	Random withConstantExtraction	Random withoutConstantExtraction	-5.0612	0.0010	-5.1558	-4.9666	True
M4	Coverage-Guided	Random	-186.8168	0.0010	-194.5256	-179.1079	True

7.5.6.2 Number of Test Cases

Table 7.16 presents the Tukey test's results associated with the impact of the considered search strategies on the *Number of Test Cases* evaluation metric. By analyzing the before-mentioned results, it is possible to conclude the following:

- The coverage-guided search strategies require fewer test cases than the random-based search strategies to find bugs in this section benchmark's variants. Such a conclusion is only verified for the benchmark variant **M2**.
- From all the possible features, only the *constants extraction* feature revealed significant statistical differences. Such differences are only exhibited by the benchmarks variants **M2** and **M4**.
- The *constants extraction* feature usage on coverage-guided strategies increases the number of test cases required to find bugs on the benchmark variant **M2**.
- The *constants extraction* feature usage on both coverage-guided and random-based search strategies decreases the number of test cases required to find bugs on the benchmark variant **M4**.

Table 7.16: Tukey's test results concerning the *Number of Test Cases* evaluation metric for the *Zodiac* benchmark.

Mutation	Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
M2	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-1093.75	0.0111	-1832.6464	-354.8536	True
M2	Coverage-Guided	Random	3843.75	0.0024	1617.3262	6070.1738	True
M4	Random withConstantExtraction	Random withoutConstantExtraction	1.50	0.0240	0.2765	2.7235	True
M4	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	2.25	0.0240	0.4148	4.0852	True

7.5.6.3 Coverage

Table 7.17 presents the Tukey test's results associated with the impact of the considered search strategies on the *Coverage* evaluation metric. By analyzing the before-mentioned results, it is possible to conclude the following:

- From all the possible features, only the *constants extraction* feature revealed significant statistical differences. Such differences are only exhibited by the benchmark variant **M2**.
- The *constants extraction* feature usage on coverage-guided strategies increases the coverage achieved while finding bugs on the benchmark variant **M2**.

Table 7.17: Tukey's test results concerning the *Coverage* evaluation metric for the *Zodiac* benchmark.

Mutation	Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
M2	Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-5.8475	0.001	-8.0826	-3.6124	True

7.5.6.4 Conclusions

According to this section benchmark's results, we can conclude that the ideal strategy depends on whether we want to minimize the *Time* or the *Number of Test Cases* evaluation metrics. If the former is chosen, then a random-based search strategy not provided with the *constants extraction* feature composes the ideal search strategy. However, if the latter is selected, then a random-based search strategy endowed with the *constants extraction* feature composes the ideal strategy for the `Zodiac` benchmark. Such a conclusion is based on the increase in time and decrease in the number of test cases caused by the *constants extraction* feature.

Furthermore, it is noteworthy that the before-mentioned conclusion aligns with what was already expected, considering the restricted test input's domain, and the heavy presence of constants in the source code, and their direct influence, on the integrity-oriented test specifications associated with this section's benchmark.

7.6 Hypothesis Evaluation

According to the results presented and discussed in the last section, it is possible to infer that none of the considered search strategies proved to be superior in terms of obtaining better results for all the considered benchmarks. Moreover, we can also conclude that the success of each search strategy is heavily dependent on the scenario in which the search strategy is used. Table 7.18 portrays the before-mentioned scenario dependency by presenting what was considered the ideal search strategy for each benchmark.

Based on the aforementioned scenario dependency, we can conclude that it is impossible to answer objectively the research questions outlined in Section 7.2 because without having knowledge about the scenario in which a given search strategy is applied, we cannot determine how a specific search strategy will impact a given evaluation metric. However, it is noteworthy that even though the before-mentioned research questions cannot be answered generically, the answer for each of those questions was implicitly given while discussing and presenting the results achieved, in the last section, for each of the considered benchmarks.

Table 7.18: Evaluation benchmarks' ideal search strategies. Check marks (✓) mean that the respective feature **is** included, X marks (✗) mean that the respective feature **is not** included, and hyphens (-) mean that the respective feature's inclusion is optional.

Benchmark	Base Search Strategy	Biased Sampling	Constants Extraction	Pairwise Testing
Defective Calculator	Coverage-Guided	-	✓	-
JS Algorithms - Stack	Random	-	✗	-
Multiplication Sign	Coverage-Guided / Random	-	-	✗
RERS Challenge 2012	Random	-	✗	✗
Triangle Classification Program	Coverage-Guided	-	-	✗
Zodiac	Random	-	✗	-
	Coverage-Guided	-	✓	-

Furthermore, by not being able to generically answer the defined research questions, we are implicitly validating this dissertation's central hypothesis (Section 4.2) by proving that there is no evidence of the existence of a meta-strategy capable of outperforming the current alternatives in every possible scenario and that the existence of an extensible PBT framework, capable of supporting multiple search strategies, is critical in terms of allowing a developer to choose and control the most suitable search strategy for a particular scenario, in order to increase both robustness and confidence in the outcomes.

7.7 Lessons Learned

Several lessons were learned during this chapter's evaluation process, not only concerning the search strategies as a whole but also involving each of the considered search strategy features.

Concerning the search strategies and their corresponding possible configurations, it was possible to learn that the most simplistic search strategy generally ends up being the more effective one. Such a lesson is supported by the high rate of optionality depicted in Table 7.18, which reveals that the impact of specific features on the base strategy is not significant in most of the considered benchmarks.

Furthermore, it was also possible to learn, through the `RERS Challenge 2012` benchmark, that the coverage-guided search strategies are very inefficient when provided with a seed collection with inputs that are on the opposite extreme of the input that triggers a given behavior, especially when such search strategies are endowed with an entirely random mutation logic. Such a lesson reinforces the critical role of the mutation logic in the coverage-guided search strategies' success.

Ultimately, the most important lesson learned was that selecting a search strategy not only heavily depends on the scenario in which it is applied but also on the evaluation metrics we want such search strategy to excel on. This is because while in some scenarios, we may want to minimize the time, in others, we may want to boost our confidence in the test results by maximizing the coverage achieved. Thus, we have to understand the scenario and which evaluation metrics we want to favor while testing in order to select the most suitable search strategy.

7.8 Threats to Validity

The current section aims to recognize any validity threats, either internal or external, that might have undue influence on the conducted research and data collection processes. Although no external validity threats concerning this dissertation's evaluation process were found, we were able to identify a series of internal validity threats related to the following areas:

- **Implementation**: The implementation of this dissertation's proposed solution composes an internal validity threat in the sense that such implementation, if improved at several different levels, can entirely influence the results achieved.
- **Benchmarks**: Even though the benchmarks used during this chapter's evaluation process were selected through a systematic process, such benchmarks may not sufficiently represent real-world programs, given the total number of benchmarks considered and their complexity.
- **Experimental Evaluation Setup**: The number of benchmark variants considered, coupled with the number of times each variant was executed, compose this section's main internal validity threat, given the fact that such values determine our confidence over the extracted metrics. In fact, the greater these values are, the more reliable our results are. However, it is worth mentioning that the approach followed for this chapter's evaluation process (4x3) was adequate enough to unveil patterns in the extracted evaluation metrics for the considered benchmarks.
- **Test Specifications**: The test specifications can be considered a significant internal validity threat, given the fact that by manipulating how each test specification is defined (e.g., number of arbitraries, arbitraries domain, etc), we can directly influence the results achieved by the considered search strategies. Nevertheless, we minimized the before-mentioned influence by following an unbiased process to avoid favoring a given search strategy during the definition of such test specifications. In fact, some of the tests specified were the same as the ones already used by the benchmarks, but adapted to PBT.

Chapter 8

Conclusions

This chapter presents an overview of this dissertation and reflects on the developed solution. Section 8.1 describes the difficulties faced while developing the proposed solution. Section 8.2 presents the main contributions made during this dissertation's development. Section 8.3 outlines the most promising paths to be explored as future work. Lastly, Section 8.4 concludes this dissertation by presenting an overview of the developed work.

8.1 Difficulties

Throughout the development of this dissertation's solution, there were several obstacles that we had to overcome, ranging from the design of the solution's architecture to the implementation of specific features.

Concerning the solution's architectural design, the main challenge was to ensure that not only the designed architecture was capable of supporting multiple search strategies but also capable of providing a high degree of scalability and extensibility that would allow the usage of such search strategies, either together or independently.

Additionally to the architectural challenge just described, we also faced difficulties while implementing the constants extraction and pairwise testing features. The main difficulty while implementing the former was to ensure that each arbitrary was capable of using the extracted constants. To overcome such difficulty, we endowed each arbitrary with the ability to determine which constants it is capable of using. Thus, deeper insights about all FluentCheck's arbitraries were required. Concerning the implementation of the pairwise testing feature, the main challenge was the definition of a complex algorithm capable of generating all pairs of test cases in a specific order that would allow FluentCheck's satisfiability checker to solve the *Boolean Satisfiability Problem* for both universal and existential quantifiers.

Lastly, there were also several difficulties associated with the implementation of the coverage-guided base search strategy. The main difficulty was related to the fact that such a search strategy requires source code instrumentation and involves an iterative process for tracking coverage on each test iteration. To overcome such difficulty, we decide to use a third-party test coverage tool. However, given the typical workflow associated with this type of tools, in which coverage is measured over an entire test suite, we had to manipulate such a tool in order to be able to use it to fulfill our objectives, which required deeper insights into the tool's functionalities. Moreover, we also faced the difficulty of implementing the mutation logic associated with the before-mentioned search strategy. To overcome such a difficulty, we followed the same approach as the one followed by the constants extraction feature, meaning that we proceed with the implementation, on each arbitrary, of all the logic responsible for determining how a given arbitrary mutates a given input, which reinforced the need for profoundly understanding the FluentCheck arbitraries.

8.2 Main Contributions

Given the unprecedented nature of this dissertation's solution, we believe that its main contributions were:

- **Systematic Literature Review of PBT Search Strategies**: A review was conducted to assess the state of the art of PBT search strategies, which ended up including search strategies pertaining to other software testing domains. SBT, CGF, Concolic Testing, and Hybrid Fuzzing compose the additional considered domains.

- **Developed Solution**: The development of a unique solution capable of supporting multiple search strategies and enabling a thorough control of the search strategy used during the testing process. Moreover, several unique search strategies were also implemented. While some of them can be seen as an enhancement of other already proposed solutions, others are a complete innovation.

- **Search Strategies Evaluation**: This dissertation's conducted evaluation process compares the behavior of multiple search strategies in multiple benchmarks, all while using a single framework. Furthermore, such a process was conducted systematically to allow the corresponding results to be the most significant and trustworthy as possible.

8.3 Future Work

Even though architectural wise our proposed solution did not unveil any apparent way of being enhanced, we still could identify a set of improvements that can be performed in order to complement this dissertation's solution. Such improvements range from the implementation of new search strategies to the enhancement of the ones already supported.

According to the results presented in Section 7.5, we can conclude that the constants extraction feature can positively impact specific scenarios. Therefore, and by considering that such a feature partially composes one of the four strategies proposed by Rojas et al. [43], we believe that exploring Rojas et al.'s remaining strategies is a promising path to follow in order to enhance the implementation of the before-mentioned feature.

Furthermore, and by considering what we learned about the coverage-guided search strategy mutation logic, in Section 7.7, we believe that it is worthy of exploring the implementation of a new and dynamic weighted mutation logic in which test cases are gradually mutated according to a given weight. By doing so, we would address the inefficiency associated with the currently supported mutation logic by allowing the before-mentioned search strategy to start by generating new inputs closer to the one being mutated, and as the weight dynamically increases, such inputs would tend to become more and more distant from the original input.

Ultimately, the development of a meta-search strategy capable of automatically determining the most suitable search strategy for a given scenario and according to a given evaluation metric composes what we consider the most promising path to be explored. Moreover, it is worth mentioning that the development of such a strategy was already partially made throughout this dissertation's evaluation systematic process, considering the corresponding results achieved. Accordingly, to complete the development of such a strategy, we only need to define a process capable of automatically analyzing the data resulting from the evaluation process.

8.4 Conclusions

The heavy dominance of the random concept generally associated with PBT generators composes a critical problem in PBT since it implies a total arbitrary search over the input space. Aiming to address such a problem, we decided to build this dissertation around the belief that by integrating other types of search strategies in PBT, namely those which are informed and have already proved their value, we would guide the test data generation process in an informed and more efficient way.

Accordingly, we conducted a Systematic Literature Review to review and assess the current state of the art of PBT search strategies and informed search strategies, in general. According to the results derived from the literature review, it was possible to conclude that recent works addressing the same problem as the one identified in this dissertation have been performed not only in the PBT domain but also in several other domains. Apart from the PBT domain, SBT, CGF, Concolic Testing, and Hybrid Fuzzing compose the list of domains holding the most promising results regarding this dissertation's scope.

Despite the encouraging results found in the literature review, it was possible to verify that the identified solutions tend to be problem-dependent, considering that there is no evidence of a solution whose search strategy is capable of outperforming all the current alternatives in every possible scenario. Hence, and by assuming that the development of a meta-search strategy capable of ensuring the absence of errors in any system is not something trivial or even possible, we identified a gap in the literature in terms of providing a PBT framework with the ability to support several search strategies from multiple domains.

Striving to fill the aforementioned literature gap, we developed a solution that consisted of endowing the FluentCheck PBT framework with an easily extensible and scalable architecture capable of supporting several different search strategies. Furthermore, to make out the most of the before-mentioned architectural characteristics, such a solution also involved the development of an entirely new coverage-guided search strategy and the implementation of the unique constants extraction and pairwise testing features.

The developed solution was evaluated through the comparison of a set of evaluation metrics resulting from the execution of a set of sixteen unique search strategies over six different benchmarks, all while only using the framework in which our solution was developed. Moreover, the conducted evaluation process was performed throughout a systematic process in order to allow the corresponding results to be the most significant and trustworthy as possible.

According to the results obtained from the evaluation process, it was possible to conclude that the success of each search strategy cannot be generally determined since it heavily depends on the scenario in which the strategy is applied. Furthermore, such a conclusion validated the hypothesis underlying this dissertation in the sense that it proved that there is, undoubtedly, the need for having a framework capable of supporting multiple search strategies in order to increase both robustness and confidence in the outcomes.

Lastly, several ways of improving the developed solution were identified. However, by considering the developed solution and the conducted evaluation process, we consider the development of a meta-search strategy capable of automatically determining the most suitable search strategy for a given scenario to be the most promising path to be explored.

Appendix A

Benchmarks Selection Process Data

This appendix presents all the data associated with the systematic process conducted to select the benchmarks to be used during the evaluation phase (Section 7.3.1).

Table A.1: Benchmarks selection process preliminary phase. Hyphens (-) mean *no information available*

Reference	Benchmarks	Language	Potential
[45]	-	-	-
[32]	All Programs Used by *CollAFL* [16]	-	Low
[49]	9 Multi-Purpose Programs	C	Low
[52]	*LAVA-M* and *Google Fuzzer* Test Suite Benchmark Datasets	C	Low
[53]	9 C Popular Applications	C	Low
[44]	**Triangle Classification Program + Remainder Calculation Program + Single Source Shortest Path**	**MATLAB**	**High**
[42]	5 Real-World Java Benchmarks	Java	Medium
[27]	*LAVA-M* Benchmark Programs	C	Low
[34]	20 Real-World Applications (18 From *Google Fuzzer* Test Suite)	C	Low
[5]	8 Open-Source Programs	C	Low
[25]	**2 Stack-Based Machines - Noninterference**	**Coq**	**High**
[12]	**14 Product Series Programs + 11 Random Benchmark Programs**	**Java**	**High**
[8]	*LAVA-M* Benchmark Programs and *GNU* Real Programs	C	Low
[36]	*OpenJDK*, *Apache Projects*, *Google Closure*, and *Mozilla Rhino*	Java	Medium
[37]	5 Real-World Java Benchmarks	Java	Medium
[54]	10 Real-World Programs from Fuzzing Projects and 10 *Google Fuzzer* Test-Suite Programs	C	Low
[30]	Energy Efficiency of MAC Protocols, Routing Tree for Directional Antennas, Noninterference	-	Low
[16]	24 Popular Open-Source Linux Applications and *LAVA-M* Dataset	-	Low
[35]	9 Open-Source Programs	C	Low
[26]	4 Real-World Programs	C	Low
[55]	*OSS-Fuzz* Applications and Libraries	C	Low
[28]	Energy Efficiency of MAC Protocols, Routing Tree for Directional Antennas, Noninterference	-	Low
[43]	SF110 Corpus + 10 Most Popular *SourceForge* Projects	Java	Medium

Table A.2: Analysis of the most promising benchmarks. Check marks (✓) mean that the respective benchmark **is** selected, empty () means that the respective benchmark **is not** selected.

Reference	Benchmark	Type	Observations	Status
[44]	Triangle Classification Program	Research	Most famous example to be considered in a research paper concerning the software testing domain due to its interesting control flow.	✓
	Remainder Calculation Program	Artificial	Excluded given the nature of the synthetically designed benchmark.	
	Single Source Shortest Path	Artificial	Source code not directly available.	
[12]	Product Series	Open-Source	Several different alternatives available, which make them more appealing.	
	Zodiac	Online Program	Real and online problem with complex predicates.	✓
	Tritype	Research	Equivalent to the *Triangle Classification Program*	✓
	TCAS	Real-World	Not available at the time the benchmarking process occurred.	
	Weight	Artificial	Selected as a motivational example to present the respective paper.	
	Elevator	Academic	Scenario with great potential but the available source-code limits the properties definition.	
	CFG	Academic	A binary problem that makes the process of defining properties too simple.	
	Vending Machine	Academic	The program's simplicity is reflected in the lack of possibility to define multiple properties.	
	Biggest Of 5 Numbers	Academic	Discarded given the lack of properties associated with it.	
	Zero Subset	Academic	The process of defining properties is not trivial, given the complexity of the problem.	
	Multiplication Sign	Academic	Simple yet with a couple of easy properties to define, making it more appellative.	✓
	Problem-1-Int	Challenge	Special benchmark due to its nature and complexity reflected by a considerable number of predicates.	✓
[25]	Stack Machine	Real-World	Source code not available.	

Appendix B

Defective Calculator Results

This appendix presents all the data associated with the results achieved for the `Defective Calculator` benchmark, including each search strategy's results and all the data extracted for each property.

B.1 Search Strategies

Table B.1: PBT_R_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	2136.275	118.71766	1000	1000000	0	0.0	0.0	0.0
2	2299.850	81.99325	1000	1000000	0	0.0	0.0	0.0
3	2321.650	42.38269	1000	1000000	0	0.0	0.0	0.0
4	2370.125	96.98826	1000	1000000	0	0.0	0.0	0.0
5	2318.050	47.54306	1000	1000000	0	0.0	0.0	0.0
6	2309.700	25.49951	1000	1000000	0	0.0	0.0	0.0
7	2400.125	38.90202	1000	1000000	0	0.0	0.0	0.0
8	2351.650	54.80956	1000	1000000	0	0.0	0.0	0.0

Table B.2: PBT_R_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	2181.975	2.72706	1000	1000000	0	0.0	0.0	0.0
2	2298.375	18.26477	1000	1000000	0	0.0	0.0	0.0
3	2346.000	24.42478	1000	1000000	0	0.0	0.0	0.0
4	2403.600	60.05214	1000	1000000	0	0.0	0.0	0.0
5	2399.175	90.10290	1000	1000000	0	0.0	0.0	0.0
6	2354.350	47.27666	1000	1000000	0	0.0	0.0	0.0
7	2382.650	49.57507	1000	1000000	0	0.0	0.0	0.0
8	2355.850	54.93589	1000	1000000	0	0.0	0.0	0.0

Table B.3: PBT_R_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	311.7825	5.22356	1000	2	0	0.0	0.0	100.0
2	291.2050	7.06346	1000	2007	0	0.0	0.0	100.0
3	334.8400	15.05671	1000	3	0	0.0	0.0	100.0
4	313.6850	17.51939	1000	3	0	0.0	0.0	100.0
5	312.9550	15.04052	1000	34	0	0.0	0.0	100.0
6	314.7500	11.01582	1000	34	0	0.0	0.0	100.0
7	418.8400	15.09369	1000	39001	0	0.0	0.0	100.0
8	308.8300	12.43036	1000	31	0	0.0	0.0	100.0

Table B.4: PBT_R_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	2182.475	25.91384	1000	1000000	0	0.0	0.0	0.0
2	2299.225	33.86085	1000	1000000	0	0.0	0.0	0.0
3	2286.775	13.22561	1000	1000000	0	0.0	0.0	0.0
4	2331.700	50.42346	1000	1000000	0	0.0	0.0	0.0
5	2406.750	23.94875	1000	1000000	0	0.0	0.0	0.0
6	2302.975	15.41142	1000	1000000	0	0.0	0.0	0.0
7	2326.275	33.04976	1000	1000000	0	0.0	0.0	0.0
8	2375.625	72.18020	1000	1000000	0	0.0	0.0	0.0

Table B.5: PBT_R_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	306.2450	6.41822	1000	2	0	0.0	0.0	100.0
2	304.1075	17.86592	1000	2007	0	0.0	0.0	100.0
3	326.7525	25.95437	1000	3	0	0.0	0.0	100.0
4	335.6500	4.49111	1000	3	0	0.0	0.0	100.0
5	309.7650	18.79717	1000	34	0	0.0	0.0	100.0
6	310.3350	21.81751	1000	34	0	0.0	0.0	100.0
7	418.8850	11.36944	1000	39001	0	0.0	0.0	100.0
8	312.2800	18.17602	1000	31	0	0.0	0.0	100.0

Table B.6: PBT_R_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	2198.525	17.41944	1000	1000000	0	0.0	0.0	0.0
2	2320.700	40.94692	1000	1000000	0	0.0	0.0	0.0
3	2321.550	26.08357	1000	1000000	0	0.0	0.0	0.0
4	2395.100	34.91282	1000	1000000	0	0.0	0.0	0.0
5	2390.375	45.11005	1000	1000000	0	0.0	0.0	0.0
6	2345.825	32.06177	1000	1000000	0	0.0	0.0	0.0
7	2374.175	84.20705	1000	1000000	0	0.0	0.0	0.0
8	2348.750	29.08939	1000	1000000	0	0.0	0.0	0.0

Table B.7: PBT_R_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	303.9850	3.91296	1000	2	0	0.0	0.0	100.0
2	303.0275	22.57648	1000	2007	0	0.0	0.0	100.0
3	334.6775	26.41179	1000	3	0	0.0	0.0	100.0
4	322.7150	14.48499	1000	3	0	0.0	0.0	100.0
5	308.2750	19.98972	1000	34	0	0.0	0.0	100.0
6	334.2150	7.05602	1000	34	0	0.0	0.0	100.0
7	414.2250	15.64677	1000	39001	0	0.0	0.0	100.0
8	316.5375	20.08620	1000	31	0	0.0	0.0	100.0

Table B.8: PBT_R_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	306.4800	4.79674	1000	2	0	0.0	0.0	100.0
2	327.3525	22.25523	1000	2007	0	0.0	0.0	100.0
3	305.2800	27.59124	1000	3	0	0.0	0.0	100.0
4	318.5550	15.74773	1000	3	0	0.0	0.0	100.0
5	322.6750	7.99614	1000	34	0	0.0	0.0	100.0
6	305.2575	14.45204	1000	34	0	0.0	0.0	100.0
7	417.2250	18.28269	1000	39001	0	0.0	0.0	100.0
8	307.9725	18.05277	1000	31	0	0.0	0.0	100.0

Table B.9: PBT_CG_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	3392.275	104.07546	1000	128582	3716	17.54	0.0	100.0
2	3829.450	480.11631	1000	145290	19441	17.54	0.0	100.0
3	3428.800	45.67674	1000	127522	1232	17.54	0.0	100.0
4	3468.575	90.22858	1000	128133	3814	17.54	0.0	100.0
5	3520.025	111.14543	1000	131673	5168	17.54	0.0	100.0
6	3472.825	47.60312	1000	130198	1471	17.54	0.0	100.0
7	3821.425	163.92877	1000	144400	6876	17.54	0.0	100.0
8	3318.725	34.99438	1000	125013	7	17.54	0.0	100.0

Table B.10: PBT_CG_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	3390.500	73.00798	1000	128495	2809	17.54	0.0	100.0
2	4206.150	472.81243	1000	158317	19106	17.54	0.0	100.0
3	3325.125	23.74041	1000	125619	617	17.54	0.0	100.0
4	3305.200	10.30946	1000	125152	229	17.54	0.0	100.0
5	3380.925	66.29568	1000	128098	2848	17.54	0.0	100.0
6	3434.725	58.19276	1000	129815	1146	17.54	0.0	100.0
7	4177.550	637.39595	1000	161441	26239	17.54	0.0	100.0
8	3331.550	11.51618	1000	125044	59	17.54	0.0	100.0

Table B.11: PBT_CG_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	271.3525	3.87822	1000	2	0	17.54	0.0	100.0
2	207.5600	8.84606	1000	507	0	17.54	0.0	100.0
3	154.2075	5.47082	1000	3	0	17.54	0.0	100.0
4	150.1425	2.71402	1000	3	0	17.54	0.0	100.0
5	138.4025	2.11819	1000	40	0	17.54	0.0	100.0
6	137.4150	1.98865	1000	40	0	17.54	0.0	100.0
7	445.0900	4.57434	1000	11001	0	17.54	0.0	100.0
8	143.3075	0.60615	1000	251	0	17.54	0.0	100.0

Table B.12: PBT_CG_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	3411.850	79.29718	1000	128958	3456	17.54	0.0	100.0
2	3891.175	384.10168	1000	146889	14716	17.54	0.0	100.0
3	3388.750	58.82672	1000	127560	1618	17.54	0.0	100.0
4	3433.475	74.14423	1000	127437	2778	17.54	0.0	100.0
5	3475.800	77.53357	1000	130570	2825	17.54	0.0	100.0
6	3524.500	172.85942	1000	132594	5756	17.54	0.0	100.0
7	4249.975	786.96042	1000	164012	31784	17.54	0.0	100.0
8	3325.825	30.23304	1000	125039	25	17.54	0.0	100.0

Table B.13: PBT_CG_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	268.9550	5.79391	1000	2	0	17.54	0.0	100.0
2	205.7475	3.03409	1000	507	0	17.54	0.0	100.0
3	152.2650	2.81802	1000	3	0	17.54	0.0	100.0
4	147.9025	1.34306	1000	3	0	17.54	0.0	100.0
5	140.0975	1.52803	1000	40	0	17.54	0.0	100.0
6	141.6625	2.01362	1000	40	0	17.54	0.0	100.0
7	449.4075	2.84798	1000	11001	0	17.54	0.0	100.0
8	145.6900	0.84074	1000	251	0	17.54	0.0	100.0

Table B.14: PBT_CG_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	3341.825	47.21951	1000	125656	370	17.54	0.0	100.0
2	3658.350	367.51573	1000	137445	11754	17.54	0.0	100.0
3	3370.500	69.50464	1000	126547	665	17.54	0.0	100.0
4	3375.050	58.37527	1000	125408	362	17.54	0.0	100.0
5	3413.175	76.18101	1000	126051	1019	17.54	0.0	100.0
6	3360.650	41.23472	1000	126881	342	17.54	0.0	100.0
7	3689.325	200.52330	1000	139768	8857	17.54	0.0	100.0
8	3315.325	47.31371	1000	125045	54	17.54	0.0	100.0

Table B.15: PBT_CG_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	277.9150	15.53234	1000	2	0	17.54	0.0	100.0
2	214.6500	27.29234	1000	507	0	17.54	0.0	100.0
3	160.6675	21.04158	1000	3	0	17.54	0.0	100.0
4	159.8225	8.99013	1000	3	0	17.54	0.0	100.0
5	156.1600	19.33533	1000	40	0	17.54	0.0	100.0
6	143.4775	5.55638	1000	40	0	17.54	0.0	100.0
7	459.0475	17.95027	1000	11001	0	17.54	0.0	100.0
8	143.5775	1.84144	1000	251	0	17.54	0.0	100.0

Table B.16: PBT_CG_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	266.3525	4.03190	1000	2	0	17.54	0.0	100.0
2	201.2050	1.23777	1000	507	0	17.54	0.0	100.0
3	154.0400	3.28165	1000	3	0	17.54	0.0	100.0
4	150.6125	2.02081	1000	3	0	17.54	0.0	100.0
5	137.8425	2.14918	1000	40	0	17.54	0.0	100.0
6	138.0825	2.35462	1000	40	0	17.54	0.0	100.0
7	452.9450	2.08850	1000	11001	0	17.54	0.0	100.0
8	146.9725	5.66565	1000	251	0	17.54	0.0	100.0

B.2 Properties

B.2.1 P1

Table B.17: P1 - Search strategies results.

Strategy	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
PBT_R_S4	2182.4750	25.91384	1000.0	1000000.0	0.0	0.00	0.0	0.0
PBT_R_S1	2136.2750	118.71766	1000.0	1000000.0	0.0	0.00	0.0	0.0
PBT_R_S8	306.4800	4.79674	1000.0	2.0	0.0	0.00	0.0	100.0
PBT_CG_S4	3411.8500	79.29718	1000.0	128958.0	3456.0	17.54	0.0	100.0
PBT_CG_S6	3341.8250	47.21951	1000.0	125656.0	370.0	17.54	0.0	100.0
PBT_CG_S3	271.3525	3.87822	1000.0	2.0	0.0	17.54	0.0	100.0
PBT_R_S3	311.7825	5.22356	1000.0	2.0	0.0	0.00	0.0	100.0
PBT_R_S5	306.2450	6.41822	1000.0	2.0	0.0	0.00	0.0	100.0
PBT_R_S7	303.9850	3.91296	1000.0	2.0	0.0	0.00	0.0	100.0
PBT_CG_S5	268.9550	5.79391	1000.0	2.0	0.0	17.54	0.0	100.0
PBT_CG_S7	277.9150	15.53234	1000.0	2.0	0.0	17.54	0.0	100.0
PBT_CG_S1	3392.2750	104.07546	1000.0	128582.0	3716.0	17.54	0.0	100.0
PBT_CG_S8	266.3525	4.03190	1000.0	2.0	0.0	17.54	0.0	100.0
PBT_CG_S2	3390.5000	73.00798	1000.0	128495.0	2809.0	17.54	0.0	100.0
PBT_R_S6	2198.5250	17.41944	1000.0	1000000.0	0.0	0.00	0.0	0.0
PBT_R_S2	2181.9750	2.72706	1000.0	1000000.0	0.0	0.00	0.0	0.0

Table B.18: P1 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-14.6769	0.9000	-1880.6359	1851.2821	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-13.7969	0.9000	-1879.7626	1852.1688	False
Random withConstantExtraction	Random withoutConstantExtraction	1867.6894	0.0010	1834.6321	1900.7467	True
Coverage-Guided withBias	Coverage-Guided withoutBias	21.4400	0.9000	-3088.3985	3131.2785	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	6.2850	0.9000	-3103.6210	3116.1910	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	3112.9687	0.0010	3076.0032	3149.9343	True
Coverage-Guided	Random	-586.6603	0.4069	-2058.2416	884.9210	False

Table B.19: P1 - Tukey's test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.000	0.9000	-998945.5813	998945.5813	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.000	0.9000	-998945.5813	998945.5813	False
Coverage-Guided withBias	Coverage-Guided withoutBias	847.250	0.9000	-126949.6815	128644.1815	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	615.750	0.9000	-127182.5038	128414.0038	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	127920.750	0.0010	126055.6345	129785.8655	True
Coverage-Guided	Random	436038.625	0.0382	27409.2752	844667.9748	True

Table B.20: P1 - Tukey's test results concerning the *Satisfiability* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.0	0.9000	-99.8948	99.8948	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.0	0.9000	-99.8948	99.8948	False
Coverage-Guided	Random	-50.0	0.0192	-90.5327	-9.4673	True

B.2.2 P2

Table B.21: P2 - Search strategies results.

Strategy	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
PBT_R_S4	2299.2250	33.86085	1000.0	1000000.0	0.0	0.00	0.0	0.0
PBT_R_S1	2299.8500	81.99325	1000.0	1000000.0	0.0	0.00	0.0	0.0
PBT_R_S8	327.3525	22.25523	1000.0	2007.0	0.0	0.00	0.0	100.0
PBT_CG_S4	3891.1750	384.10168	1000.0	146889.0	14716.0	17.54	0.0	100.0
PBT_CG_S6	3658.3500	367.51573	1000.0	137445.0	11754.0	17.54	0.0	100.0
PBT_CG_S3	207.5600	8.84606	1000.0	507.0	0.0	17.54	0.0	100.0
PBT_R_S3	291.2050	7.06346	1000.0	2007.0	0.0	0.00	0.0	100.0
PBT_R_S5	304.1075	17.86592	1000.0	2007.0	0.0	0.00	0.0	100.0
PBT_R_S7	303.0275	22.57648	1000.0	2007.0	0.0	0.00	0.0	100.0
PBT_CG_S5	205.7475	3.03409	1000.0	507.0	0.0	17.54	0.0	100.0
PBT_CG_S7	214.6500	27.29234	1000.0	507.0	0.0	17.54	0.0	100.0
PBT_CG_S1	3829.4500	480.11631	1000.0	145290.0	19441.0	17.54	0.0	100.0
PBT_CG_S8	201.2050	1.23777	1000.0	507.0	0.0	17.54	0.0	100.0
PBT_CG_S2	4206.1500	472.81243	1000.0	158317.0	19106.0	17.54	0.0	100.0
PBT_R_S6	2320.7000	40.94692	1000.0	1000000.0	0.0	0.00	0.0	0.0
PBT_R_S2	2298.3750	18.26477	1000.0	1000000.0	0.0	0.00	0.0	0.0

Table B.22: P2 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-14.3069	0.9000	-2010.3967	1981.7830	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-14.1919	0.9000	-2010.2825	1981.8988	False
Random withConstantExtraction	Random withoutConstantExtraction	1998.1144	0.0010	1975.3796	2020.8492	True
Coverage-Guided withBias	Coverage-Guided withoutBias	-32.1544	0.9000	-3727.7509	3663.4422	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	120.8819	0.9000	-3572.8810	3814.6447	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	3688.9906	0.0010	3408.9145	3969.0668	True
Coverage-Guided	Random	-746.3056	0.3636	-2450.6178	958.0066	False

Table B.23: P2 - Tukey's test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.000	0.9000	-996942.6914	996942.6914	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.000	0.9000	-996942.6914	996942.6914	False
Coverage-Guided withBias	Coverage-Guided withoutBias	-895.750	0.9000	-147595.6470	145804.1470	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	4818.250	0.9000	-141805.3967	151441.8967	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	146478.250	0.0010	135946.3844	157010.1156	True
Coverage-Guided	Random	427257.375	0.0417	18388.1005	836126.6495	True

Table B.24: P2 - Tukey's test results concerning the *Satisfiability* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.0	0.9000	-99.8948	99.8948	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.0	0.9000	-99.8948	99.8948	False
Coverage-Guided	Random	-50.0	0.0192	-90.5327	-9.4673	True

B.2.3 P3

Table B.25: P3 - Search strategies results.

Strategy	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
PBT_R_S4	2286.7750	13.22561	1000.0	1000000.0	0.0	0.00	0.0	0.0
PBT_R_S1	2321.6500	42.38269	1000.0	1000000.0	0.0	0.00	0.0	0.0
PBT_R_S8	305.2800	27.59124	1000.0	3.0	0.0	0.00	0.0	100.0
PBT_CG_S4	3388.7500	58.82672	1000.0	127560.0	1618.0	17.54	0.0	100.0
PBT_CG_S6	3370.5000	69.50464	1000.0	126547.0	665.0	17.54	0.0	100.0
PBT_CG_S3	154.2075	5.47082	1000.0	3.0	0.0	17.54	0.0	100.0
PBT_R_S3	334.8400	15.05671	1000.0	3.0	0.0	0.00	0.0	100.0
PBT_R_S5	326.7525	25.95437	1000.0	3.0	0.0	0.00	0.0	100.0
PBT_R_S7	334.6775	26.41179	1000.0	3.0	0.0	0.00	0.0	100.0
PBT_CG_S5	152.2650	2.81802	1000.0	3.0	0.0	17.54	0.0	100.0
PBT_CG_S7	160.6675	21.04158	1000.0	3.0	0.0	17.54	0.0	100.0
PBT_CG_S1	3428.8000	45.67674	1000.0	127522.0	1232.0	17.54	0.0	100.0
PBT_CG_S8	154.0400	3.28165	1000.0	3.0	0.0	17.54	0.0	100.0
PBT_CG_S2	3325.1250	23.74041	1000.0	125619.0	617.0	17.54	0.0	100.0
PBT_R_S6	2321.5500	26.08357	1000.0	1000000.0	0.0	0.00	0.0	0.0
PBT_R_S2	2346.0000	24.42478	1000.0	1000000.0	0.0	0.00	0.0	0.0

Table B.26: P3 - Tukey’s test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-5.4100	0.9000	-1997.2068	1986.3868	False
Random withPairWiseTesting	Random withoutPairWiseTesting	20.2400	0.9000	-1971.4615	2011.9415	False
Random withConstantExtraction	Random withoutConstantExtraction	1993.6062	0.0010	1959.2709	2027.9416	True
Coverage-Guided withBias	Coverage-Guided withoutBias	32.6237	0.9000	-3187.2511	3252.4986	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-3.3900	0.9000	-3223.4280	3216.6480	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	3222.9987	0.0010	3170.1929	3275.8046	True
Coverage-Guided	Random	-444.6038	0.5508	-1980.9018	1091.6943	False

Table B.27: P3 - Tukey’s test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.0	0.9000	-998944.5823	998944.5823	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.0	0.9000	-998944.5823	998944.5823	False
Coverage-Guided withBias	Coverage-Guided withoutBias	729.0	0.9000	-125949.4862	127407.4862	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-241.5	0.9000	-126921.8497	126438.8497	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	126809.0	0.0010	125679.4650	127938.5350	True
Coverage-Guided	Random	436594.0	0.0379	28022.4296	845165.5704	True

Table B.28: P3 - Tukey’s test results concerning the *Satisfiability* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.0	0.9000	-99.8948	99.8948	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.0	0.9000	-99.8948	99.8948	False
Coverage-Guided	Random	-50.0	0.0192	-90.5327	-9.4673	True

B.2.4 P4

Table B.29: P4 - Search strategies results.

Strategy	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
PBT_R_S4	2331.7000	50.42346	1000.0	1000000.0	0.0	0.00	0.0	0.0
PBT_R_S1	2370.1250	96.98826	1000.0	1000000.0	0.0	0.00	0.0	0.0
PBT_R_S8	318.5550	15.74773	1000.0	3.0	0.0	0.00	0.0	100.0
PBT_CG_S4	3433.4750	74.14423	1000.0	127437.0	2778.0	17.54	0.0	100.0
PBT_CG_S6	3375.0500	58.37527	1000.0	125408.0	362.0	17.54	0.0	100.0
PBT_CG_S3	150.1425	2.71402	1000.0	3.0	0.0	17.54	0.0	100.0
PBT_R_S3	313.6850	17.51939	1000.0	3.0	0.0	0.00	0.0	100.0
PBT_R_S5	335.6500	4.49111	1000.0	3.0	0.0	0.00	0.0	100.0
PBT_R_S7	322.7150	14.48499	1000.0	3.0	0.0	0.00	0.0	100.0
PBT_CG_S5	147.9025	1.34306	1000.0	3.0	0.0	17.54	0.0	100.0
PBT_CG_S7	159.8225	8.99013	1000.0	3.0	0.0	17.54	0.0	100.0
PBT_CG_S1	3468.5750	90.22858	1000.0	128133.0	3814.0	17.54	0.0	100.0
PBT_CG_S8	150.6125	2.02081	1000.0	3.0	0.0	17.54	0.0	100.0
PBT_CG_S2	3305.2000	10.30946	1000.0	125152.0	229.0	17.54	0.0	100.0
PBT_R_S6	2395.1000	34.91282	1000.0	1000000.0	0.0	0.00	0.0	0.0
PBT_R_S2	2403.6000	60.05214	1000.0	1000000.0	0.0	0.00	0.0	0.0

Table B.30: P4 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-28.6700	0.9000	-2079.2020	2021.8620	False
Random withPairWiseTesting	Random withoutPairWiseTesting	13.7475	0.9000	-2036.9385	2064.4335	False
Random withConstantExtraction	Random withoutConstantExtraction	2052.4800	0.0010	2011.3730	2093.5870	True
Coverage-Guided withBias	Coverage-Guided withoutBias	58.3125	0.9000	-3182.3940	3299.0190	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-11.7850	0.9000	-3252.9936	3229.4236	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	3243.4550	0.0010	3155.6907	3331.2193	True
Coverage-Guided	Random	-424.9562	0.5757	-1981.2249	1131.3124	False

Table B.31: P4 - Tukey's test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.00	0.9000	-998944.5823	998944.5823	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.00	0.9000	-998944.5823	998944.5823	False
Coverage-Guided withBias	Coverage-Guided withoutBias	1252.50	0.9000	-125150.5729	127655.5729	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	110.00	0.9000	-126299.2173	126519.2173	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	126529.50	0.0010	124721.7083	128337.2917	True
Coverage-Guided	Random	436733.75	0.0379	28176.0146	845291.4854	True

Table B.32: P4 - Tukey's test results concerning the *Satisfiability* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.0	0.9000	-99.8948	99.8948	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.0	0.9000	-99.8948	99.8948	False
Coverage-Guided	Random	-50.0	0.0192	-90.5327	-9.4673	True

B.2.5 P5

Table B.33: P5 - Search strategies results.

Strategy	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
PBT_R_S4	2406.7500	23.94875	1000.0	1000000.0	0.0	0.00	0.0	0.0
PBT_R_S1	2318.0500	47.54306	1000.0	1000000.0	0.0	0.00	0.0	0.0
PBT_R_S8	322.6750	7.99614	1000.0	34.0	0.0	0.00	0.0	100.0
PBT_CG_S4	3475.8000	77.53357	1000.0	130570.0	2825.0	17.54	0.0	100.0
PBT_CG_S6	3413.1750	76.18101	1000.0	126051.0	1019.0	17.54	0.0	100.0
PBT_CG_S3	138.4025	2.11819	1000.0	40.0	0.0	17.54	0.0	100.0
PBT_R_S3	312.9550	15.04052	1000.0	34.0	0.0	0.00	0.0	100.0
PBT_R_S5	309.7650	18.79717	1000.0	34.0	0.0	0.00	0.0	100.0
PBT_R_S7	308.2750	19.98972	1000.0	34.0	0.0	0.00	0.0	100.0
PBT_CG_S5	140.0975	1.52803	1000.0	40.0	0.0	17.54	0.0	100.0
PBT_CG_S7	156.1600	19.33533	1000.0	40.0	0.0	17.54	0.0	100.0
PBT_CG_S1	3520.0250	111.14543	1000.0	131673.0	5168.0	17.54	0.0	100.0
PBT_CG_S8	137.8425	2.14918	1000.0	40.0	0.0	17.54	0.0	100.0
PBT_CG_S2	3380.9250	66.29568	1000.0	128098.0	2848.0	17.54	0.0	100.0
PBT_R_S6	2390.3750	45.11005	1000.0	1000000.0	0.0	0.00	0.0	0.0
PBT_R_S2	2399.1750	90.10290	1000.0	1000000.0	0.0	0.00	0.0	0.0

Table B.34: P5 - Tukey’s test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-18.9900	0.9000	-2082.5216	2044.5416	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-22.0325	0.9000	-2085.5340	2041.4690	False
Random withConstantExtraction	Random withoutConstantExtraction	2065.1700	0.0010	2014.4966	2115.8434	True
Coverage-Guided withBias	Coverage-Guided withoutBias	54.5869	0.9000	-3246.7402	3355.9139	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-0.8819	0.9000	-3302.6591	3300.8954	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	3304.3556	0.0010	3227.2990	3381.4123	True
Coverage-Guided	Random	-449.3009	0.5584	-2029.1502	1130.5484	False

Table B.35: P5 - Tukey’s test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.0	0.9000	-998913.6150	998913.6150	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.0	0.9000	-998913.6150	998913.6150	False
Coverage-Guided withBias	Coverage-Guided withoutBias	2023.5	0.9000	-126919.7528	130966.7528	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	787.5	0.9000	-128169.1963	129744.1963	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	129058.0	0.0010	125972.4307	132143.5693	True
Coverage-Guided	Random	435448.0	0.0384	26771.5516	844124.4484	True

Table B.36: P5 - Tukey’s test results concerning the *Satisfiability* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.0	0.9000	-99.8948	99.8948	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.0	0.9000	-99.8948	99.8948	False
Coverage-Guided	Random	-50.0	0.0192	-90.5327	-9.4673	True

B.2.6 P6

Table B.37: P6 - Search strategies results.

Strategy	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
PBT_R_S4	2302.9750	15.41142	1000.0	1000000.0	0.0	0.00	0.0	0.0
PBT_R_S1	2309.7000	25.49951	1000.0	1000000.0	0.0	0.00	0.0	0.0
PBT_R_S8	305.2575	14.45204	1000.0	34.0	0.0	0.00	0.0	100.0
PBT_CG_S4	3524.5000	172.85942	1000.0	132594.0	5756.0	17.54	0.0	100.0
PBT_CG_S6	3360.6500	41.23472	1000.0	126881.0	342.0	17.54	0.0	100.0
PBT_CG_S3	137.4150	1.98865	1000.0	40.0	0.0	17.54	0.0	100.0
PBT_R_S3	314.7500	11.01582	1000.0	34.0	0.0	0.00	0.0	100.0
PBT_R_S5	310.3350	21.81751	1000.0	34.0	0.0	0.00	0.0	100.0
PBT_R_S7	334.2150	7.05602	1000.0	34.0	0.0	0.00	0.0	100.0
PBT_CG_S5	141.6625	2.01362	1000.0	40.0	0.0	17.54	0.0	100.0
PBT_CG_S7	143.4775	5.55638	1000.0	40.0	0.0	17.54	0.0	100.0
PBT_CG_S1	3472.8250	47.60312	1000.0	130198.0	1471.0	17.54	0.0	100.0
PBT_CG_S8	138.0825	2.35462	1000.0	40.0	0.0	17.54	0.0	100.0
PBT_CG_S2	3434.7250	58.19276	1000.0	129815.0	1146.0	17.54	0.0	100.0
PBT_R_S6	2345.8250	32.06177	1000.0	1000000.0	0.0	0.00	0.0	0.0
PBT_R_S2	2354.3500	47.27666	1000.0	1000000.0	0.0	0.00	0.0	0.0

Table B.38: P6 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-13.5319	0.9000	-2023.7465	1996.6828	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.2156	0.9000	-2010.0445	2010.4757	False
Random withConstantExtraction	Random withoutConstantExtraction	2012.0731	0.0010	1977.0830	2047.0632	True
Coverage-Guided withBias	Coverage-Guided withoutBias	50.7744	0.9000	-3254.4501	3355.9988	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	4.9794	0.9000	-3300.6305	3310.5892	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	3308.0156	0.0010	3223.5476	3392.4836	True
Coverage-Guided	Random	-471.9912	0.5337	-2041.8030	1097.8205	False

Table B.39: P6 - Tukey's test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.0	0.9000	-998913.6150	998913.6150	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.0	0.9000	-998913.6150	998913.6150	False
Coverage-Guided withBias	Coverage-Guided withoutBias	1524.0	0.9000	-128194.0958	131242.0958	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	134.5	0.9000	-129592.4595	129861.4595	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	129832.0	0.0010	126965.7959	132698.2041	True
Coverage-Guided	Random	435061.0	0.0386	26344.5396	843777.4604	True

Table B.40: P6 - Tukey's test results concerning the *Satisfiability* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.0	0.9000	-99.8948	99.8948	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.0	0.9000	-99.8948	99.8948	False
Coverage-Guided	Random	-50.0	0.0192	-90.5327	-9.4673	True

B.2.7 P7

Table B.41: P7 - Search strategies results.

Strategy	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
PBT_R_S4	2326.2750	33.04976	1000.0	1000000.0	0.0	0.00	0.0	0.0
PBT_R_S1	2400.1250	38.90202	1000.0	1000000.0	0.0	0.00	0.0	0.0
PBT_R_S8	417.2250	18.28269	1000.0	39001.0	0.0	0.00	0.0	100.0
PBT_CG_S4	4249.9750	786.96042	1000.0	164012.0	31784.0	17.54	0.0	100.0
PBT_CG_S6	3689.3250	200.52330	1000.0	139768.0	8857.0	17.54	0.0	100.0
PBT_CG_S3	445.0900	4.57434	1000.0	11001.0	0.0	17.54	0.0	100.0
PBT_R_S3	418.8400	15.09369	1000.0	39001.0	0.0	0.00	0.0	100.0
PBT_R_S5	418.8850	11.36944	1000.0	39001.0	0.0	0.00	0.0	100.0
PBT_R_S7	414.2250	15.64677	1000.0	39001.0	0.0	0.00	0.0	100.0
PBT_CG_S5	449.4075	2.84798	1000.0	11001.0	0.0	17.54	0.0	100.0
PBT_CG_S7	459.0475	17.95027	1000.0	11001.0	0.0	17.54	0.0	100.0
PBT_CG_S1	3821.4250	163.92877	1000.0	144400.0	6876.0	17.54	0.0	100.0
PBT_CG_S8	452.9450	2.08850	1000.0	11001.0	0.0	17.54	0.0	100.0
PBT_CG_S2	4177.5500	637.39595	1000.0	161441.0	26239.0	17.54	0.0	100.0
PBT_R_S6	2374.1750	84.20705	1000.0	1000000.0	0.0	0.00	0.0	0.0
PBT_R_S2	2382.6500	49.57507	1000.0	1000000.0	0.0	0.00	0.0	0.0

Table B.42: P7 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-8.3675	0.9000	-1960.1908	1943.4558	False
Random withPairWiseTesting	Random withoutPairWiseTesting	22.1500	0.9000	-1929.5657	1973.8657	False
Random withConstantExtraction	Random withoutConstantExtraction	1953.5125	0.0010	1914.7681	1992.2569	True
Coverage-Guided withBias	Coverage-Guided withoutBias	51.5775	0.9000	-3492.9040	3596.0590	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	10.5450	0.9000	-3534.2953	3555.3853	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	3532.9463	0.0010	3200.4516	3865.4409	True
Coverage-Guided	Random	-824.0456	0.2999	-2466.0041	817.9128	False

Table B.43: P7 - Tukey's test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.000	0.9000	-959987.6246	959987.6246	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.000	0.9000	-959987.6246	959987.6246	False
Coverage-Guided withBias	Coverage-Guided withoutBias	1800.750	0.9000	-140218.5139	143820.0139	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	515.250	0.9000	-141514.4732	142544.9732	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	141404.250	0.0010	126585.0127	156223.4873	True
Coverage-Guided	Random	437797.375	0.0318	44038.8563	831555.8937	True

Table B.44: P7 - Tukey's test results concerning the *Satisfiability* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.0	0.9000	-99.8948	99.8948	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.0	0.9000	-99.8948	99.8948	False
Coverage-Guided	Random	-50.0	0.0192	-90.5327	-9.4673	True

B.2.8 P8

Table B.45: P8 - Search strategies results.

Strategy	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
PBT_R_S4	2375.6250	72.18020	1000.0	1000000.0	0.0	0.00	0.0	0.0
PBT_R_S1	2351.6500	54.80956	1000.0	1000000.0	0.0	0.00	0.0	0.0
PBT_R_S8	307.9725	18.05277	1000.0	31.0	0.0	0.00	0.0	100.0
PBT_CG_S4	3325.8250	30.23304	1000.0	125039.0	25.0	17.54	0.0	100.0
PBT_CG_S6	3315.3250	47.31371	1000.0	125045.0	54.0	17.54	0.0	100.0
PBT_CG_S3	143.3075	0.60615	1000.0	251.0	0.0	17.54	0.0	100.0
PBT_R_S3	308.8300	12.43036	1000.0	31.0	0.0	0.00	0.0	100.0
PBT_R_S5	312.2800	18.17602	1000.0	31.0	0.0	0.00	0.0	100.0
PBT_R_S7	316.5375	20.08620	1000.0	31.0	0.0	0.00	0.0	100.0
PBT_CG_S5	145.6900	0.84074	1000.0	251.0	0.0	17.54	0.0	100.0
PBT_CG_S7	143.5775	1.84144	1000.0	251.0	0.0	17.54	0.0	100.0
PBT_CG_S1	3318.7250	34.99438	1000.0	125013.0	7.0	17.54	0.0	100.0
PBT_CG_S8	146.9725	5.66565	1000.0	251.0	0.0	17.54	0.0	100.0
PBT_CG_S2	3331.5500	11.51618	1000.0	125044.0	59.0	17.54	0.0	100.0
PBT_R_S6	2348.7500	29.08939	1000.0	1000000.0	0.0	0.00	0.0	0.0
PBT_R_S2	2355.8500	54.93589	1000.0	1000000.0	0.0	0.00	0.0	0.0

Table B.46: P8 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	6.9475	0.9000	-2037.5100	2051.4050	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-5.0687	0.9000	-2049.5318	2039.3943	False
Random withConstantExtraction	Random withoutConstantExtraction	2046.5638	0.0010	2030.9810	2062.1465	True
Coverage-Guided withBias	Coverage-Guided withoutBias	-2.0256	0.9000	-3176.6629	3172.6117	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	1.8931	0.9000	-3172.7443	3176.5305	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	3177.9694	0.0010	3168.8306	3187.1081	True
Coverage-Guided	Random	-399.1847	0.5947	-1931.3106	1132.9412	False

Table B.47: P8 - Tukey's test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.000	0.9000	-998916.6118	998916.6118	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.000	0.9000	-998916.6118	998916.6118	False
Coverage-Guided withBias	Coverage-Guided withoutBias	-9.250	0.9000	-124662.1755	124643.6755	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-6.750	0.9000	-124659.6756	124646.1756	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	124784.250	0.0010	124765.8202	124802.6798	True
Coverage-Guided	Random	437372.375	0.0376	28914.7557	845829.9943	True

Table B.48: P8 - Tukey's test results concerning the *Satisfiability* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.0	0.9000	-99.8948	99.8948	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.0	0.9000	-99.8948	99.8948	False
Coverage-Guided	Random	-50.0	0.0192	-90.5327	-9.4673	True

Appendix C

JS Algorithms - Stack Results

This appendix presents all the data associated with the results achieved for all the `JS Algorithms - Stack` benchmark variants, including each search strategy's results and all the data extracted concerning all the evaluation metrics.

C.1 M0

Table C.1: M0 - Search strategies results.

Strategy	Min Mean Time (ms)	Std Min Mean Time (ms)	Max Mean Time (ms)	Std Max Mean Time (ms)	Min Sample Size	Max Sample Size	Min Mean Test Cases	Std Min Mean Test Cases	Max Mean Test Cases	Std Max Mean Test Cases	Min Mean Coverage (%)	Std Min Mean Coverage (%)	Max Mean Coverage (%)	Std Max Mean Coverage (%)	Bug Found
PBT_R_S4	0.45505	0.06932	2.85415	0.71497	1000	1000	2	0	3	0	0.0	0.0	0.00	0.0	True
PBT_R_S1	0.46845	0.07261	2.92430	0.32566	1000	1000	2	0	3	0	0.0	0.0	0.00	0.0	True
PBT_R_S8	1.43763	0.13013	3.45003	0.14197	1000	1000	2	0	3	0	0.0	0.0	0.00	0.0	True
PBT_CG_S4	206.29500	4.87250	209.31000	2.66458	1000	1000	2	0	3	0	11.2	0.0	12.13	0.0	True
PBT_CG_S6	202.96750	2.06015	206.32500	1.22964	1000	1000	2	0	3	0	11.2	0.0	12.13	0.0	True
PBT_CG_S3	210.30000	6.39478	211.58500	7.74237	1000	1000	2	0	3	0	11.2	0.0	12.13	0.0	True
PBT_R_S3	1.66365	0.08367	3.87242	0.21558	1000	1000	2	0	3	0	0.0	0.0	0.00	0.0	True
PBT_R_S5	1.64888	0.10730	4.01088	0.36255	1000	1000	2	0	3	0	0.0	0.0	0.00	0.0	True
PBT_R_S7	1.41295	0.12469	3.60918	0.41144	1000	1000	2	0	3	0	0.0	0.0	0.00	0.0	True
PBT_CG_S5	205.66750	2.46459	210.45000	7.82284	1000	1000	2	0	3	0	11.2	0.0	12.13	0.0	True
PBT_CG_S7	208.51500	3.09639	219.24500	7.20810	1000	1000	2	0	3	0	11.2	0.0	12.13	0.0	True
PBT_CG_S1	205.69250	3.40865	208.33750	3.63112	1000	1000	2	0	3	0	11.2	0.0	12.13	0.0	True
PBT_CG_S8	205.55750	3.75359	209.06000	0.64393	1000	1000	2	0	3	0	11.2	0.0	12.13	0.0	True
PBT_CG_S2	209.12000	4.73450	214.68500	8.26249	1000	1000	2	0	3	0	11.2	0.0	12.13	0.0	True
PBT_R_S6	0.66626	0.03751	2.57655	0.15459	1000	1000	2	0	3	0	0.0	0.0	0.00	0.0	True
PBT_R_S2	0.55894	0.05033	2.45090	0.16500	1000	1000	2	0	3	0	0.0	0.0	0.00	0.0	True

Table C.2: M0 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-0.0779	0.8619	-1.0978	0.9420	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.0920	0.8387	-0.9267	1.1107	False
Random withConstantExtraction	Random withoutConstantExtraction	-1.0036	0.0010	-1.2063	-0.8009	True
Coverage-Guided withBias	Coverage-Guided withoutBias	1.8725	0.2989	-2.1573	5.9023	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	1.8613	0.3021	-2.1737	5.8962	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-1.4913	0.4168	-5.6768	2.6943	False
Coverage-Guided	Random	-205.7254	0.0010	-207.5752	-203.8756	True

C.1.1 Search Strategies

Table C.3: PBT_R_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	8.05350	1.15520	1000	1000	0	0.0	0.0	0.0
2	13.29075	1.30554	1000	1000	0	0.0	0.0	0.0
3	5.94685	0.26796	1000	15	9	0.0	0.0	0.0
4	14.53750	1.62658	1000	1000	0	0.0	0.0	0.0
5	9.06547	1.00971	1000	1000	0	0.0	0.0	0.0
6	8.81405	0.66780	1000	1000	0	0.0	0.0	0.0
7	2.92430	0.32566	1000	2	0	0.0	0.0	100.0
8	0.46845	0.07261	1000	3	0	0.0	0.0	100.0
9	5.16540	0.65661	1000	1000	0	0.0	0.0	0.0
10	8.37675	0.32892	1000	1000	0	0.0	0.0	0.0
11	9.89715	1.03872	1000	1000	0	0.0	0.0	0.0

Table C.4: PBT_R_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	7.82033	0.81252	1000	1000	0	0.0	0.0	0.0
2	13.29950	1.45419	1000	1000	0	0.0	0.0	0.0
3	5.94765	0.70425	1000	2	0	0.0	0.0	0.0
4	15.45850	2.25606	1000	1000	0	0.0	0.0	0.0
5	8.51245	0.68129	1000	1000	0	0.0	0.0	0.0
6	8.05297	0.66563	1000	1000	0	0.0	0.0	0.0
7	2.45090	0.16500	1000	2	0	0.0	0.0	100.0
8	0.55894	0.05033	1000	3	0	0.0	0.0	100.0
9	4.84088	0.44177	1000	1000	0	0.0	0.0	0.0
10	7.89677	0.82152	1000	1000	0	0.0	0.0	0.0
11	9.75820	0.92266	1000	1000	0	0.0	0.0	0.0

Table C.5: PBT_R_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14.82450	0.92106	1000	1000	0	0.0	0.0	0.0
2	17.40350	1.17413	1000	1000	0	0.0	0.0	0.0
3	7.58320	0.46470	1000	2	0	0.0	0.0	0.0
4	16.53900	1.06862	1000	1000	0	0.0	0.0	0.0
5	10.94065	0.73603	1000	1000	0	0.0	0.0	0.0
6	9.64640	0.65477	1000	1000	0	0.0	0.0	0.0
7	3.87242	0.21558	1000	2	0	0.0	0.0	100.0
8	1.66365	0.08367	1000	3	0	0.0	0.0	100.0
9	7.28277	0.45876	1000	1000	0	0.0	0.0	0.0
10	8.08565	0.47841	1000	1000	0	0.0	0.0	0.0
11	12.14850	0.52082	1000	1000	0	0.0	0.0	0.0

Table C.6: PBT_R_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	7.95830	0.43693	1000	1000	0	0.0	0.0	0.0
2	13.61450	0.61698	1000	1000	0	0.0	0.0	0.0
3	6.20215	0.38586	1000	26	30	0.0	0.0	0.0
4	15.35175	1.07112	1000	1000	0	0.0	0.0	0.0
5	9.04628	0.49701	1000	1000	0	0.0	0.0	0.0
6	7.93830	0.86124	1000	1000	0	0.0	0.0	0.0
7	2.85415	0.71497	1000	2	0	0.0	0.0	100.0
8	0.45505	0.06932	1000	3	0	0.0	0.0	100.0
9	5.02002	0.58473	1000	1000	0	0.0	0.0	0.0
10	8.43407	1.13946	1000	1000	0	0.0	0.0	0.0
11	9.83895	1.00544	1000	1000	0	0.0	0.0	0.0

Table C.7: PBT_R_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	13.84150	0.63961	1000	1000	0	0.0	0.0	0.0
2	17.27375	1.01115	1000	1000	0	0.0	0.0	0.0
3	7.86545	0.58442	1000	2	0	0.0	0.0	0.0
4	16.02300	0.78098	1000	1000	0	0.0	0.0	0.0
5	11.83400	0.49484	1000	1000	0	0.0	0.0	0.0
6	9.69303	0.36991	1000	1000	0	0.0	0.0	0.0
7	4.01088	0.36255	1000	2	0	0.0	0.0	100.0
8	1.64888	0.10730	1000	3	0	0.0	0.0	100.0
9	7.48462	0.27690	1000	1000	0	0.0	0.0	0.0
10	8.15442	0.91820	1000	1000	0	0.0	0.0	0.0
11	12.73425	1.33212	1000	1000	0	0.0	0.0	0.0

Table C.8: PBT_R_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	9.47632	1.77735	1000	1000	0	0.0	0.0	0.0
2	14.83125	3.24767	1000	1000	0	0.0	0.0	0.0
3	6.02520	0.36418	1000	2	0	0.0	0.0	0.0
4	15.18900	1.54000	1000	1000	0	0.0	0.0	0.0
5	10.04637	0.41589	1000	1000	0	0.0	0.0	0.0
6	8.34895	0.45080	1000	1000	0	0.0	0.0	0.0
7	2.57655	0.15459	1000	2	0	0.0	0.0	100.0
8	0.66626	0.03751	1000	3	0	0.0	0.0	100.0
9	5.17535	0.24544	1000	1000	0	0.0	0.0	0.0
10	9.05327	1.38255	1000	1000	0	0.0	0.0	0.0
11	10.05232	0.57084	1000	1000	0	0.0	0.0	0.0

Table C.9: PBT_R_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	13.91675	1.09994	1000	1000	0	0.0	0.0	0.0
2	17.39400	1.01781	1000	1000	0	0.0	0.0	0.0
3	8.11707	0.55579	1000	2	0	0.0	0.0	0.0
4	16.15800	0.87121	1000	1000	0	0.0	0.0	0.0
5	11.53175	0.78761	1000	1000	0	0.0	0.0	0.0
6	9.32537	0.66874	1000	1000	0	0.0	0.0	0.0
7	3.60918	0.41144	1000	2	0	0.0	0.0	100.0
8	1.41295	0.12469	1000	3	0	0.0	0.0	100.0
9	6.22703	0.46476	1000	1000	0	0.0	0.0	0.0
10	7.30270	0.56458	1000	1000	0	0.0	0.0	0.0
11	11.55375	0.70445	1000	1000	0	0.0	0.0	0.0

Table C.10: PBT_R_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14.02475	0.41539	1000	1000	0	0.0	0.0	0.0
2	17.94500	0.50256	1000	1000	0	0.0	0.0	0.0
3	8.71095	0.44354	1000	2	0	0.0	0.0	0.0
4	17.68300	1.55972	1000	1000	0	0.0	0.0	0.0
5	11.17395	0.94717	1000	1000	0	0.0	0.0	0.0
6	9.78080	1.26599	1000	1000	0	0.0	0.0	0.0
7	3.45003	0.14197	1000	2	0	0.0	0.0	100.0
8	1.43763	0.13013	1000	3	0	0.0	0.0	100.0
9	6.16705	0.44609	1000	1000	0	0.0	0.0	0.0
10	7.09452	0.06325	1000	1000	0	0.0	0.0	0.0
11	11.35187	0.83186	1000	1000	0	0.0	0.0	0.0

Table C.11: PBT_CG_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	535.9675	21.05703	1000	1000	0	9.85	0.0	0.0
2	361.0225	9.43556	1000	1000	0	10.62	0.0	0.0
3	247.4775	7.81937	1000	24	8	11.75	0.0	0.0
4	356.8700	9.46136	1000	1000	0	15.47	0.0	0.0
5	330.2250	10.43450	1000	1000	0	12.82	0.0	0.0
6	309.7750	5.81887	1000	1000	0	10.04	0.0	0.0
7	208.3375	3.63112	1000	2	0	11.20	0.0	100.0
8	205.6925	3.40865	1000	3	0	12.13	0.0	100.0
9	296.0450	8.69877	1000	1000	0	13.55	0.0	0.0
10	330.0650	10.36897	1000	1000	0	14.04	0.0	0.0
11	384.1625	30.73206	1000	1000	0	10.23	0.0	0.0

Table C.12: PBT_CG_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	541.8050	10.44939	1000	1000	0	9.85	0.0	0.0
2	367.4725	7.84458	1000	1000	0	10.62	0.0	0.0
3	246.2800	6.08486	1000	2	0	10.86	0.0	0.0
4	351.5200	4.33978	1000	1000	0	15.47	0.0	0.0
5	327.4150	11.67428	1000	1000	0	12.82	0.0	0.0
6	319.0100	8.81321	1000	1000	0	10.04	0.0	0.0
7	214.6850	8.26249	1000	2	0	11.20	0.0	100.0
8	209.1200	4.73450	1000	3	0	12.13	0.0	100.0
9	297.7075	6.08161	1000	1000	0	13.55	0.0	0.0
10	329.8025	8.34375	1000	1000	0	14.04	0.0	0.0
11	342.8325	2.71854	1000	1000	0	10.23	0.0	0.0

Table C.13: PBT_CG_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	547.8075	8.08677	1000	1000	0	9.85	0.0	0.0
2	384.0425	17.70079	1000	1000	0	10.62	0.0	0.0
3	251.9225	8.26725	1000	2	0	10.86	0.0	0.0
4	360.2325	14.09260	1000	1000	0	15.47	0.0	0.0
5	323.0425	11.04414	1000	1000	0	12.82	0.0	0.0
6	309.1275	6.62991	1000	1000	0	10.04	0.0	0.0
7	211.5850	7.74237	1000	2	0	11.20	0.0	100.0
8	210.3000	6.39478	1000	3	0	12.13	0.0	100.0
9	299.6575	6.01137	1000	1000	0	13.55	0.0	0.0
10	333.2925	6.25901	1000	1000	0	14.04	0.0	0.0
11	345.6475	30.52408	1000	1000	0	10.23	0.0	0.0

Table C.14: PBT_CG_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	532.0125	10.51116	1000	1000	0	9.85	0.0	0.0
2	361.5675	15.11199	1000	1000	0	10.62	0.0	0.0
3	246.6975	6.16730	1000	17	9	11.75	0.0	0.0
4	349.1975	4.22816	1000	1000	0	15.47	0.0	0.0
5	319.1325	10.83744	1000	1000	0	12.82	0.0	0.0
6	310.4200	7.10011	1000	1000	0	10.04	0.0	0.0
7	209.3100	2.66458	1000	2	0	11.20	0.0	100.0
8	206.2950	4.87250	1000	3	0	12.13	0.0	100.0
9	310.7175	7.12908	1000	1000	0	13.55	0.0	0.0
10	347.5175	25.48839	1000	1000	0	14.04	0.0	0.0
11	364.5300	44.53128	1000	1000	0	10.23	0.0	0.0

Table C.15: PBT_CG_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	543.5750	13.32020	1000	1000	0	9.85	0.0	0.0
2	362.0600	6.79574	1000	1000	0	10.62	0.0	0.0
3	244.9975	5.30625	1000	2	0	10.86	0.0	0.0
4	373.3800	51.50255	1000	1000	0	15.47	0.0	0.0
5	324.2350	15.18283	1000	1000	0	12.82	0.0	0.0
6	316.7250	18.86073	1000	1000	0	10.04	0.0	0.0
7	210.4500	7.82284	1000	2	0	11.20	0.0	100.0
8	205.6675	2.46459	1000	3	0	12.13	0.0	100.0
9	291.9825	0.92416	1000	1000	0	13.55	0.0	0.0
10	322.7425	5.66457	1000	1000	0	14.04	0.0	0.0
11	343.3250	39.49732	1000	1000	0	10.23	0.0	0.0

Table C.16: PBT_CG_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	548.3225	10.22576	1000	1000	0	9.85	0.0	0.0
2	355.8225	5.59571	1000	1000	0	10.62	0.0	0.0
3	240.4325	3.87606	1000	2	0	10.86	0.0	0.0
4	345.7525	7.50886	1000	1000	0	15.47	0.0	0.0
5	314.7150	2.17962	1000	1000	0	12.82	0.0	0.0
6	305.1125	2.84721	1000	1000	0	10.04	0.0	0.0
7	206.3250	1.22964	1000	2	0	11.20	0.0	100.0
8	202.9675	2.06015	1000	3	0	12.13	0.0	100.0
9	295.3425	3.30466	1000	1000	0	13.55	0.0	0.0
10	330.1050	6.34059	1000	1000	0	14.04	0.0	0.0
11	338.6700	0.72873	1000	1000	0	10.23	0.0	0.0

Table C.17: PBT_CG_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	547.5350	13.07337	1000	1000	0	9.85	0.0	0.0
2	358.8000	7.18907	1000	1000	0	10.62	0.0	0.0
3	245.1075	2.51349	1000	2	0	10.86	0.0	0.0
4	344.3025	5.06944	1000	1000	0	15.47	0.0	0.0
5	321.1700	6.31871	1000	1000	0	12.82	0.0	0.0
6	317.5225	6.93276	1000	1000	0	10.04	0.0	0.0
7	219.2450	7.20810	1000	2	0	11.20	0.0	100.0
8	208.5150	3.09639	1000	3	0	12.13	0.0	100.0
9	296.5225	3.16256	1000	1000	0	13.55	0.0	0.0
10	319.8025	6.86141	1000	1000	0	14.04	0.0	0.0
11	329.2200	3.71652	1000	1000	0	10.23	0.0	0.0

Table C.18: PBT_CG_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	541.8025	3.57376	1000	1000	0	9.85	0.0	0.0
2	361.6575	4.94127	1000	1000	0	10.62	0.0	0.0
3	243.6300	5.38011	1000	2	0	10.86	0.0	0.0
4	350.2150	3.06855	1000	1000	0	15.47	0.0	0.0
5	330.2150	12.54419	1000	1000	0	12.82	0.0	0.0
6	316.6425	9.82439	1000	1000	0	10.04	0.0	0.0
7	209.0600	0.64393	1000	2	0	11.20	0.0	100.0
8	205.5575	3.75359	1000	3	0	12.13	0.0	100.0
9	296.6950	3.26501	1000	1000	0	13.55	0.0	0.0
10	322.1700	8.04239	1000	1000	0	14.04	0.0	0.0
11	331.7075	5.61941	1000	1000	0	10.23	0.0	0.0

C.2 M1

Table C.19: M1 - Search strategies results.

Strategy	Min Mean Time (ms)	Std Min Mean Time (ms)	Max Mean Time (ms)	Std Max Mean Time (ms)	Min Sample Size	Max Sample Size	Min Mean Test Cases	Std Min Mean Test Cases	Max Mean Test Cases	Std Max Mean Test Cases	Min Mean Coverage (%)	Std Min Mean Coverage (%)	Max Mean Coverage (%)	Std Max Mean Coverage (%)	Bug Found
PBT_R_S4	3.05842	0.60081	3.05842	0.60081	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S1	2.63213	0.04970	2.63213	0.04970	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S8	3.49505	0.01615	3.49505	0.01615	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_CG_S4	208.94250	1.27125	208.94250	1.27125	1000	1000	1	0	1	0	11.24	0.0	11.24	0.0	True
PBT_CG_S6	207.52000	2.25041	207.52000	2.25041	1000	1000	1	0	1	0	11.24	0.0	11.24	0.0	True
PBT_CG_S3	210.36500	0.91067	210.36500	0.91067	1000	1000	1	0	1	0	11.24	0.0	11.24	0.0	True
PBT_R_S3	4.07645	0.26249	4.07645	0.26249	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S5	4.27763	0.54551	4.27763	0.54551	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S7	3.68562	0.18255	3.68562	0.18255	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_CG_S5	215.67500	6.65804	215.67500	6.65804	1000	1000	1	0	1	0	11.24	0.0	11.24	0.0	True
PBT_CG_S7	208.64750	2.09961	208.64750	2.09961	1000	1000	1	0	1	0	11.24	0.0	11.24	0.0	True
PBT_CG_S1	204.67250	3.31402	204.67250	3.31402	1000	1000	1	0	1	0	11.24	0.0	11.24	0.0	True
PBT_CG_S8	208.42750	3.39855	208.42750	3.39855	1000	1000	1	0	1	0	11.24	0.0	11.24	0.0	True
PBT_CG_S2	206.80250	1.95270	206.80250	1.95270	1000	1000	1	0	1	0	11.24	0.0	11.24	0.0	True
PBT_R_S6	2.55607	0.01908	2.55607	0.01908	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S2	2.68635	0.16044	2.68635	0.16044	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True

Table C.20: M1 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.1094	0.8439	-1.1453	1.3641	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.2193	0.6942	-1.0209	1.4596	False
Random withConstantExtraction	Random withoutConstantExtraction	-1.1504	0.0016	-1.6657	-0.6352	True
Coverage-Guided withBias	Coverage-Guided withoutBias	-1.4494	0.5751	-7.2873	4.3886	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	0.9944	0.7097	-4.9379	6.9266	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-3.7944	0.0940	-8.4646	0.8759	False
Coverage-Guided	Random	-205.5731	0.0010	-208.0666	-203.0796	True

C.2.1 Search Strategies

Table C.21: PBT_R_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	8.61212	0.48255	1000	1000	0	0.0	0.0	0.0
2	14.08550	0.28524	1000	1000	0	0.0	0.0	0.0
3	6.50960	0.42725	1000	55	68	0.0	0.0	0.0
4	15.57075	0.38823	1000	1000	0	0.0	0.0	0.0
5	9.80922	0.51128	1000	1000	0	0.0	0.0	0.0
6	9.11835	0.26543	1000	1000	0	0.0	0.0	0.0
7	2.63213	0.04970	1000	1	0	0.0	0.0	100.0
8	2.65835	0.07824	1000	1000	0	0.0	0.0	0.0
9	5.30127	0.06413	1000	1000	0	0.0	0.0	0.0
10	8.79610	1.04686	1000	1000	0	0.0	0.0	0.0
11	10.62510	1.08759	1000	1000	0	0.0	0.0	0.0

Table C.22: PBT_R_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	8.34705	0.06990	1000	1000	0	0.0	0.0	0.0
2	14.02825	0.11722	1000	1000	0	0.0	0.0	0.0
3	6.47283	0.45752	1000	2	0	0.0	0.0	0.0
4	16.83150	1.99267	1000	1000	0	0.0	0.0	0.0
5	9.22495	0.32697	1000	1000	0	0.0	0.0	0.0
6	8.91607	0.51524	1000	1000	0	0.0	0.0	0.0
7	2.68635	0.16044	1000	1	0	0.0	0.0	100.0
8	2.77038	0.21979	1000	1000	0	0.0	0.0	0.0
9	5.20957	0.12454	1000	1000	0	0.0	0.0	0.0
10	8.32438	0.09223	1000	1000	0	0.0	0.0	0.0
11	9.83253	0.13676	1000	1000	0	0.0	0.0	0.0

Table C.23: PBT_R_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14.32425	0.32956	1000	1000	0	0.0	0.0	0.0
2	17.55425	0.53690	1000	1000	0	0.0	0.0	0.0
3	8.09292	0.43803	1000	2	0	0.0	0.0	0.0
4	16.13550	0.12315	1000	1000	0	0.0	0.0	0.0
5	11.96925	0.30819	1000	1000	0	0.0	0.0	0.0
6	9.68602	0.27561	1000	1000	0	0.0	0.0	0.0
7	4.07645	0.26249	1000	1	0	0.0	0.0	100.0
8	4.17685	0.01831	1000	1000	0	0.0	0.0	0.0
9	7.33142	0.29833	1000	1000	0	0.0	0.0	0.0
10	8.25347	0.12893	1000	1000	0	0.0	0.0	0.0
11	11.86500	0.36708	1000	1000	0	0.0	0.0	0.0

Table C.24: PBT_R_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	8.57007	0.50238	1000	1000	0	0.0	0.0	0.0
2	13.93475	0.24703	1000	1000	0	0.0	0.0	0.0
3	6.07480	0.06895	1000	20	12	0.0	0.0	0.0
4	15.51925	0.36744	1000	1000	0	0.0	0.0	0.0
5	9.45805	0.14507	1000	1000	0	0.0	0.0	0.0
6	8.85823	0.59917	1000	1000	0	0.0	0.0	0.0
7	3.05842	0.60081	1000	1	0	0.0	0.0	100.0
8	2.57215	0.06359	1000	1000	0	0.0	0.0	0.0
9	5.19273	0.18563	1000	1000	0	0.0	0.0	0.0
10	8.13652	0.48550	1000	1000	0	0.0	0.0	0.0
11	10.83550	0.62704	1000	1000	0	0.0	0.0	0.0

Table C.25: PBT_R_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14.30275	0.08840	1000	1000	0	0.0	0.0	0.0
2	17.44400	0.18746	1000	1000	0	0.0	0.0	0.0
3	8.27960	0.85713	1000	2	0	0.0	0.0	0.0
4	16.51625	0.44298	1000	1000	0	0.0	0.0	0.0
5	11.64250	0.52440	1000	1000	0	0.0	0.0	0.0
6	10.22888	0.89122	1000	1000	0	0.0	0.0	0.0
7	4.27763	0.54551	1000	1	0	0.0	0.0	100.0
8	4.35807	0.13234	1000	1000	0	0.0	0.0	0.0
9	8.08980	0.63531	1000	1000	0	0.0	0.0	0.0
10	8.31090	0.18576	1000	1000	0	0.0	0.0	0.0
11	12.39775	1.01118	1000	1000	0	0.0	0.0	0.0

Table C.26: PBT_R_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	8.61947	0.46944	1000	1000	0	0.0	0.0	0.0
2	14.05000	0.28485	1000	1000	0	0.0	0.0	0.0
3	6.37348	0.29936	1000	2	0	0.0	0.0	0.0
4	15.36225	0.47234	1000	1000	0	0.0	0.0	0.0
5	9.40682	0.36134	1000	1000	0	0.0	0.0	0.0
6	8.62215	0.43399	1000	1000	0	0.0	0.0	0.0
7	2.55607	0.01908	1000	1	0	0.0	0.0	100.0
8	2.62850	0.01700	1000	1000	0	0.0	0.0	0.0
9	5.18615	0.04505	1000	1000	0	0.0	0.0	0.0
10	8.13325	0.36124	1000	1000	0	0.0	0.0	0.0
11	10.61647	0.99343	1000	1000	0	0.0	0.0	0.0

Table C.27: PBT_R_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14.15975	0.13185	1000	1000	0	0.0	0.0	0.0
2	17.51925	0.35998	1000	1000	0	0.0	0.0	0.0
3	8.08310	0.37173	1000	2	0	0.0	0.0	0.0
4	16.53850	0.32501	1000	1000	0	0.0	0.0	0.0
5	12.04125	0.38082	1000	1000	0	0.0	0.0	0.0
6	9.77140	0.53359	1000	1000	0	0.0	0.0	0.0
7	3.68562	0.18255	1000	1	0	0.0	0.0	100.0
8	3.60420	0.11528	1000	1000	0	0.0	0.0	0.0
9	6.66310	0.33124	1000	1000	0	0.0	0.0	0.0
10	7.25403	0.14058	1000	1000	0	0.0	0.0	0.0
11	11.71000	0.16144	1000	1000	0	0.0	0.0	0.0

Table C.28: PBT_R_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14.39850	0.37413	1000	1000	0	0.0	0.0	0.0
2	18.29150	1.10191	1000	1000	0	0.0	0.0	0.0
3	8.49080	0.71991	1000	2	0	0.0	0.0	0.0
4	17.18925	1.59999	1000	1000	0	0.0	0.0	0.0
5	11.38700	0.31582	1000	1000	0	0.0	0.0	0.0
6	9.50618	0.42158	1000	1000	0	0.0	0.0	0.0
7	3.49505	0.01615	1000	1	0	0.0	0.0	100.0
8	3.50338	0.05311	1000	1000	0	0.0	0.0	0.0
9	6.50335	0.22848	1000	1000	0	0.0	0.0	0.0
10	7.61268	0.86208	1000	1000	0	0.0	0.0	0.0
11	11.92500	0.17617	1000	1000	0	0.0	0.0	0.0

Table C.29: PBT_CG_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	525.3475	7.94073	1000	1000	0	9.89	0.0	0.0
2	354.4000	8.78122	1000	1000	0	10.66	0.0	0.0
3	245.1000	8.62331	1000	16	13	11.79	0.0	0.0
4	344.5100	3.44577	1000	1000	0	15.53	0.0	0.0
5	317.8225	3.50496	1000	1000	0	12.87	0.0	0.0
6	305.6600	2.81756	1000	1000	0	10.08	0.0	0.0
7	204.6725	3.31402	1000	1	0	11.24	0.0	100.0
8	283.2125	4.91489	1000	1000	0	12.18	0.0	0.0
9	304.1100	6.93548	1000	1000	0	13.60	0.0	0.0
10	326.1500	7.16469	1000	1000	0	14.09	0.0	0.0
11	359.8750	42.95185	1000	1000	0	10.27	0.0	0.0

Table C.30: PBT_CG_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	534.2825	2.68884	1000	1000	0	9.89	0.0	0.0
2	358.2650	3.54872	1000	1000	0	10.66	0.0	0.0
3	252.7000	8.31411	1000	2	0	10.90	0.0	0.0
4	351.6825	1.29432	1000	1000	0	15.53	0.0	0.0
5	316.6375	2.67067	1000	1000	0	12.87	0.0	0.0
6	304.6425	4.73428	1000	1000	0	10.08	0.0	0.0
7	206.8025	1.95270	1000	1	0	11.24	0.0	100.0
8	280.3925	1.98274	1000	1000	0	12.18	0.0	0.0
9	299.3275	0.44268	1000	1000	0	13.60	0.0	0.0
10	331.0175	1.33618	1000	1000	0	14.09	0.0	0.0
11	342.6925	24.01583	1000	1000	0	10.27	0.0	0.0

Table C.31: PBT_CG_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	547.2525	4.82217	1000	1000	0	9.89	0.0	0.0
2	356.7200	10.94127	1000	1000	0	10.66	0.0	0.0
3	244.6550	5.61461	1000	2	0	10.90	0.0	0.0
4	355.6025	7.60589	1000	1000	0	15.53	0.0	0.0
5	324.1200	8.78382	1000	1000	0	12.87	0.0	0.0
6	309.9200	3.82307	1000	1000	0	10.08	0.0	0.0
7	210.3650	0.91067	1000	1	0	11.24	0.0	100.0
8	287.3225	2.97428	1000	1000	0	12.18	0.0	0.0
9	294.9950	1.57201	1000	1000	0	13.60	0.0	0.0
10	327.0425	6.51323	1000	1000	0	14.09	0.0	0.0
11	352.2650	27.93592	1000	1000	0	10.27	0.0	0.0

Table C.32: PBT_CG_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	535.6725	6.24095	1000	1000	0	9.89	0.0	0.0
2	360.2400	6.34969	1000	1000	0	10.66	0.0	0.0
3	249.3175	5.19930	1000	37	35	11.79	0.0	0.0
4	350.5350	6.39050	1000	1000	0	15.53	0.0	0.0
5	327.0150	8.02940	1000	1000	0	12.87	0.0	0.0
6	315.4975	9.85462	1000	1000	0	10.08	0.0	0.0
7	208.9425	1.27125	1000	1	0	11.24	0.0	100.0
8	285.8250	8.02167	1000	1000	0	12.18	0.0	0.0
9	294.3800	2.03675	1000	1000	0	13.60	0.0	0.0
10	322.5525	6.45235	1000	1000	0	14.09	0.0	0.0
11	346.4675	25.95737	1000	1000	0	10.27	0.0	0.0

Table C.33: PBT_CG_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	540.4275	5.85921	1000	1000	0	9.89	0.0	0.0
2	359.2450	6.27023	1000	1000	0	10.66	0.0	0.0
3	244.9000	1.92111	1000	2	0	10.90	0.0	0.0
4	356.2175	6.33406	1000	1000	0	15.53	0.0	0.0
5	319.1925	2.55715	1000	1000	0	12.87	0.0	0.0
6	307.0800	3.69274	1000	1000	0	10.08	0.0	0.0
7	215.6750	6.65804	1000	1	0	11.24	0.0	100.0
8	289.7300	2.58986	1000	1000	0	12.18	0.0	0.0
9	294.6025	4.10145	1000	1000	0	13.60	0.0	0.0
10	326.4300	5.63877	1000	1000	0	14.09	0.0	0.0
11	334.8100	7.80739	1000	1000	0	10.27	0.0	0.0

Table C.34: PBT_CG_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	533.1375	4.66595	1000	1000	0	9.89	0.0	0.0
2	355.2250	9.58781	1000	1000	0	10.66	0.0	0.0
3	240.3200	5.58601	1000	2	0	10.90	0.0	0.0
4	345.7450	7.69183	1000	1000	0	15.53	0.0	0.0
5	311.9175	1.40340	1000	1000	0	12.87	0.0	0.0
6	307.0800	7.08880	1000	1000	0	10.08	0.0	0.0
7	207.5200	2.25041	1000	1	0	11.24	0.0	100.0
8	281.0650	3.38415	1000	1000	0	12.18	0.0	0.0
9	297.0325	4.99465	1000	1000	0	13.60	0.0	0.0
10	325.5025	13.27388	1000	1000	0	14.09	0.0	0.0
11	361.1800	22.57080	1000	1000	0	10.27	0.0	0.0

Table C.35: PBT_CG_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	533.9425	6.12342	1000	1000	0	9.89	0.0	0.0
2	358.5775	8.00361	1000	1000	0	10.66	0.0	0.0
3	244.9775	6.33598	1000	2	0	10.90	0.0	0.0
4	356.2900	6.08776	1000	1000	0	15.53	0.0	0.0
5	320.1575	5.03171	1000	1000	0	12.87	0.0	0.0
6	308.4425	2.40197	1000	1000	0	10.08	0.0	0.0
7	208.6475	2.09961	1000	1	0	11.24	0.0	100.0
8	286.1075	9.82417	1000	1000	0	12.18	0.0	0.0
9	295.5325	3.48016	1000	1000	0	13.60	0.0	0.0
10	322.9600	6.06499	1000	1000	0	14.09	0.0	0.0
11	331.6575	4.57426	1000	1000	0	10.27	0.0	0.0

Table C.36: PBT_CG_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	537.0475	5.61908	1000	1000	0	9.89	0.0	0.0
2	361.5300	3.91024	1000	1000	0	10.66	0.0	0.0
3	242.8975	3.70557	1000	2	0	10.90	0.0	0.0
4	345.4725	2.97758	1000	1000	0	15.53	0.0	0.0
5	313.5825	2.54818	1000	1000	0	12.87	0.0	0.0
6	305.1125	6.79059	1000	1000	0	10.08	0.0	0.0
7	208.4275	3.39855	1000	1	0	11.24	0.0	100.0
8	281.3650	1.44766	1000	1000	0	12.18	0.0	0.0
9	296.5850	4.48606	1000	1000	0	13.60	0.0	0.0
10	333.0600	12.15314	1000	1000	0	14.09	0.0	0.0
11	350.6775	22.23166	1000	1000	0	10.27	0.0	0.0

C.3 M2

Table C.37: M2 - Search strategies results.

Strategy	Min Mean Time (ms)	Std Min Mean Time (ms)	Max Mean Time (ms)	Std Max Mean Time (ms)	Min Sample Size	Max Sample Size	Min Mean Test Cases	Std Min Mean Test Cases	Max Mean Test Cases	Std Max Mean Test Cases	Min Mean Coverage (%)	Std Min Mean Coverage (%)	Max Mean Coverage (%)	Std Max Mean Coverage (%)	Bug Found
PBT_R_S4	0.63360	0.01495	6.39300	2.06803	1000	1000	1	0	2	0	0.00	0.0	0.00	0.0	True
PBT_R_S1	0.64528	0.03110	7.32685	1.72655	1000	1000	1	0	2	1	0.00	0.0	0.00	0.0	True
PBT_R_S8	1.55217	0.03215	10.20850	0.11209	1000	1000	1	0	5	0	0.00	0.0	0.00	0.0	True
PBT_CG_S4	204.16750	1.85691	442.76000	20.24550	1000	1000	1	0	2	0	9.89	0.0	15.53	0.0	True
PBT_CG_S6	206.81000	1.24220	425.73750	6.10911	1000	1000	1	0	5	0	9.89	0.0	15.53	0.0	True
PBT_CG_S3	205.86750	1.74939	438.05000	6.52225	1000	1000	1	0	5	0	9.89	0.0	15.53	0.0	True
PBT_R_S3	1.74505	0.02905	10.43800	0.17596	1000	1000	1	0	5	0	0.00	0.0	0.00	0.0	True
PBT_R_S5	1.86470	0.06300	10.88275	0.56526	1000	1000	1	0	5	0	0.00	0.0	0.00	0.0	True
PBT_R_S7	2.11092	0.50478	10.75200	0.64402	1000	1000	1	0	5	0	0.00	0.0	0.00	0.0	True
PBT_CG_S5	205.69500	0.83614	425.44250	6.62696	1000	1000	1	0	5	0	9.89	0.0	15.53	0.0	True
PBT_CG_S7	204.12750	2.78844	435.90500	15.34837	1000	1000	1	0	5	0	9.89	0.0	15.53	0.0	True
PBT_CG_S1	205.12500	4.89118	412.88750	2.90105	1000	1000	1	0	2	1	9.89	0.0	15.53	0.0	True
PBT_CG_S8	203.97750	1.40745	428.94500	3.50691	1000	1000	1	0	5	0	9.89	0.0	15.53	0.0	True
PBT_CG_S2	203.66000	1.42074	442.51250	13.28248	1000	1000	1	0	5	0	9.89	0.0	15.53	0.0	True
PBT_R_S6	0.63892	0.01878	5.99923	0.14250	1000	1000	1	0	5	0	0.00	0.0	0.00	0.0	True
PBT_R_S2	0.65129	0.00791	7.16485	1.74900	1000	1000	1	0	5	0	0.00	0.0	0.00	0.0	True

Table C.38: M2 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.1069	0.8412	-1.0974	1.3113	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-0.0073	0.9000	-1.2164	1.2017	False
Random withConstantExtraction	Random withoutConstantExtraction	-1.1759	0.0010	-1.4621	-0.8898	True
Coverage-Guided withBias	Coverage-Guided withoutBias	-0.2138	0.8181	-2.2961	1.8686	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	0.3162	0.7343	-1.7531	2.3856	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	0.0237	0.9000	-2.0694	2.1169	False
Coverage-Guided	Random	-203.6985	0.0010	-204.6794	-202.7177	True

C.3.1 Search Strategies

Table C.39: PBT_R_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2.26640	0.04877	1000	1	0	0.0	0.0	100.0
2	7.32685	1.72655	1000	1	0	0.0	0.0	100.0
3	5.90280	0.21254	1000	1	0	0.0	0.0	0.0
4	5.20920	0.45310	1000	2	1	0.0	0.0	100.0
5	5.40673	0.50016	1000	1	0	0.0	0.0	100.0
6	3.03377	0.14123	1000	1	0	0.0	0.0	100.0
7	2.57935	0.14054	1000	2	0	0.0	0.0	100.0
8	0.64528	0.03110	1000	2	0	0.0	0.0	100.0
9	1.99968	0.02786	1000	1	0	0.0	0.0	100.0
10	4.71152	0.27314	1000	1	0	0.0	0.0	100.0
11	3.81725	0.22492	1000	1	0	0.0	0.0	100.0

Table C.40: PBT_R_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2.29413	0.03186	1000	1	0	0.0	0.0	100.0
2	6.41680	0.60010	1000	2	0	0.0	0.0	100.0
3	5.69793	0.08632	1000	2	0	0.0	0.0	0.0
4	7.16485	1.74900	1000	5	0	0.0	0.0	100.0
5	5.48100	0.20258	1000	1	0	0.0	0.0	100.0
6	3.08887	0.05350	1000	1	0	0.0	0.0	100.0
7	2.63478	0.05420	1000	2	0	0.0	0.0	100.0
8	0.65129	0.00791	1000	2	0	0.0	0.0	100.0
9	2.09715	0.06560	1000	1	0	0.0	0.0	100.0
10	4.39907	0.16124	1000	1	0	0.0	0.0	100.0
11	3.94558	0.14629	1000	1	0	0.0	0.0	100.0

Table C.41: PBT_R_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	8.14040	0.19812	1000	1	0	0.0	0.0	100.0
2	10.43800	0.17596	1000	2	0	0.0	0.0	100.0
3	7.39358	0.41013	1000	2	0	0.0	0.0	0.0
4	6.39142	0.27869	1000	5	0	0.0	0.0	100.0
5	7.21000	0.14288	1000	1	0	0.0	0.0	100.0
6	4.03100	0.06631	1000	1	0	0.0	0.0	100.0
7	3.86547	0.05762	1000	2	0	0.0	0.0	100.0
8	1.74505	0.02905	1000	2	0	0.0	0.0	100.0
9	3.53702	0.05665	1000	1	0	0.0	0.0	100.0
10	3.35408	0.06370	1000	1	0	0.0	0.0	100.0
11	4.91635	0.20649	1000	1	0	0.0	0.0	100.0

Table C.42: PBT_R_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2.21442	0.03534	1000	1	0	0.0	0.0	100.0
2	6.09398	0.22926	1000	1	0	0.0	0.0	100.0
3	5.56173	0.04715	1000	1	0	0.0	0.0	0.0
4	6.39300	2.06803	1000	2	0	0.0	0.0	100.0
5	5.16205	0.32463	1000	1	0	0.0	0.0	100.0
6	2.92568	0.09396	1000	1	0	0.0	0.0	100.0
7	2.53543	0.07125	1000	2	0	0.0	0.0	100.0
8	0.63360	0.01495	1000	2	0	0.0	0.0	100.0
9	1.92140	0.07071	1000	1	0	0.0	0.0	100.0
10	4.42545	0.13845	1000	1	0	0.0	0.0	100.0
11	3.74050	0.12909	1000	1	0	0.0	0.0	100.0

Table C.43: PBT_R_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	8.17643	0.16853	1000	1	0	0.0	0.0	100.0
2	10.88275	0.56526	1000	2	0	0.0	0.0	100.0
3	8.43435	1.13596	1000	2	0	0.0	0.0	0.0
4	6.73477	0.55612	1000	5	0	0.0	0.0	100.0
5	6.94540	0.11420	1000	1	0	0.0	0.0	100.0
6	4.05987	0.08969	1000	1	0	0.0	0.0	100.0
7	4.00755	0.08738	1000	2	0	0.0	0.0	100.0
8	1.86470	0.06300	1000	2	0	0.0	0.0	100.0
9	3.53665	0.08878	1000	1	0	0.0	0.0	100.0
10	3.18140	0.14303	1000	1	0	0.0	0.0	100.0
11	5.05030	0.08664	1000	1	0	0.0	0.0	100.0

Table C.44: PBT_R_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2.27237	0.02707	1000	1	0	0.0	0.0	100.0
2	5.99923	0.14250	1000	2	0	0.0	0.0	100.0
3	6.04383	0.24260	1000	2	0	0.0	0.0	0.0
4	5.04855	0.08726	1000	5	0	0.0	0.0	100.0
5	5.16092	0.10402	1000	1	0	0.0	0.0	100.0
6	3.01430	0.05511	1000	1	0	0.0	0.0	100.0
7	2.56668	0.04696	1000	2	0	0.0	0.0	100.0
8	0.63892	0.01878	1000	2	0	0.0	0.0	100.0
9	2.08785	0.03945	1000	1	0	0.0	0.0	100.0
10	4.61032	0.06862	1000	1	0	0.0	0.0	100.0
11	3.82705	0.14487	1000	1	0	0.0	0.0	100.0

Table C.45: PBT_R_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	8.07520	0.04252	1000	1	0	0.0	0.0	100.0
2	10.75200	0.64402	1000	2	0	0.0	0.0	100.0
3	7.68742	0.90380	1000	2	0	0.0	0.0	0.0
4	6.67265	0.36839	1000	5	0	0.0	0.0	100.0
5	6.69120	0.42773	1000	1	0	0.0	0.0	100.0
6	3.96270	0.12262	1000	1	0	0.0	0.0	100.0
7	4.25770	0.57614	1000	2	0	0.0	0.0	100.0
8	2.11092	0.50478	1000	2	0	0.0	0.0	100.0
9	2.86090	0.08703	1000	1	0	0.0	0.0	100.0
10	3.36408	0.45773	1000	1	0	0.0	0.0	100.0
11	5.39630	1.13114	1000	1	0	0.0	0.0	100.0

Table C.46: PBT_R_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	8.02802	0.09845	1000	1	0	0.0	0.0	100.0
2	10.20850	0.11209	1000	2	0	0.0	0.0	100.0
3	7.63185	0.23777	1000	2	0	0.0	0.0	0.0
4	7.46813	1.24940	1000	5	0	0.0	0.0	100.0
5	6.73067	0.12799	1000	1	0	0.0	0.0	100.0
6	3.94477	0.08943	1000	1	0	0.0	0.0	100.0
7	3.54153	0.07644	1000	2	0	0.0	0.0	100.0
8	1.55217	0.03215	1000	2	0	0.0	0.0	100.0
9	2.93167	0.11427	1000	1	0	0.0	0.0	100.0
10	3.09505	0.13021	1000	1	0	0.0	0.0	100.0
11	4.75978	0.21178	1000	1	0	0.0	0.0	100.0

Table C.47: PBT_CG_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	412.8875	2.90105	1000	1	0	9.89	0.00	100.0
2	245.1125	5.38712	1000	1	0	10.66	0.00	100.0
3	240.4325	3.09244	1000	1	0	11.57	0.39	0.0
4	234.5325	6.62969	1000	2	1	15.53	0.00	100.0
5	218.8575	2.74745	1000	1	0	12.87	0.00	100.0
6	210.0950	0.81310	1000	1	0	10.08	0.00	100.0
7	212.4650	8.11384	1000	2	0	11.24	0.00	100.0
8	205.1250	4.89118	1000	2	0	12.18	0.00	100.0
9	213.8175	8.83131	1000	1	0	13.60	0.00	100.0
10	210.4250	2.63917	1000	1	0	10.27	0.00	100.0

Table C.48: PBT_CG_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	442.5125	13.28248	1000	1	0	9.89	0.0	100.0
2	245.7925	7.98479	1000	2	0	10.66	0.0	100.0
3	237.8125	2.00351	1000	2	0	10.90	0.0	0.0
4	232.8400	3.38467	1000	5	0	15.53	0.0	100.0
5	216.5050	2.16761	1000	1	0	12.87	0.0	100.0
6	212.1750	1.06805	1000	1	0	10.08	0.0	100.0
7	209.3025	1.70175	1000	2	0	11.24	0.0	100.0
8	203.6600	1.42074	1000	2	0	12.18	0.0	100.0
9	210.6550	1.39478	1000	1	0	13.60	0.0	100.0
10	221.8175	18.48144	1000	1	0	10.27	0.0	100.0

Table C.49: PBT_CG_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	438.0500	6.52225	1000	1	0	9.89	0.0	100.0
2	247.7900	4.07890	1000	2	0	10.66	0.0	100.0
3	236.7925	1.01846	1000	2	0	10.90	0.0	0.0
4	235.5125	1.40646	1000	5	0	15.53	0.0	100.0
5	221.8025	6.84880	1000	1	0	12.87	0.0	100.0
6	214.8675	2.80322	1000	1	0	10.08	0.0	100.0
7	211.1600	2.71635	1000	2	0	11.24	0.0	100.0
8	205.8675	1.74939	1000	2	0	12.18	0.0	100.0
9	217.6975	8.78187	1000	1	0	13.60	0.0	100.0
10	209.9500	1.06485	1000	1	0	10.27	0.0	100.0

Table C.50: PBT_CG_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	442.7600	20.24550	1000	1	0	9.89	0.0	100.0
2	251.4425	10.56842	1000	1	0	10.66	0.0	100.0
3	239.3400	3.56168	1000	1	0	11.79	0.0	0.0
4	230.8050	1.91398	1000	2	0	15.53	0.0	100.0
5	217.2775	1.80640	1000	1	0	12.87	0.0	100.0
6	213.1875	3.41804	1000	1	0	10.08	0.0	100.0
7	210.1350	1.73359	1000	2	0	11.24	0.0	100.0
8	204.1675	1.85691	1000	2	0	12.18	0.0	100.0
9	213.5475	3.33863	1000	1	0	13.60	0.0	100.0
10	208.6125	2.89707	1000	1	0	10.27	0.0	100.0

Table C.51: PBT_CG_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	425.4425	6.62696	1000	1	0	9.89	0.0	100.0
2	247.0650	1.19441	1000	2	0	10.66	0.0	100.0
3	247.1925	5.89606	1000	2	0	10.90	0.0	0.0
4	241.5500	6.02084	1000	5	0	15.53	0.0	100.0
5	221.3800	5.51608	1000	1	0	12.87	0.0	100.0
6	213.6850	3.41600	1000	1	0	10.08	0.0	100.0
7	211.6150	2.86730	1000	2	0	11.24	0.0	100.0
8	205.6950	0.83614	1000	2	0	12.18	0.0	100.0
9	212.7325	1.99165	1000	1	0	13.60	0.0	100.0
10	209.9375	0.70190	1000	1	0	10.27	0.0	100.0

Table C.52: PBT_CG_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	425.7375	6.10911	1000	1	0	9.89	0.0	100.0
2	243.0175	3.34163	1000	2	0	10.66	0.0	100.0
3	246.3550	9.32165	1000	2	0	10.90	0.0	0.0
4	235.2800	1.81545	1000	5	0	15.53	0.0	100.0
5	218.2475	2.02160	1000	1	0	12.87	0.0	100.0
6	214.9050	2.64577	1000	1	0	10.08	0.0	100.0
7	213.7700	3.49001	1000	2	0	11.24	0.0	100.0
8	208.0925	6.45224	1000	2	0	12.18	0.0	100.0
9	209.0925	2.06392	1000	1	0	13.60	0.0	100.0
10	206.8100	1.24220	1000	1	0	10.27	0.0	100.0

Table C.53: PBT_CG_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	435.9050	15.34837	1000	1	0	9.89	0.0	100.0
2	247.6775	10.23683	1000	2	0	10.66	0.0	100.0
3	241.0150	5.63147	1000	2	0	10.90	0.0	0.0
4	236.7775	3.02489	1000	5	0	15.53	0.0	100.0
5	220.0950	3.39493	1000	1	0	12.87	0.0	100.0
6	212.1825	3.93541	1000	1	0	10.08	0.0	100.0
7	207.8675	1.87386	1000	2	0	11.24	0.0	100.0
8	204.1275	2.78844	1000	2	0	12.18	0.0	100.0
9	208.8425	2.79929	1000	1	0	13.60	0.0	100.0
10	206.7050	2.41016	1000	1	0	10.27	0.0	100.0

Table C.54: PBT_CG_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	428.9450	3.50691	1000	1	0	9.89	0.0	100.0
2	248.6975	4.05443	1000	2	0	10.66	0.0	100.0
3	240.6650	1.91103	1000	2	0	10.90	0.0	0.0
4	234.6475	1.11131	1000	5	0	15.53	0.0	100.0
5	225.0575	8.41833	1000	1	0	12.87	0.0	100.0
6	211.3850	2.85493	1000	1	0	10.08	0.0	100.0
7	209.9350	1.41298	1000	2	0	11.24	0.0	100.0
8	203.9775	1.40745	1000	2	0	12.18	0.0	100.0
9	210.5875	2.89339	1000	1	0	13.60	0.0	100.0
10	207.6800	2.81934	1000	1	0	10.27	0.0	100.0

C.4 M3

Table C.55: M3 - Search strategies results.

Strategy	Min Mean Time (ms)	Std Min Mean Time (ms)	Max Mean Time (ms)	Std Max Mean Time (ms)	Min Sample Size	Max Sample Size	Min Mean Test Cases	Std Min Mean Test Cases	Max Mean Test Cases	Std Max Mean Test Cases	Min Mean Coverage (%)	Std Min Mean Coverage (%)	Max Mean Coverage (%)	Std Max Mean Coverage (%)	Bug Found
PBT_R_S4	0.49240	0.05230	2.96480	0.63653	1000	1000	1	0	3	0	0.0	0.0	0.00	0.0	True
PBT_R_S1	0.50014	0.03350	2.53262	0.07888	1000	1000	1	0	3	0	0.0	0.0	0.00	0.0	True
PBT_R_S8	1.46083	0.08307	3.59562	0.01755	1000	1000	1	0	3	0	0.0	0.0	0.00	0.0	True
PBT_CG_S4	207.82750	1.24162	211.22500	2.65300	1000	1000	1	0	3	0	11.2	0.0	13.55	0.0	True
PBT_CG_S6	206.62250	2.60240	237.70500	47.16166	1000	1000	1	0	3	0	11.2	0.0	13.55	0.0	True
PBT_CG_S3	210.26250	4.26161	216.40250	6.04495	1000	1000	1	0	3	0	11.2	0.0	13.55	0.0	True
PBT_R_S3	1.77892	0.07161	4.25962	0.49634	1000	1000	1	0	3	0	0.0	0.0	0.00	0.0	True
PBT_R_S5	1.68945	0.09065	3.98875	0.12078	1000	1000	1	0	3	0	0.0	0.0	0.00	0.0	True
PBT_R_S7	1.46588	0.08954	4.17812	0.74663	1000	1000	1	0	3	0	0.0	0.0	0.00	0.0	True
PBT_CG_S5	206.57250	1.10513	212.72500	2.03682	1000	1000	1	0	3	0	11.2	0.0	13.55	0.0	True
PBT_CG_S7	207.11250	1.90630	211.31750	1.56677	1000	1000	1	0	3	0	11.2	0.0	13.55	0.0	True
PBT_CG_S1	208.08250	3.38345	212.38250	2.93639	1000	1000	1	0	3	0	11.2	0.0	13.55	0.0	True
PBT_CG_S8	209.32250	9.33334	210.94000	2.16383	1000	1000	1	0	3	0	11.2	0.0	13.55	0.0	True
PBT_CG_S2	206.52000	2.13008	211.02500	6.03941	1000	1000	1	0	3	0	11.2	0.0	13.55	0.0	True
PBT_R_S6	0.64653	0.03244	2.78123	0.29594	1000	1000	1	0	3	0	0.0	0.0	0.00	0.0	True
PBT_R_S2	0.58022	0.03993	2.72290	0.26196	1000	1000	1	0	3	0	0.0	0.0	0.00	0.0	True

Table C.56: M3 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-0.0349	0.9000	-1.0993	1.0294	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.1208	0.7989	-0.9373	1.1789	False
Random withConstantExtraction	Random withoutConstantExtraction	-1.0439	0.0010	-1.2597	-0.8282	True
Coverage-Guided withBias	Coverage-Guided withoutBias	1.0619	0.3141	-1.3032	3.4270	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	0.1381	0.9000	-2.4503	2.7265	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-1.0544	0.3178	-3.4228	1.3141	False
Coverage-Guided	Random	-206.7135	0.0010	-207.8506	-205.5765	True

C.4.1 Search Strategies

Table C.57: PBT_R_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	8.95750	0.61871	1000	1000	0	0.0	0.0	0.0
2	14.59850	1.15157	1000	1000	0	0.0	0.0	0.0
3	6.16792	0.37686	1000	5	3	0.0	0.0	0.0
4	14.99225	0.44748	1000	1000	0	0.0	0.0	0.0
5	9.34893	0.39327	1000	1000	0	0.0	0.0	0.0
6	8.64613	0.40592	1000	1000	0	0.0	0.0	0.0
7	2.53262	0.07888	1000	1	0	0.0	0.0	100.0
8	0.50014	0.03350	1000	1	0	0.0	0.0	100.0
9	1.90565	0.05947	1000	3	0	0.0	0.0	100.0
10	7.99862	0.31282	1000	1000	0	0.0	0.0	0.0
11	10.35650	0.32809	1000	1000	0	0.0	0.0	0.0

Table C.58: PBT_R_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	8.43233	0.16576	1000	1000	0	0.0	0.0	0.0
2	13.87200	0.54667	1000	1000	0	0.0	0.0	0.0
3	6.51668	0.52705	1000	2	0	0.0	0.0	0.0
4	15.23875	0.55345	1000	1000	0	0.0	0.0	0.0
5	9.52680	0.58039	1000	1000	0	0.0	0.0	0.0
6	8.92318	0.42138	1000	1000	0	0.0	0.0	0.0
7	2.72290	0.26196	1000	1	0	0.0	0.0	100.0
8	0.58022	0.03993	1000	1	0	0.0	0.0	100.0
9	1.95780	0.06551	1000	3	0	0.0	0.0	100.0
10	8.02890	0.49965	1000	1000	0	0.0	0.0	0.0
11	10.52165	0.33134	1000	1000	0	0.0	0.0	0.0

Table C.59: PBT_R_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14.84175	0.57414	1000	1000	0	0.0	0.0	0.0
2	18.05775	0.72622	1000	1000	0	0.0	0.0	0.0
3	8.21740	0.28077	1000	2	0	0.0	0.0	0.0
4	16.94225	0.13348	1000	1000	0	0.0	0.0	0.0
5	11.92600	0.47943	1000	1000	0	0.0	0.0	0.0
6	9.72878	0.44716	1000	1000	0	0.0	0.0	0.0
7	4.25962	0.49634	1000	1	0	0.0	0.0	100.0
8	1.77892	0.07161	1000	1	0	0.0	0.0	100.0
9	3.67518	0.22966	1000	3	0	0.0	0.0	100.0
10	8.36100	0.62043	1000	1000	0	0.0	0.0	0.0
11	12.64500	0.37539	1000	1000	0	0.0	0.0	0.0

Table C.60: PBT_R_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	8.77435	0.66543	1000	1000	0	0.0	0.0	0.0
2	14.36800	0.37987	1000	1000	0	0.0	0.0	0.0
3	6.69690	1.02031	1000	13	10	0.0	0.0	0.0
4	14.96975	0.32346	1000	1000	0	0.0	0.0	0.0
5	9.93898	0.23857	1000	1000	0	0.0	0.0	0.0
6	8.82988	0.51379	1000	1000	0	0.0	0.0	0.0
7	2.96480	0.63653	1000	1	0	0.0	0.0	100.0
8	0.49240	0.05230	1000	1	0	0.0	0.0	100.0
9	1.92932	0.00716	1000	3	0	0.0	0.0	100.0
10	8.14763	0.08225	1000	1000	0	0.0	0.0	0.0
11	10.49287	0.59790	1000	1000	0	0.0	0.0	0.0

Table C.61: PBT_R_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14.50150	0.30638	1000	1000	0	0.0	0.0	0.0
2	17.74750	0.33848	1000	1000	0	0.0	0.0	0.0
3	8.12885	0.30988	1000	2	0	0.0	0.0	0.0
4	16.74975	0.40755	1000	1000	0	0.0	0.0	0.0
5	11.63500	0.46709	1000	1000	0	0.0	0.0	0.0
6	10.04888	0.58255	1000	1000	0	0.0	0.0	0.0
7	3.98875	0.12078	1000	1	0	0.0	0.0	100.0
8	1.68945	0.09065	1000	1	0	0.0	0.0	100.0
9	3.72680	0.19269	1000	3	0	0.0	0.0	100.0
10	8.88332	0.66277	1000	1000	0	0.0	0.0	0.0
11	12.88275	0.21352	1000	1000	0	0.0	0.0	0.0

Table C.62: PBT_R_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	8.68318	0.49024	1000	1000	0	0.0	0.0	0.0
2	13.95850	0.12857	1000	1000	0	0.0	0.0	0.0
3	6.16290	0.08563	1000	2	0	0.0	0.0	0.0
4	15.17250	0.28085	1000	1000	0	0.0	0.0	0.0
5	9.36670	0.37041	1000	1000	0	0.0	0.0	0.0
6	8.61265	0.47886	1000	1000	0	0.0	0.0	0.0
7	2.78123	0.29594	1000	1	0	0.0	0.0	100.0
8	0.64653	0.03244	1000	1	0	0.0	0.0	100.0
9	1.97395	0.00506	1000	3	0	0.0	0.0	100.0
10	8.16098	0.31659	1000	1000	0	0.0	0.0	0.0
11	10.81750	0.30630	1000	1000	0	0.0	0.0	0.0

Table C.63: PBT_R_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14.84450	0.57596	1000	1000	0	0.0	0.0	0.0
2	18.05550	0.30920	1000	1000	0	0.0	0.0	0.0
3	8.42195	0.73989	1000	2	0	0.0	0.0	0.0
4	16.85525	1.18231	1000	1000	0	0.0	0.0	0.0
5	11.53450	0.41306	1000	1000	0	0.0	0.0	0.0
6	9.91388	0.69684	1000	1000	0	0.0	0.0	0.0
7	4.17812	0.74663	1000	1	0	0.0	0.0	100.0
8	1.46588	0.08954	1000	1	0	0.0	0.0	100.0
9	3.11720	0.43212	1000	3	0	0.0	0.0	100.0
10	8.20438	0.87643	1000	1000	0	0.0	0.0	0.0
11	12.11625	0.68487	1000	1000	0	0.0	0.0	0.0

Table C.64: PBT_R_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14.97175	0.37340	1000	1000	0	0.0	0.0	0.0
2	19.34625	2.20147	1000	1000	0	0.0	0.0	0.0
3	8.82917	1.35152	1000	2	0	0.0	0.0	0.0
4	17.18625	1.25635	1000	1000	0	0.0	0.0	0.0
5	11.47175	0.60943	1000	1000	0	0.0	0.0	0.0
6	9.67402	0.18474	1000	1000	0	0.0	0.0	0.0
7	3.59562	0.01755	1000	1	0	0.0	0.0	100.0
8	1.46083	0.08307	1000	1	0	0.0	0.0	100.0
9	2.93590	0.04664	1000	3	0	0.0	0.0	100.0
10	7.55128	0.30028	1000	1000	0	0.0	0.0	0.0
11	11.69125	0.26793	1000	1000	0	0.0	0.0	0.0

Table C.65: PBT_CG_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	551.4775	30.54440	1000	1000	0	9.85	0.0	0.0
2	355.0550	7.57712	1000	1000	0	10.62	0.0	0.0
3	245.3900	4.35014	1000	13	6	11.75	0.0	0.0
4	351.3225	5.23007	1000	1000	0	15.47	0.0	0.0
5	323.1775	5.33394	1000	1000	0	12.82	0.0	0.0
6	306.4425	4.45518	1000	1000	0	10.04	0.0	0.0
7	212.3825	2.93639	1000	1	0	11.20	0.0	100.0
8	208.0825	3.38345	1000	1	0	12.13	0.0	100.0
9	210.2500	3.26846	1000	3	0	13.55	0.0	100.0
10	329.6575	6.31161	1000	1000	0	14.04	0.0	0.0
11	314.4650	11.59273	1000	1000	0	10.23	0.0	0.0

Table C.66: PBT_CG_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	539.2900	12.73800	1000	1000	0	9.85	0.0	0.0
2	357.7800	1.69700	1000	1000	0	10.62	0.0	0.0
3	242.8300	3.70889	1000	2	0	10.86	0.0	0.0
4	346.6400	7.66847	1000	1000	0	15.47	0.0	0.0
5	326.0350	6.64453	1000	1000	0	12.82	0.0	0.0
6	308.0225	4.22224	1000	1000	0	10.04	0.0	0.0
7	208.9225	2.43912	1000	1	0	11.20	0.0	100.0
8	206.5200	2.13008	1000	1	0	12.13	0.0	100.0
9	211.0250	6.03941	1000	3	0	13.55	0.0	100.0
10	322.6800	5.58441	1000	1000	0	14.04	0.0	0.0
11	324.2650	12.91835	1000	1000	0	10.23	0.0	0.0

Table C.67: PBT_CG_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	536.1500	8.44678	1000	1000	0	9.85	0.0	0.0
2	358.0575	7.55566	1000	1000	0	10.62	0.0	0.0
3	249.7725	11.79207	1000	2	0	10.86	0.0	0.0
4	351.5250	2.80885	1000	1000	0	15.47	0.0	0.0
5	325.7750	13.06090	1000	1000	0	12.82	0.0	0.0
6	326.6350	5.93047	1000	1000	0	10.04	0.0	0.0
7	213.5800	3.67922	1000	1	0	11.20	0.0	100.0
8	210.2625	4.26161	1000	1	0	12.13	0.0	100.0
9	216.4025	6.04495	1000	3	0	13.55	0.0	100.0
10	329.4775	7.70152	1000	1000	0	14.04	0.0	0.0
11	330.6075	13.09440	1000	1000	0	10.23	0.0	0.0

Table C.68: PBT_CG_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	533.2825	3.25420	1000	1000	0	9.85	0.0	0.0
2	358.2400	6.32251	1000	1000	0	10.62	0.0	0.0
3	242.9050	3.44451	1000	16	5	11.75	0.0	0.0
4	347.4475	2.38753	1000	1000	0	15.47	0.0	0.0
5	317.5350	2.40850	1000	1000	0	12.82	0.0	0.0
6	304.8650	1.87849	1000	1000	0	10.04	0.0	0.0
7	207.8275	1.24162	1000	1	0	11.20	0.0	100.0
8	208.1975	7.22908	1000	1	0	12.13	0.0	100.0
9	211.2250	2.65300	1000	3	0	13.55	0.0	100.0
10	328.3800	6.12581	1000	1000	0	14.04	0.0	0.0
11	325.8450	11.66216	1000	1000	0	10.23	0.0	0.0

Table C.69: PBT_CG_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	553.5450	17.50383	1000	1000	0	9.85	0.0	0.0
2	359.9250	4.12540	1000	1000	0	10.62	0.0	0.0
3	245.6800	3.71064	1000	2	0	10.86	0.0	0.0
4	355.2200	3.40389	1000	1000	0	15.47	0.0	0.0
5	319.3175	4.58219	1000	1000	0	12.82	0.0	0.0
6	303.8200	0.92136	1000	1000	0	10.04	0.0	0.0
7	209.9150	2.40300	1000	1	0	11.20	0.0	100.0
8	206.5725	1.10513	1000	1	0	12.13	0.0	100.0
9	212.7250	2.03682	1000	3	0	13.55	0.0	100.0
10	334.2125	10.41412	1000	1000	0	14.04	0.0	0.0
11	339.0800	6.76664	1000	1000	0	10.23	0.0	0.0

Table C.70: PBT_CG_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	539.9700	3.53466	1000	1000	0	9.85	0.0	0.0
2	361.5600	2.02209	1000	1000	0	10.62	0.0	0.0
3	246.7900	4.98460	1000	2	0	10.86	0.0	0.0
4	349.9650	4.29475	1000	1000	0	15.47	0.0	0.0
5	316.4050	5.14487	1000	1000	0	12.82	0.0	0.0
6	309.9525	1.34513	1000	1000	0	10.04	0.0	0.0
7	212.3400	5.44906	1000	1	0	11.20	0.0	100.0
8	206.6225	2.60240	1000	1	0	12.13	0.0	100.0
9	237.7050	47.16166	1000	3	0	13.55	0.0	100.0
10	337.6475	11.14390	1000	1000	0	14.04	0.0	0.0
11	329.0175	14.63804	1000	1000	0	10.23	0.0	0.0

Table C.71: PBT_CG_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	546.0375	5.25308	1000	1000	0	9.85	0.0	0.0
2	357.7925	2.56629	1000	1000	0	10.62	0.0	0.0
3	243.8025	4.01470	1000	2	0	10.86	0.0	0.0
4	352.1775	4.38532	1000	1000	0	15.47	0.0	0.0
5	317.5100	5.03884	1000	1000	0	12.82	0.0	0.0
6	307.1550	4.12122	1000	1000	0	10.04	0.0	0.0
7	209.4125	1.54199	1000	1	0	11.20	0.0	100.0
8	207.1125	1.90630	1000	1	0	12.13	0.0	100.0
9	211.3175	1.56677	1000	3	0	13.55	0.0	100.0
10	337.1150	8.93727	1000	1000	0	14.04	0.0	0.0
11	328.1775	4.67832	1000	1000	0	10.23	0.0	0.0

Table C.72: PBT_CG_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	547.1625	13.62505	1000	1000	0	9.85	0.0	0.0
2	360.8950	9.76614	1000	1000	0	10.62	0.0	0.0
3	254.5650	2.15843	1000	2	0	10.86	0.0	0.0
4	351.7650	2.66880	1000	1000	0	15.47	0.0	0.0
5	315.1175	2.64105	1000	1000	0	12.82	0.0	0.0
6	312.8425	3.78465	1000	1000	0	10.04	0.0	0.0
7	209.4050	1.37424	1000	1	0	11.20	0.0	100.0
8	209.3225	9.33334	1000	1	0	12.13	0.0	100.0
9	210.9400	2.16383	1000	3	0	13.55	0.0	100.0
10	328.9525	3.36695	1000	1000	0	14.04	0.0	0.0
11	342.5600	15.91447	1000	1000	0	10.23	0.0	0.0

C.5 M4

Table C.73: M4 - Search strategies results.

Strategy	Min Mean Time (ms)	Std Min Mean Time (ms)	Max Mean Time (ms)	Std Max Mean Time (ms)	Min Sample Size	Max Sample Size	Min Mean Test Cases	Std Min Mean Test Cases	Max Mean Test Cases	Std Max Mean Test Cases	Min Mean Coverage (%)	Std Min Mean Coverage (%)	Max Mean Coverage (%)	Std Max Mean Coverage (%)	Bug Found
PBT_R_S4	2.55032	0.07350	2.55032	0.07350	1000	1000	1	0	1	0	0.0	0.0	0.0	0.0	True
PBT_R_S1	2.59212	0.06654	2.59212	0.06654	1000	1000	1	0	1	0	0.0	0.0	0.0	0.0	True
PBT_R_S8	3.63552	0.09239	3.63552	0.09239	1000	1000	1	0	1	0	0.0	0.0	0.0	0.0	True
PBT_CG_S4	208.32000	0.88031	208.32000	0.88031	1000	1000	1	0	1	0	11.2	0.0	11.2	0.0	True
PBT_CG_S6	208.22500	2.63784	208.22500	2.63784	1000	1000	1	0	1	0	11.2	0.0	11.2	0.0	True
PBT_CG_S3	210.80500	4.10454	210.80500	4.10454	1000	1000	1	0	1	0	11.2	0.0	11.2	0.0	True
PBT_R_S3	3.98102	0.10879	3.98102	0.10879	1000	1000	1	0	1	0	0.0	0.0	0.0	0.0	True
PBT_R_S5	4.51428	0.90527	4.51428	0.90527	1000	1000	1	0	1	0	0.0	0.0	0.0	0.0	True
PBT_R_S7	3.57327	0.01432	3.57327	0.01432	1000	1000	1	0	1	0	0.0	0.0	0.0	0.0	True
PBT_CG_S5	214.02000	6.68803	214.02000	6.68803	1000	1000	1	0	1	0	11.2	0.0	11.2	0.0	True
PBT_CG_S7	211.24750	3.80642	211.24750	3.80642	1000	1000	1	0	1	0	11.2	0.0	11.2	0.0	True
PBT_CG_S1	209.46750	3.62313	209.46750	3.62313	1000	1000	1	0	1	0	11.2	0.0	11.2	0.0	True
PBT_CG_S8	210.01250	1.75363	210.01250	1.75363	1000	1000	1	0	1	0	11.2	0.0	11.2	0.0	True
PBT_CG_S2	209.21750	3.00351	209.21750	3.00351	1000	1000	1	0	1	0	11.2	0.0	11.2	0.0	True
PBT_R_S6	3.15160	0.66294	3.15160	0.66294	1000	1000	1	0	1	0	0.0	0.0	0.0	0.0	True
PBT_R_S2	2.71752	0.07605	2.71752	0.07605	1000	1000	1	0	1	0	0.0	0.0	0.0	0.0	True

Table C.74: M4 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-0.3305	0.5607	-1.6174	0.9563	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.2236	0.7045	-1.0861	1.5332	False
Random withConstantExtraction	Random withoutConstantExtraction	-1.1731	0.0037	-1.7990	-0.5473	True
Coverage-Guided withBias	Coverage-Guided withoutBias	-0.4087	0.7950	-3.9181	3.1006	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	1.4262	0.3219	-1.8068	4.6593	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-2.7138	0.0263	-4.9795	-0.4480	True
Coverage-Guided	Random	-206.8249	0.0010	-208.3565	-205.2934	True

C.5.1 Search Strategies

Table C.75: PBT_R_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	8.66278	0.37760	1000	1000	0	0.0	0.0	0.0
2	13.68950	0.31267	1000	1000	0	0.0	0.0	0.0
3	6.16315	0.11325	1000	34	22	0.0	0.0	0.0
4	14.89750	0.12892	1000	1000	0	0.0	0.0	0.0
5	9.68470	0.39289	1000	1000	0	0.0	0.0	0.0
6	8.85952	0.41718	1000	1000	0	0.0	0.0	0.0
7	2.59212	0.06654	1000	1	0	0.0	0.0	100.0
8	2.75478	0.15025	1000	1000	0	0.0	0.0	0.0
9	5.39527	0.32433	1000	1000	0	0.0	0.0	0.0
10	8.10502	0.25545	1000	1000	0	0.0	0.0	0.0
11	10.33160	0.30153	1000	1000	0	0.0	0.0	0.0

Table C.76: PBT_R_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	8.44015	0.13829	1000	1000	0	0.0	0.0	0.0
2	14.31850	0.39005	1000	1000	0	0.0	0.0	0.0
3	6.19645	0.12563	1000	2	0	0.0	0.0	0.0
4	15.50200	0.58125	1000	1000	0	0.0	0.0	0.0
5	9.42033	0.13537	1000	1000	0	0.0	0.0	0.0
6	8.87050	0.22964	1000	1000	0	0.0	0.0	0.0
7	2.71752	0.07605	1000	1	0	0.0	0.0	100.0
8	2.81398	0.18603	1000	1000	0	0.0	0.0	0.0
9	5.34315	0.15411	1000	1000	0	0.0	0.0	0.0
10	9.16765	0.54117	1000	1000	0	0.0	0.0	0.0
11	11.15500	0.80240	1000	1000	0	0.0	0.0	0.0

Table C.77: PBT_R_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14.92125	0.17258	1000	1000	0	0.0	0.0	0.0
2	18.59425	1.28120	1000	1000	0	0.0	0.0	0.0
3	8.26585	0.65788	1000	2	0	0.0	0.0	0.0
4	16.79300	0.42370	1000	1000	0	0.0	0.0	0.0
5	11.42675	0.38144	1000	1000	0	0.0	0.0	0.0
6	9.92732	0.25030	1000	1000	0	0.0	0.0	0.0
7	3.98102	0.10879	1000	1	0	0.0	0.0	100.0
8	4.33638	0.23626	1000	1000	0	0.0	0.0	0.0
9	7.54242	0.21671	1000	1000	0	0.0	0.0	0.0
10	8.28862	0.36326	1000	1000	0	0.0	0.0	0.0
11	12.13625	0.33242	1000	1000	0	0.0	0.0	0.0

Table C.78: PBT_R_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	8.29547	0.08742	1000	1000	0	0.0	0.0	0.0
2	13.92650	0.15969	1000	1000	0	0.0	0.0	0.0
3	6.32372	0.35685	1000	19	27	0.0	0.0	0.0
4	15.08175	0.44477	1000	1000	0	0.0	0.0	0.0
5	9.25550	0.20168	1000	1000	0	0.0	0.0	0.0
6	8.22060	0.13187	1000	1000	0	0.0	0.0	0.0
7	2.55032	0.07350	1000	1	0	0.0	0.0	100.0
8	2.79330	0.22088	1000	1000	0	0.0	0.0	0.0
9	5.15680	0.07114	1000	1000	0	0.0	0.0	0.0
10	8.78602	0.50386	1000	1000	0	0.0	0.0	0.0
11	10.14172	0.16143	1000	1000	0	0.0	0.0	0.0

Table C.79: PBT_R_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14.62400	0.34970	1000	1000	0	0.0	0.0	0.0
2	18.35950	0.51209	1000	1000	0	0.0	0.0	0.0
3	8.16710	0.38489	1000	2	0	0.0	0.0	0.0
4	16.79850	0.37303	1000	1000	0	0.0	0.0	0.0
5	11.64925	0.33192	1000	1000	0	0.0	0.0	0.0
6	9.88157	0.45888	1000	1000	0	0.0	0.0	0.0
7	4.51428	0.90527	1000	1	0	0.0	0.0	100.0
8	4.48043	0.21505	1000	1000	0	0.0	0.0	0.0
9	7.76145	0.28047	1000	1000	0	0.0	0.0	0.0
10	8.38523	0.35803	1000	1000	0	0.0	0.0	0.0
11	12.08525	0.22040	1000	1000	0	0.0	0.0	0.0

Table C.80: PBT_R_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	8.50072	0.18489	1000	1000	0	0.0	0.0	0.0
2	14.78925	0.79184	1000	1000	0	0.0	0.0	0.0
3	6.27593	0.16751	1000	2	0	0.0	0.0	0.0
4	15.44200	0.35161	1000	1000	0	0.0	0.0	0.0
5	9.66682	0.40176	1000	1000	0	0.0	0.0	0.0
6	9.55315	1.10734	1000	1000	0	0.0	0.0	0.0
7	3.15160	0.66294	1000	1	0	0.0	0.0	100.0
8	2.74693	0.17995	1000	1000	0	0.0	0.0	0.0
9	5.24927	0.10120	1000	1000	0	0.0	0.0	0.0
10	8.82338	0.97885	1000	1000	0	0.0	0.0	0.0
11	10.34225	0.27559	1000	1000	0	0.0	0.0	0.0

Table C.81: PBT_R_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14.52300	0.40618	1000	1000	0	0.0	0.0	0.0
2	17.33400	0.23277	1000	1000	0	0.0	0.0	0.0
3	7.66308	0.11166	1000	2	0	0.0	0.0	0.0
4	16.42100	0.22267	1000	1000	0	0.0	0.0	0.0
5	11.56475	0.69838	1000	1000	0	0.0	0.0	0.0
6	9.98702	0.53916	1000	1000	0	0.0	0.0	0.0
7	3.57327	0.01432	1000	1	0	0.0	0.0	100.0
8	3.56163	0.02964	1000	1000	0	0.0	0.0	0.0
9	6.66210	0.33994	1000	1000	0	0.0	0.0	0.0
10	7.81922	0.67605	1000	1000	0	0.0	0.0	0.0
11	11.94925	0.37812	1000	1000	0	0.0	0.0	0.0

Table C.82: PBT_R_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	15.03275	0.87066	1000	1000	0	0.0	0.0	0.0
2	18.07825	0.83290	1000	1000	0	0.0	0.0	0.0
3	8.04337	0.35991	1000	2	0	0.0	0.0	0.0
4	16.97475	0.69520	1000	1000	0	0.0	0.0	0.0
5	11.43575	0.43548	1000	1000	0	0.0	0.0	0.0
6	10.08778	0.46409	1000	1000	0	0.0	0.0	0.0
7	3.63552	0.09239	1000	1	0	0.0	0.0	100.0
8	3.92975	0.29457	1000	1000	0	0.0	0.0	0.0
9	7.20710	1.24695	1000	1000	0	0.0	0.0	0.0
10	7.72917	0.37698	1000	1000	0	0.0	0.0	0.0
11	11.77000	0.10306	1000	1000	0	0.0	0.0	0.0

Table C.83: PBT_CG_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	534.0825	4.89303	1000	1000	0	9.85	0.0	0.0
2	355.7625	7.34572	1000	1000	0	10.62	0.0	0.0
3	251.7900	20.12112	1000	10	2	11.75	0.0	0.0
4	345.4075	6.55353	1000	1000	0	15.47	0.0	0.0
5	318.2975	4.18855	1000	1000	0	12.82	0.0	0.0
6	306.6250	3.41358	1000	1000	0	10.04	0.0	0.0
7	209.4675	3.62313	1000	1	0	11.20	0.0	100.0
8	282.8850	3.26232	1000	1000	0	12.13	0.0	0.0
9	302.2525	11.75109	1000	1000	0	13.55	0.0	0.0
10	322.0225	7.18275	1000	1000	0	14.04	0.0	0.0
11	350.6200	23.37844	1000	1000	0	10.23	0.0	0.0

Table C.84: PBT_CG_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	533.4775	8.72333	1000	1000	0	9.85	0.0	0.0
2	360.7850	10.95365	1000	1000	0	10.62	0.0	0.0
3	243.2275	5.43371	1000	2	0	10.86	0.0	0.0
4	359.1900	8.68448	1000	1000	0	15.47	0.0	0.0
5	324.5900	4.51462	1000	1000	0	12.82	0.0	0.0
6	309.3300	8.64693	1000	1000	0	10.04	0.0	0.0
7	209.2175	3.00351	1000	1	0	11.20	0.0	100.0
8	280.9675	2.16264	1000	1000	0	12.13	0.0	0.0
9	296.7650	4.47333	1000	1000	0	13.55	0.0	0.0
10	326.2925	4.57423	1000	1000	0	14.04	0.0	0.0
11	352.5650	26.49323	1000	1000	0	10.23	0.0	0.0

Table C.85: PBT_CG_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	538.9950	6.49593	1000	1000	0	9.85	0.0	0.0
2	359.4625	5.92180	1000	1000	0	10.62	0.0	0.0
3	245.6225	5.23826	1000	2	0	10.86	0.0	0.0
4	359.9975	10.67825	1000	1000	0	15.47	0.0	0.0
5	332.2250	7.25230	1000	1000	0	12.82	0.0	0.0
6	316.3800	9.69931	1000	1000	0	10.04	0.0	0.0
7	210.8050	4.10454	1000	1	0	11.20	0.0	100.0
8	284.7275	2.39441	1000	1000	0	12.13	0.0	0.0
9	300.5575	8.70649	1000	1000	0	13.55	0.0	0.0
10	329.8425	5.44193	1000	1000	0	14.04	0.0	0.0
11	346.3050	4.16417	1000	1000	0	10.23	0.0	0.0

Table C.86: PBT_CG_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	536.7425	5.32293	1000	1000	0	9.85	0.0	0.0
2	360.3900	4.11041	1000	1000	0	10.62	0.0	0.0
3	244.8600	3.72183	1000	23	7	11.75	0.0	0.0
4	350.5275	8.06327	1000	1000	0	15.47	0.0	0.0
5	318.7900	1.54785	1000	1000	0	12.82	0.0	0.0
6	309.5675	2.83849	1000	1000	0	10.04	0.0	0.0
7	208.3200	0.88031	1000	1	0	11.20	0.0	100.0
8	284.4175	1.47089	1000	1000	0	12.13	0.0	0.0
9	297.2025	3.86150	1000	1000	0	13.55	0.0	0.0
10	321.8850	6.33617	1000	1000	0	14.04	0.0	0.0
11	392.5300	10.35823	1000	1000	0	10.23	0.0	0.0

Table C.87: PBT_CG_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	542.4300	6.37806	1000	1000	0	9.85	0.0	0.0
2	363.0200	3.83417	1000	1000	0	10.62	0.0	0.0
3	247.7400	2.40855	1000	2	0	10.86	0.0	0.0
4	357.2350	7.41567	1000	1000	0	15.47	0.0	0.0
5	319.6100	4.94114	1000	1000	0	12.82	0.0	0.0
6	310.1475	5.30375	1000	1000	0	10.04	0.0	0.0
7	214.0200	6.68803	1000	1	0	11.20	0.0	100.0
8	291.4625	6.19183	1000	1000	0	12.13	0.0	0.0
9	302.0925	8.49386	1000	1000	0	13.55	0.0	0.0
10	327.1775	9.43414	1000	1000	0	14.04	0.0	0.0
11	347.9550	25.70385	1000	1000	0	10.23	0.0	0.0

Table C.88: PBT_CG_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	532.4400	5.11076	1000	1000	0	9.85	0.0	0.0
2	351.6900	2.98517	1000	1000	0	10.62	0.0	0.0
3	242.3825	4.21521	1000	2	0	10.86	0.0	0.0
4	352.8075	6.24110	1000	1000	0	15.47	0.0	0.0
5	316.1500	2.55697	1000	1000	0	12.82	0.0	0.0
6	313.1200	8.10056	1000	1000	0	10.04	0.0	0.0
7	208.2250	2.63784	1000	1	0	11.20	0.0	100.0
8	286.0150	3.03185	1000	1000	0	12.13	0.0	0.0
9	297.5000	2.98700	1000	1000	0	13.55	0.0	0.0
10	329.2300	9.30912	1000	1000	0	14.04	0.0	0.0
11	365.6200	28.77823	1000	1000	0	10.23	0.0	0.0

Table C.89: PBT_CG_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	538.8900	3.01529	1000	1000	0	9.85	0.0	0.0
2	355.7375	3.18515	1000	1000	0	10.62	0.0	0.0
3	242.8425	2.75882	1000	2	0	10.86	0.0	0.0
4	350.5925	4.16355	1000	1000	0	15.47	0.0	0.0
5	321.9275	4.18067	1000	1000	0	12.82	0.0	0.0
6	313.6575	8.95266	1000	1000	0	10.04	0.0	0.0
7	211.2475	3.80642	1000	1	0	11.20	0.0	100.0
8	287.5475	8.44990	1000	1000	0	12.13	0.0	0.0
9	295.3375	1.40869	1000	1000	0	13.55	0.0	0.0
10	329.3050	9.71671	1000	1000	0	14.04	0.0	0.0
11	354.5425	19.26567	1000	1000	0	10.23	0.0	0.0

Table C.90: PBT_CG_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	543.6800	16.27494	1000	1000	0	9.85	0.0	0.0
2	358.3750	4.31400	1000	1000	0	10.62	0.0	0.0
3	244.0375	3.73268	1000	2	0	10.86	0.0	0.0
4	358.8275	7.25581	1000	1000	0	15.47	0.0	0.0
5	321.3075	3.87742	1000	1000	0	12.82	0.0	0.0
6	310.2950	3.35066	1000	1000	0	10.04	0.0	0.0
7	210.0125	1.75363	1000	1	0	11.20	0.0	100.0
8	282.9850	1.85775	1000	1000	0	12.13	0.0	0.0
9	297.5375	4.37411	1000	1000	0	13.55	0.0	0.0
10	326.7700	7.24007	1000	1000	0	14.04	0.0	0.0
11	364.4775	27.98056	1000	1000	0	10.23	0.0	0.0

Appendix D

Multiplication Sign Results

This appendix presents all the data associated with the results achieved for all the `Multiplication Sign` benchmark variants, including each search strategy's results and all the data extracted concerning all the evaluation metrics.

D.1 M0

Table D.1: M0 - Search strategies results.

Strategy	Min Mean Time (ms)	Std Min Mean Time (ms)	Max Mean Time (ms)	Std Max Mean Time (ms)	Min Sample Size	Max Sample Size	Min Mean Test Cases	Std Min Mean Test Cases	Max Mean Test Cases	Std Max Mean Test Cases	Min Mean Coverage (%)	Std Min Mean Coverage (%)	Max Mean Coverage (%)	Std Max Mean Coverage (%)	Bug Found
PBT_R_S4	1478.1250	18.17545	1506.6250	28.33464	1000	1000	1	0	1	0	0.0	0.0	0.00	0.0	True
PBT_R_S1	475.4775	24.98263	512.1000	25.23949	100	100	1	0	1	0	0.0	0.0	0.00	0.0	True
PBT_R_S8	1520.0500	36.94249	1525.8750	42.02216	1000	1000	1	0	1	0	0.0	0.0	0.00	0.0	True
PBT_CG_S4	238.5875	25.37159	266.0425	18.07512	1000	1000	1	0	1	0	58.4	0.0	89.07	0.0	True
PBT_CG_S6	238.2550	18.51653	248.2525	9.16629	1000	1000	1	0	1	0	58.4	0.0	89.07	0.0	True
PBT_CG_S3	138.3375	2.87879	150.9300	4.52975	100	100	1	0	1	0	58.4	0.0	89.07	0.0	True
PBT_R_S3	507.7475	11.08134	521.0625	6.09098	100	100	1	0	1	0	0.0	0.0	0.00	0.0	True
PBT_R_S5	496.8200	4.32654	516.0900	15.13285	100	100	1	0	1	0	0.0	0.0	0.00	0.0	True
PBT_R_S7	1559.4000	48.39215	1577.0250	67.25371	1000	1000	1	0	1	0	0.0	0.0	0.00	0.0	True
PBT_CG_S5	148.2450	8.18261	152.8275	6.14311	100	100	1	0	1	0	58.4	0.0	89.07	0.0	True
PBT_CG_S7	237.4450	8.25174	238.2400	18.79843	1000	1000	1	0	1	0	58.4	0.0	89.07	0.0	True
PBT_CG_S1	141.7775	18.28360	149.8825	16.34237	100	100	1	0	1	0	58.4	0.0	89.07	0.0	True
PBT_CG_S8	246.4725	19.06471	248.8575	21.81537	1000	1000	1	0	1	0	58.4	0.0	89.07	0.0	True
PBT_CG_S2	149.3800	11.09162	159.9125	11.51231	100	100	1	0	1	0	58.4	0.0	89.07	0.0	True
PBT_R_S6	1498.9000	52.18252	1536.3000	42.08818	1000	1000	1	0	1	0	0.0	0.0	0.00	0.0	True
PBT_R_S2	490.4450	23.93745	538.5850	13.14264	100	100	1	0	1	0	0.0	0.0	0.00	0.0	True

Table D.2: M0 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	3.6337	0.9000	-1017.7970	1025.0645	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-1021.4963	0.0010	-1067.0433	-975.9492	True
Random withConstantExtraction	Random withoutConstantExtraction	-35.2675	0.9000	-1056.0970	985.5620	False
Coverage-Guided withBias	Coverage-Guided withoutBias	-6.5512	0.8753	-102.3376	89.2351	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-95.7550	0.0010	-104.0085	-87.5015	True
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-0.6250	0.9000	-96.6326	95.3826	False
Coverage-Guided	Random	811.0581	0.0010	394.7795	1227.3368	True

D.1.1 Search Strategies

Table D.3: PBT_R_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1113.10000	58.52807	100	1000000	0	0.0	0.0	0.0
2	1240.30000	117.07237	100	1000000	0	0.0	0.0	0.0
3	475.47750	24.98263	100	1	0	0.0	0.0	100.0
4	512.10000	25.23949	100	1	0	0.0	0.0	100.0
5	4.21685	2.16346	1000	1000	0	0.0	0.0	0.0

Table D.4: PBT_R_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1113.6000	55.00345	100	1000000	0	0.0	0.0	0.0
2	1214.0250	73.93465	100	1000000	0	0.0	0.0	0.0
3	490.4450	23.93745	100	1	0	0.0	0.0	100.0
4	538.5850	13.14264	100	1	0	0.0	0.0	100.0
5	7.1994	2.36215	1000	1000	0	0.0	0.0	0.0

Table D.5: PBT_R_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1141.27500	8.99177	100	1000000	0	0.0	0.0	0.0
2	1322.10000	77.77799	100	1000000	0	0.0	0.0	0.0
3	507.74750	11.08134	100	1	0	0.0	0.0	100.0
4	521.06250	6.09098	100	1	0	0.0	0.0	100.0
5	5.57493	2.12213	1000	1000	0	0.0	0.0	0.0

Table D.6: PBT_R_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2376.65000	104.50001	1000	1000000	0	0.0	0.0	0.0
2	2230.52500	49.32268	1000	1000000	0	0.0	0.0	0.0
3	1506.62500	28.33464	1000	1	0	0.0	0.0	100.0
4	1478.12500	18.17545	1000	1	0	0.0	0.0	100.0
5	9.30133	0.07398	1000	1000	0	0.0	0.0	0.0

Table D.7: PBT_R_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1116.37500	6.09359	100	1000000	0	0.0	0.0	0.0
2	1200.47500	54.09752	100	1000000	0	0.0	0.0	0.0
3	496.82000	4.32654	100	1	0	0.0	0.0	100.0
4	516.09000	15.13285	100	1	0	0.0	0.0	100.0
5	7.88422	2.15593	1000	1000	0	0.0	0.0	0.0

Table D.8: PBT_R_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2282.8750	10.60103	1000	1000000	0	0.0	0.0	0.0
2	2229.8750	18.02129	1000	1000000	0	0.0	0.0	0.0
3	1498.9000	52.18252	1000	1	0	0.0	0.0	100.0
4	1536.3000	42.08818	1000	1	0	0.0	0.0	100.0
5	9.4329	0.30778	1000	1000	0	0.0	0.0	0.0

Table D.9: PBT_R_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2269.1750	47.02948	1000	1000000	0	0.0	0.0	0.0
2	2285.5500	81.01819	1000	1000000	0	0.0	0.0	0.0
3	1577.0250	67.25371	1000	1	0	0.0	0.0	100.0
4	1559.4000	48.39215	1000	1	0	0.0	0.0	100.0
5	11.0005	0.85985	1000	1000	0	0.0	0.0	0.0

Table D.10: PBT_R_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2294.500	5.84679	1000	1000000	0	0.0	0.0	0.0
2	2225.975	23.89209	1000	1000000	0	0.0	0.0	0.0
3	1520.050	36.94249	1000	1	0	0.0	0.0	100.0
4	1525.875	42.02216	1000	1	0	0.0	0.0	100.0
5	10.836	0.73866	1000	1000	0	0.0	0.0	0.0

Table D.11: PBT_CG_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	13914.0000	120.25182	100	1000000	0	55.36	0.0	0.0
2	14825.5000	230.72549	100	1000000	0	75.89	0.0	0.0
3	149.8825	16.34237	100	1	0	89.07	0.0	100.0
4	141.7775	18.28360	100	1	0	58.40	0.0	100.0
5	162.1850	14.24130	1000	1000	0	89.28	0.0	0.0

Table D.12: PBT_CG_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	13979.0000	25.07987	100	1000000	0	55.36	0.0	0.0
2	14758.2500	224.61342	100	1000000	0	75.89	0.0	0.0
3	159.9125	11.51231	100	1	0	89.07	0.0	100.0
4	149.3800	11.09162	100	1	0	58.40	0.0	100.0
5	166.6350	11.25526	1000	1000	0	89.28	0.0	0.0

Table D.13: PBT_CG_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	13981.7500	79.99805	100	1000000	0	55.36	0.0	0.0
2	14829.5000	121.24871	100	1000000	0	75.89	0.0	0.0
3	150.9300	4.52975	100	1	0	89.07	0.0	100.0
4	138.3375	2.87879	100	1	0	58.40	0.0	100.0
5	158.3300	3.12195	1000	1000	0	89.28	0.0	0.0

Table D.14: PBT_CG_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	15198.2500	52.12665	1000	1000000	0	55.36	0.0	0.0
2	15739.2500	200.32146	1000	1000000	0	75.89	0.0	0.0
3	266.0425	18.07512	1000	1	0	89.07	0.0	100.0
4	238.5875	25.37159	1000	1	0	58.40	0.0	100.0
5	163.1400	4.21729	1000	1000	0	89.28	0.0	0.0

Table D.15: PBT_CG_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	13895.7500	115.50839	100	1000000	0	55.36	0.0	0.0
2	14802.5000	73.83258	100	1000000	0	75.89	0.0	0.0
3	152.8275	6.14311	100	1	0	89.07	0.0	100.0
4	148.2450	8.18261	100	1	0	58.40	0.0	100.0
5	165.6200	7.65516	1000	1000	0	89.28	0.0	0.0

Table D.16: PBT_CG_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	15070.5000	101.74601	1000	1000000	0	55.36	0.0	0.0
2	15808.0000	62.49800	1000	1000000	0	75.89	0.0	0.0
3	248.2525	9.16629	1000	1	0	89.07	0.0	100.0
4	238.2550	18.51653	1000	1	0	58.40	0.0	100.0
5	179.9600	21.59943	1000	1000	0	89.28	0.0	0.0

Table D.17: PBT_CG_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	15037.000	18.62794	1000	1000000	0	55.36	0.0	0.0
2	15788.250	174.92481	1000	1000000	0	75.89	0.0	0.0
3	237.445	8.25174	1000	1	0	89.07	0.0	100.0
4	238.240	18.79843	1000	1	0	58.40	0.0	100.0
5	185.395	17.78008	1000	1000	0	89.28	0.0	0.0

Table D.18: PBT_CG_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	15256.5000	135.69175	1000	1000000	0	55.36	0.0	0.0
2	15834.0000	211.10306	1000	1000000	0	75.89	0.0	0.0
3	246.4725	19.06471	1000	1	0	89.07	0.0	100.0
4	248.8575	21.81537	1000	1	0	58.40	0.0	100.0
5	193.7325	19.71538	1000	1000	0	89.28	0.0	0.0

D.2 M1

Table D.19: M1 - Search strategies results.

Strategy	Min Mean Time (ms)	Std Min Mean Time (ms)	Max Mean Time (ms)	Std Max Mean Time (ms)	Min Sample Size	Max Sample Size	Min Mean Test Cases	Std Min Mean Test Cases	Max Mean Test Cases	Std Max Mean Test Cases	Min Mean Coverage (%)	Std Min Mean Coverage (%)	Max Mean Coverage (%)	Std Max Mean Coverage (%)	Bug Found
PBT_R_S4	4.03792	0.21375	1551.0000	63.78773	1000	1000	1	0	1	0	0.00	0.0	0.00	0.00	True
PBT_R_S1	0.47548	0.01595	553.8400	14.39956	100	1000	1	0	1	0	0.00	0.0	0.00	0.00	True
PBT_R_S8	8.62335	0.94586	1584.9000	57.80523	1000	1000	1	0	2	0	0.00	0.0	0.00	0.00	True
PBT_CG_S4	129.98500	6.55303	236.7000	6.34278	1000	1000	1	0	1	0	56.93	0.0	85.36	1.79	True
PBT_CG_S6	140.96500	9.83831	231.7600	8.17458	1000	1000	1	0	2	0	56.93	0.0	88.57	0.00	True
PBT_CG_S3	132.84750	12.93344	161.7400	6.86098	100	1000	1	0	2	0	56.93	0.0	85.71	0.00	True
PBT_R_S3	4.40460	2.67685	525.7025	18.74597	100	1000	1	0	2	0	0.00	0.0	0.00	0.00	True
PBT_R_S5	4.13638	2.42365	509.2125	19.83557	100	1000	1	0	2	0	0.00	0.0	0.00	0.00	True
PBT_R_S7	9.46177	0.86619	1551.5500	57.99623	1000	1000	1	0	2	0	0.00	0.0	0.00	0.00	True
PBT_CG_S5	131.40500	14.93779	163.3400	1.40849	100	1000	1	0	2	0	56.93	0.0	85.71	0.00	True
PBT_CG_S7	134.07750	6.12459	239.9525	8.23805	1000	1000	1	0	2	0	56.93	0.0	85.71	0.00	True
PBT_CG_S1	138.80000	9.93340	158.1375	11.17639	100	1000	1	0	1	0	56.93	0.0	86.25	1.55	True
PBT_CG_S8	136.76000	13.58991	245.4725	11.39250	1000	1000	1	0	2	0	56.93	0.0	85.71	0.00	True
PBT_CG_S2	141.00250	15.29668	153.5100	6.57994	100	1000	1	0	2	0	56.93	0.0	88.57	0.00	True
PBT_R_S6	4.05563	0.15840	1585.3250	47.22856	1000	1000	1	0	2	0	0.00	0.0	0.00	0.00	True
PBT_R_S2	0.49041	0.00695	549.0975	26.74074	100	1000	1	0	2	0	0.00	0.0	0.00	0.00	True

Table D.20: M1 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.2685	0.9000	-5.8198	6.3568	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-4.1679	0.0618	-8.6181	0.2822	False
Random withConstantExtraction	Random withoutConstantExtraction	-4.3917	0.0441	-8.6217	-0.1617	True
Coverage-Guided withBias	Coverage-Guided withoutBias	-3.6056	0.2633	-10.7549	3.5436	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	0.5669	0.8709	-7.4184	8.5521	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	3.9156	0.2192	-3.0690	10.9003	False
Coverage-Guided	Random	-131.2696	0.0010	-135.3519	-127.1873	True

D.2.1 Search Strategies

Table D.21: PBT_R_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1151.12500	15.76696	100	1000000	0	0.0	0.0	0.0
2	515.20750	18.74871	100	1	0	0.0	0.0	100.0
3	1357.65000	13.61185	100	1000000	0	0.0	0.0	0.0
4	553.84000	14.39956	100	1	0	0.0	0.0	100.0
5	0.47548	0.01595	1000	1	0	0.0	0.0	100.0

Table D.22: PBT_R_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1143.87500	13.48172	100	1000000	0	0.0	0.0	0.0
2	508.58000	18.99488	100	1	0	0.0	0.0	100.0
3	1355.37500	17.19671	100	1000000	0	0.0	0.0	0.0
4	549.09750	26.74074	100	1	0	0.0	0.0	100.0
5	0.49041	0.00695	1000	2	0	0.0	0.0	100.0

Table D.23: PBT_R_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1148.8500	10.63167	100	1000000	0	0.0	0.0	0.0
2	506.3575	15.82732	100	1	0	0.0	0.0	100.0
3	1388.8000	19.62307	100	1000000	0	0.0	0.0	0.0
4	525.7025	18.74597	100	1	0	0.0	0.0	100.0
5	4.4046	2.67685	1000	2	0	0.0	0.0	100.0

Table D.24: PBT_R_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2299.72500	51.26214	1000	1000000	0	0.0	0.0	0.0
2	1551.00000	63.78773	1000	1	0	0.0	0.0	100.0
3	2234.52500	68.51209	1000	1000000	0	0.0	0.0	0.0
4	1522.75000	33.19650	1000	1	0	0.0	0.0	100.0
5	4.03792	0.21375	1000	1	0	0.0	0.0	100.0

Table D.25: PBT_R_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1126.45000	8.69152	100	1000000	0	0.0	0.0	0.0
2	494.43500	24.39982	100	1	0	0.0	0.0	100.0
3	1343.32500	9.06818	100	1000000	0	0.0	0.0	0.0
4	509.21250	19.83557	100	1	0	0.0	0.0	100.0
5	4.13638	2.42365	1000	2	0	0.0	0.0	100.0

Table D.26: PBT_R_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2282.90000	32.28103	1000	1000000	0	0.0	0.0	0.0
2	1547.65000	52.61262	1000	1	0	0.0	0.0	100.0
3	2219.17500	60.36072	1000	1000000	0	0.0	0.0	0.0
4	1585.32500	47.22856	1000	1	0	0.0	0.0	100.0
5	4.05563	0.15840	1000	2	0	0.0	0.0	100.0

Table D.27: PBT_R_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2273.42500	43.74868	1000	1000000	0	0.0	0.0	0.0
2	1551.55000	57.99623	1000	1	0	0.0	0.0	100.0
3	2300.35000	115.06330	1000	1000000	0	0.0	0.0	0.0
4	1528.37500	63.92036	1000	1	0	0.0	0.0	100.0
5	9.46177	0.86619	1000	2	0	0.0	0.0	100.0

Table D.28: PBT_R_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2280.42500	9.96303	1000	1000000	0	0.0	0.0	0.0
2	1508.30000	9.37203	1000	1	0	0.0	0.0	100.0
3	2224.92500	23.19271	1000	1000000	0	0.0	0.0	0.0
4	1584.90000	57.80523	1000	1	0	0.0	0.0	100.0
5	8.62335	0.94586	1000	2	0	0.0	0.0	100.0

Table D.29: PBT_CG_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	13880.0000	71.66240	100	1000000	0	55.36	0.00	0.0
2	158.1375	11.17639	100	1	0	58.93	0.00	100.0
3	15681.5000	131.20690	100	1000000	0	64.28	0.00	0.0
4	152.9625	7.01594	100	1	0	56.93	0.00	100.0
5	138.8000	9.93340	1000	1	0	86.25	1.55	100.0

Table D.30: PBT_CG_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	13940.5000	81.82451	100	1000000	0	55.36	0.0	0.0
2	153.5100	6.57994	100	1	0	58.93	0.0	100.0
3	15830.0000	38.89087	100	1000000	0	64.28	0.0	0.0
4	150.4950	10.24297	100	1	0	56.93	0.0	100.0
5	141.0025	15.29668	1000	2	0	88.57	0.0	100.0

Table D.31: PBT_CG_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	13960.5000	156.09372	100	1000000	0	55.36	0.0	0.0
2	161.7400	6.86098	100	1	0	58.93	0.0	100.0
3	15795.2500	93.45152	100	1000000	0	64.28	0.0	0.0
4	147.8900	6.61311	100	1	0	56.93	0.0	100.0
5	132.8475	12.93344	1000	2	0	85.71	0.0	100.0

Table D.32: PBT_CG_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	15283.500	91.37423	1000	1000000	0	55.36	0.00	0.0
2	236.700	6.34278	1000	1	0	58.93	0.00	100.0
3	16880.500	205.48540	1000	1000000	0	64.28	0.00	0.0
4	224.505	4.64469	1000	1	0	56.93	0.00	100.0
5	129.985	6.55303	1000	1	0	85.36	1.79	100.0

Table D.33: PBT_CG_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14028.7500	96.91588	100	1000000	0	55.36	0.0	0.0
2	163.3400	1.40849	100	1	0	58.93	0.0	100.0
3	15832.5000	72.96061	100	1000000	0	64.28	0.0	0.0
4	150.6275	14.97481	100	1	0	56.93	0.0	100.0
5	131.4050	14.93779	1000	2	0	85.71	0.0	100.0

Table D.34: PBT_CG_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	15092.0000	127.13182	1000	1000000	0	55.36	0.0	0.0
2	229.8875	5.08653	1000	1	0	58.93	0.0	100.0
3	16716.7500	101.02568	1000	1000000	0	64.28	0.0	0.0
4	231.7600	8.17458	1000	1	0	56.93	0.0	100.0
5	140.9650	9.83831	1000	2	0	88.57	0.0	100.0

Table D.35: PBT_CG_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	15144.2500	102.56309	1000	1000000	0	55.36	0.0	0.0
2	239.9525	8.23805	1000	1	0	58.93	0.0	100.0
3	16775.2500	208.80778	1000	1000000	0	64.28	0.0	0.0
4	232.2700	8.14836	1000	1	0	56.93	0.0	100.0
5	134.0775	6.12459	1000	2	0	85.71	0.0	100.0

Table D.36: PBT_CG_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	15071.5000	58.79838	1000	1000000	0	55.36	0.0	0.0
2	245.4725	11.39250	1000	1	0	58.93	0.0	100.0
3	16780.0000	104.77357	1000	1000000	0	64.28	0.0	0.0
4	232.5050	8.72914	1000	1	0	56.93	0.0	100.0
5	136.7600	13.58991	1000	2	0	85.71	0.0	100.0

D.3 M2

Table D.37: M2 - Search strategies results.

Strategy	Min Mean Time (ms)	Std Min Mean Time (ms)	Max Mean Time (ms)	Std Max Mean Time (ms)	Min Sample Size	Max Sample Size	Min Mean Test Cases	Std Min Mean Test Cases	Max Mean Test Cases	Std Max Mean Test Cases	Min Mean Coverage (%)	Std Min Mean Coverage (%)	Max Mean Coverage (%)	Std Max Mean Coverage (%)	Bug Found
PBT_R_S4	3.72248	1.38579	1562.5750	55.25004	1000	1000	1	0	2	1	0.00	0.0	0.00	0.00	True
PBT_R_S1	1.68263	2.05258	531.6375	16.48468	100	1000	1	0	3	0	0.00	0.0	0.00	0.00	True
PBT_R_S8	10.10305	2.43468	1576.3750	74.99571	1000	1000	1	0	3	0	0.00	0.0	0.00	0.00	True
PBT_CG_S4	138.87250	15.64130	236.1925	10.49484	1000	1000	1	0	1	0	58.03	0.0	83.39	3.40	True
PBT_CG_S6	134.01500	13.57683	246.7500	19.71166	1000	1000	1	0	2	0	58.03	0.0	85.00	0.00	True
PBT_CG_S3	128.53250	10.48535	166.2550	1.89211	100	1000	1	0	3	0	58.03	0.0	89.28	0.00	True
PBT_R_S3	3.01350	2.01790	521.0450	20.64416	100	1000	1	0	3	0	0.00	0.0	0.00	0.00	True
PBT_R_S5	4.11680	2.45140	499.3775	20.16646	100	1000	1	0	3	0	0.00	0.0	0.00	0.00	True
PBT_R_S7	9.47633	0.89059	1578.5000	73.52622	1000	1000	1	0	3	0	0.00	0.0	0.00	0.00	True
PBT_CG_S5	137.33000	9.93216	168.5900	1.99481	100	1000	1	0	3	0	58.03	0.0	89.28	0.00	True
PBT_CG_S7	146.31500	7.15756	237.1000	8.45294	1000	1000	1	0	3	0	58.03	0.0	89.28	0.00	True
PBT_CG_S1	142.26250	19.01071	154.9050	19.51898	100	1000	1	0	1	0	58.03	0.0	85.36	3.92	True
PBT_CG_S8	131.54750	11.39531	240.4025	7.85808	1000	1000	1	0	3	0	58.03	0.0	89.28	0.00	True
PBT_CG_S2	130.06250	5.96792	152.3300	8.87367	100	1000	1	0	2	0	58.03	0.0	85.00	0.00	True
PBT_R_S6	3.94612	0.09619	1549.3500	87.90636	1000	1000	1	0	2	0	0.00	0.0	0.00	0.00	True
PBT_R_S2	0.65821	0.28328	503.8875	13.35400	100	1000	1	0	2	0	0.00	0.0	0.00	0.00	True

Table D.38: M2 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-0.2323	0.9000	-6.6268	6.1621	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-4.4442	0.0563	-9.0522	0.1638	False
Random withConstantExtraction	Random withoutConstantExtraction	-4.1751	0.0799	-9.0277	0.6776	False
Coverage-Guided withBias	Coverage-Guided withoutBias	5.7569	0.2121	-4.3327	15.8464	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-3.1406	0.5207	-14.3222	8.0410	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	0.3719	0.9000	-11.2356	11.9793	False
Coverage-Guided	Random	-131.5273	0.0010	-136.9074	-126.1472	True

D.3.1 Search Strategies

Table D.39: PBT_R_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1137.80000	5.39768	100	1000000	0	0.0	0.0	0.0
2	506.72500	2.32309	100	1	0	0.0	0.0	100.0
3	1372.60000	20.25451	100	1000000	0	0.0	0.0	0.0
4	531.63750	16.48468	100	1	0	0.0	0.0	100.0
5	1.68263	2.05258	1000	3	0	0.0	0.0	100.0

Table D.40: PBT_R_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1134.67500	6.18036	100	1000000	0	0.0	0.0	0.0
2	494.69500	14.28442	100	1	0	0.0	0.0	100.0
3	1378.80000	17.11797	100	1000000	0	0.0	0.0	0.0
4	503.88750	13.35400	100	1	0	0.0	0.0	100.0
5	0.65821	0.28328	1000	2	0	0.0	0.0	100.0

Table D.41: PBT_R_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1133.2500	21.50285	100	1000000	0	0.0	0.0	0.0
2	495.0050	9.57167	100	1	0	0.0	0.0	100.0
3	1396.0750	21.89216	100	1000000	0	0.0	0.0	0.0
4	521.0450	20.64416	100	1	0	0.0	0.0	100.0
5	3.0135	2.01790	1000	3	0	0.0	0.0	100.0

Table D.42: PBT_R_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2301.52500	16.63842	1000	1000000	0	0.0	0.0	0.0
2	1519.72500	16.12550	1000	1	0	0.0	0.0	100.0
3	2170.85000	39.18957	1000	1000000	0	0.0	0.0	0.0
4	1562.57500	55.25004	1000	1	0	0.0	0.0	100.0
5	3.72248	1.38579	1000	2	1	0.0	0.0	100.0

Table D.43: PBT_R_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1125.9250	9.20391	100	1000000	0	0.0	0.0	0.0
2	499.3775	20.16646	100	1	0	0.0	0.0	100.0
3	1389.4500	27.12881	100	1000000	0	0.0	0.0	0.0
4	495.6800	9.46133	100	1	0	0.0	0.0	100.0
5	4.1168	2.45140	1000	3	0	0.0	0.0	100.0

Table D.44: PBT_R_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2250.90000	37.06872	1000	1000000	0	0.0	0.0	0.0
2	1548.90000	48.54045	1000	1	0	0.0	0.0	100.0
3	2261.30000	61.43199	1000	1000000	0	0.0	0.0	0.0
4	1549.35000	87.90636	1000	1	0	0.0	0.0	100.0
5	3.94612	0.09619	1000	2	0	0.0	0.0	100.0

Table D.45: PBT_R_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2280.72500	12.20336	1000	1000000	0	0.0	0.0	0.0
2	1517.10000	17.65942	1000	1	0	0.0	0.0	100.0
3	2205.92500	25.65311	1000	1000000	0	0.0	0.0	0.0
4	1578.50000	73.52622	1000	1	0	0.0	0.0	100.0
5	9.47633	0.89059	1000	3	0	0.0	0.0	100.0

Table D.46: PBT_R_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2297.77500	20.68700	1000	1000000	0	0.0	0.0	0.0
2	1506.02500	14.85924	1000	1	0	0.0	0.0	100.0
3	2175.47500	39.46317	1000	1000000	0	0.0	0.0	0.0
4	1576.37500	74.99571	1000	1	0	0.0	0.0	100.0
5	10.10305	2.43468	1000	3	0	0.0	0.0	100.0

Table D.47: PBT_CG_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	13904.5000	122.69169	100	1000000	0	55.36	0.00	0.0
2	152.7450	0.70362	100	1	0	58.03	0.00	100.0
3	15798.0000	18.50676	100	1000000	0	62.81	0.00	0.0
4	154.9050	19.51898	100	1	0	58.40	0.00	100.0
5	142.2625	19.01071	1000	1	0	85.36	3.92	100.0

Table D.48: PBT_CG_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14031.7500	123.32553	100	1000000	0	55.36	0.0	0.0
2	152.3300	8.87367	100	1	0	58.03	0.0	100.0
3	15825.0000	65.91282	100	1000000	0	62.81	0.0	0.0
4	145.8925	7.48274	100	1	0	58.40	0.0	100.0
5	130.0625	5.96792	1000	2	0	85.00	0.0	100.0

Table D.49: PBT_CG_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	13887.2500	81.52109	100	1000000	0	55.36	0.0	0.0
2	166.2550	1.89211	100	1	0	58.03	0.0	100.0
3	15704.7500	73.53358	100	1000000	0	62.81	0.0	0.0
4	147.5050	5.07490	100	1	0	58.40	0.0	100.0
5	128.5325	10.48535	1000	3	0	89.28	0.0	100.0

Table D.50: PBT_CG_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	15183.0000	176.28812	1000	1000000	0	55.36	0.0	0.0
2	236.1925	10.49484	1000	1	0	58.03	0.0	100.0
3	16751.2500	64.35983	1000	1000000	0	62.81	0.0	0.0
4	235.7475	10.92278	1000	1	0	58.40	0.0	100.0
5	138.8725	15.64130	1000	1	0	83.39	3.4	100.0

Table D.51: PBT_CG_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	13938.7500	67.11324	100	1000000	0	55.36	0.0	0.0
2	168.5900	1.99481	100	1	0	58.03	0.0	100.0
3	15804.2500	162.27041	100	1000000	0	62.81	0.0	0.0
4	150.1575	6.28775	100	1	0	58.40	0.0	100.0
5	137.3300	9.93216	1000	3	0	89.28	0.0	100.0

Table D.52: PBT_CG_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	15075.500	55.63497	1000	1000000	0	55.36	0.0	0.0
2	224.355	1.65167	1000	1	0	58.03	0.0	100.0
3	16710.500	173.90299	1000	1000000	0	62.81	0.0	0.0
4	246.750	19.71166	1000	1	0	58.40	0.0	100.0
5	134.015	13.57683	1000	2	0	85.00	0.0	100.0

Table D.53: PBT_CG_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	15150.5000	131.72225	1000	1000000	0	55.36	0.0	0.0
2	230.7125	1.45092	1000	1	0	58.03	0.0	100.0
3	16647.7500	120.60758	1000	1000000	0	62.81	0.0	0.0
4	237.1000	8.45294	1000	1	0	58.40	0.0	100.0
5	146.3150	7.15756	1000	3	0	89.28	0.0	100.0

Table D.54: PBT_CG_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	15126.7500	157.74881	1000	1000000	0	55.36	0.0	0.0
2	231.3025	6.55705	1000	1	0	58.03	0.0	100.0
3	17026.0000	377.23202	1000	1000000	0	62.81	0.0	0.0
4	240.4025	7.85808	1000	1	0	58.40	0.0	100.0
5	131.5475	11.39531	1000	3	0	89.28	0.0	100.0

D.4 M3

Table D.55: M3 - Search strategies results.

Strategy	Min Mean Time (ms)	Std Min Mean Time (ms)	Max Mean Time (ms)	Std Max Mean Time (ms)	Min Sample Size	Max Sample Size	Min Mean Test Cases	Std Min Mean Test Cases	Max Mean Test Cases	Std Max Mean Test Cases	Min Mean Coverage (%)	Std Min Mean Coverage (%)	Max Mean Coverage (%)	Std Max Mean Coverage (%)	Bug Found
PBT_R_S4	1558.5250	75.39391	1558.5250	75.39391	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S1	525.2550	27.55804	525.2550	27.55804	100	100	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S8	1622.6750	44.64736	1622.6750	44.64736	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_CG_S4	239.8925	13.79632	239.8925	13.79632	1000	1000	1	0	1	0	78.89	0.0	78.89	0.0	True
PBT_CG_S6	227.2025	19.38684	227.2025	19.38684	1000	1000	1	0	1	0	78.89	0.0	78.89	0.0	True
PBT_CG_S3	150.1125	5.17616	150.1125	5.17616	100	100	1	0	1	0	78.89	0.0	78.89	0.0	True
PBT_R_S3	500.6025	12.78935	500.6025	12.78935	100	100	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S5	508.2875	24.14038	508.2875	24.14038	100	100	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S7	1568.5500	44.09504	1568.5500	44.09504	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_CG_S5	147.6125	13.77554	147.6125	13.77554	100	100	1	0	1	0	78.89	0.0	78.89	0.0	True
PBT_CG_S7	234.7775	6.19783	234.7775	6.19783	1000	1000	1	0	1	0	78.89	0.0	78.89	0.0	True
PBT_CG_S1	151.3400	16.67257	151.3400	16.67257	100	100	1	0	1	0	78.89	0.0	78.89	0.0	True
PBT_CG_S8	246.2800	19.83516	246.2800	19.83516	1000	1000	1	0	1	0	78.89	0.0	78.89	0.0	True
PBT_CG_S2	143.4150	5.81596	143.4150	5.81596	100	100	1	0	1	0	78.89	0.0	78.89	0.0	True
PBT_R_S6	1548.4500	63.04893	1548.4500	63.04893	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S2	507.8375	17.66754	507.8375	17.66754	100	100	1	0	1	0	0.00	0.0	0.00	0.0	True

Table D.56: M3 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-8.5794	0.900	-1072.3281	1055.1694	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-1064.0544	0.001	-1106.5401	-1021.5687	True
Random withConstantExtraction	Random withoutConstantExtraction	-15.0119	0.900	-1078.6895	1048.6657	False
Coverage-Guided withBias	Coverage-Guided withoutBias	2.9031	0.900	-86.5240	92.3303	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-88.9181	0.001	-99.6804	-78.1558	True
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-4.2331	0.900	-93.6073	85.1411	False
Coverage-Guided	Random	849.9438	0.001	416.7858	1283.1017	True

D.4.1 Search Strategies

Table D.57: PBT_R_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1152.97500	4.63647	100	1000000	0	0.0	0.0	0.0
2	1214.35000	9.26107	100	1000000	0	0.0	0.0	0.0
3	1349.90000	13.03668	100	1000000	0	0.0	0.0	0.0
4	525.25500	27.55804	100	1	0	0.0	0.0	100.0
5	3.00775	0.20841	1000	1000	0	0.0	0.0	0.0

Table D.58: PBT_R_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1131.25000	11.15112	100	1000000	0	0.0	0.0	0.0
2	1235.80000	38.06619	100	1000000	0	0.0	0.0	0.0
3	1372.17500	50.74679	100	1000000	0	0.0	0.0	0.0
4	507.83750	17.66754	100	1	0	0.0	0.0	100.0
5	3.05705	0.11553	1000	1000	0	0.0	0.0	0.0

Table D.59: PBT_R_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1126.10000	25.23579	100	1000000	0	0.0	0.0	0.0
2	1261.25000	91.98023	100	1000000	0	0.0	0.0	0.0
3	1328.55000	17.85070	100	1000000	0	0.0	0.0	0.0
4	500.60250	12.78935	100	1	0	0.0	0.0	100.0
5	5.68773	1.99196	1000	1000	0	0.0	0.0	0.0

Table D.60: PBT_R_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2296.25000	17.63895	1000	1000000	0	0.0	0.0	0.0
2	2215.32500	51.86855	1000	1000000	0	0.0	0.0	0.0
3	2285.25000	60.93220	1000	1000000	0	0.0	0.0	0.0
4	1558.52500	75.39391	1000	1	0	0.0	0.0	100.0
5	8.64695	2.02682	1000	1000	0	0.0	0.0	0.0

Table D.61: PBT_R_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1120.3750	11.06444	100	1000000	0	0.0	0.0	0.0
2	1205.0750	48.64701	100	1000000	0	0.0	0.0	0.0
3	1329.5750	52.18723	100	1000000	0	0.0	0.0	0.0
4	508.2875	24.14038	100	1	0	0.0	0.0	100.0
5	6.6917	2.28857	1000	1000	0	0.0	0.0	0.0

Table D.62: PBT_R_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2284.8750	28.56111	1000	1000000	0	0.0	0.0	0.0
2	2204.5750	34.28122	1000	1000000	0	0.0	0.0	0.0
3	2299.5750	41.34981	1000	1000000	0	0.0	0.0	0.0
4	1548.4500	63.04893	1000	1	0	0.0	0.0	100.0
5	9.3285	0.21752	1000	1000	0	0.0	0.0	0.0

Table D.63: PBT_R_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2298.625	13.29537	1000	1000000	0	0.0	0.0	0.0
2	2229.275	15.96204	1000	1000000	0	0.0	0.0	0.0
3	2303.500	33.22762	1000	1000000	0	0.0	0.0	0.0
4	1568.550	44.09504	1000	1	0	0.0	0.0	100.0
5	13.761	4.68138	1000	1000	0	0.0	0.0	0.0

Table D.64: PBT_R_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2279.97500	20.33929	1000	1000000	0	0.0	0.0	0.0
2	2290.07500	98.33136	1000	1000000	0	0.0	0.0	0.0
3	2377.07500	89.56228	1000	1000000	0	0.0	0.0	0.0
4	1622.67500	44.64736	1000	1	0	0.0	0.0	100.0
5	11.05925	0.73886	1000	1000	0	0.0	0.0	0.0

Table D.65: PBT_CG_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	13873.5000	107.76015	100	1000000	0	55.36	0.0	0.0
2	14816.7500	218.66341	100	1000000	0	75.89	0.0	0.0
3	15812.2500	168.17606	100	1000000	0	65.02	0.0	0.0
4	151.3400	16.67257	100	1	0	78.89	0.0	100.0
5	171.2325	17.68224	1000	1000	0	87.14	0.0	0.0

Table D.66: PBT_CG_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14074.0000	217.91283	100	1000000	0	55.36	0.0	0.0
2	14637.7500	141.69576	100	1000000	0	75.89	0.0	0.0
3	15676.0000	158.49132	100	1000000	0	65.02	0.0	0.0
4	143.4150	5.81596	100	1	0	78.89	0.0	100.0
5	165.7025	6.94461	1000	1000	0	87.14	0.0	0.0

Table D.67: PBT_CG_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14028.0000	119.73930	100	1000000	0	55.36	0.0	0.0
2	14798.2500	228.82458	100	1000000	0	75.89	0.0	0.0
3	15811.2500	228.39918	100	1000000	0	65.02	0.0	0.0
4	150.1125	5.17616	100	1	0	78.89	0.0	100.0
5	162.8725	8.64680	1000	1000	0	87.14	0.0	0.0

Table D.68: PBT_CG_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	15086.0000	127.78498	1000	1000000	0	55.36	0.0	0.0
2	15775.0000	406.45848	1000	1000000	0	75.89	0.0	0.0
3	16596.7500	300.32680	1000	1000000	0	65.02	0.0	0.0
4	239.8925	13.79632	1000	1	0	78.89	0.0	100.0
5	171.8075	8.61376	1000	1000	0	87.14	0.0	0.0

Table D.69: PBT_CG_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	13919.0000	163.26512	100	1000000	0	55.36	0.0	0.0
2	14994.2500	226.50980	100	1000000	0	75.89	0.0	0.0
3	16065.2500	219.74118	100	1000000	0	65.02	0.0	0.0
4	147.6125	13.77554	100	1	0	78.89	0.0	100.0
5	167.4575	11.69183	1000	1000	0	87.14	0.0	0.0

Table D.70: PBT_CG_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	15258.2500	73.05264	1000	1000000	0	55.36	0.0	0.0
2	15847.0000	243.73449	1000	1000000	0	75.89	0.0	0.0
3	16817.7500	136.98791	1000	1000000	0	65.02	0.0	0.0
4	227.2025	19.38684	1000	1	0	78.89	0.0	100.0
5	166.7175	7.42044	1000	1000	0	87.14	0.0	0.0

Table D.71: PBT_CG_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	15203.5000	159.47492	1000	1000000	0	55.36	0.0	0.0
2	16035.5000	223.55033	1000	1000000	0	75.89	0.0	0.0
3	16836.5000	106.04127	1000	1000000	0	65.02	0.0	0.0
4	234.7775	6.19783	1000	1	0	78.89	0.0	100.0
5	180.0500	10.96542	1000	1000	0	87.14	0.0	0.0

Table D.72: PBT_CG_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	15144.500	81.29422	1000	1000000	0	55.36	0.0	0.0
2	15952.000	85.64170	1000	1000000	0	75.89	0.0	0.0
3	16641.750	223.33201	1000	1000000	0	65.02	0.0	0.0
4	246.280	19.83516	1000	1	0	78.89	0.0	100.0
5	180.645	18.83455	1000	1000	0	87.14	0.0	0.0

D.5 M4

Table D.73: M4 - Search strategies results.

Strategy	Min Mean Time (ms)	Std Min Mean Time (ms)	Max Mean Time (ms)	Std Max Mean Time (ms)	Min Sample Size	Max Sample Size	Min Mean Test Cases	Std Min Mean Test Cases	Max Mean Test Cases	Std Max Mean Test Cases	Min Mean Coverage (%)	Std Min Mean Coverage (%)	Max Mean Coverage (%)	Std Max Mean Coverage (%)	Bug Found
PBT_R_S4	1.23025	0.00748	1.23025	0.00748	1000	1000	2	1	2	1	0.00	0.00	0.00	0.00	True
PBT_R_S1	0.56756	0.06284	0.56756	0.06284	1000	1000	1	0	1	0	0.00	0.00	0.00	0.00	True
PBT_R_S8	9.30852	3.47054	9.30852	3.47054	1000	1000	3	0	3	0	0.00	0.00	0.00	0.00	True
PBT_CG_S4	139.80750	5.69232	139.80750	5.69232	1000	1000	1	0	1	0	85.36	3.92	85.36	3.92	True
PBT_CG_S6	153.66500	8.32496	153.66500	8.32496	1000	1000	2	0	2	0	85.00	0.00	85.00	0.00	True
PBT_CG_S3	135.97750	6.74495	135.97750	6.74495	1000	1000	3	0	3	0	89.28	0.00	89.28	0.00	True
PBT_R_S3	1.96600	0.09622	1.96600	0.09622	1000	1000	3	0	3	0	0.00	0.00	0.00	0.00	True
PBT_R_S5	1.93605	0.05899	1.93605	0.05899	1000	1000	3	0	3	0	0.00	0.00	0.00	0.00	True
PBT_R_S7	5.16632	1.35024	5.16632	1.35024	1000	1000	3	0	3	0	0.00	0.00	0.00	0.00	True
PBT_CG_S5	134.40750	6.49717	134.40750	6.49717	1000	1000	3	0	3	0	89.28	0.00	89.28	0.00	True
PBT_CG_S7	138.68000	2.21571	138.68000	2.21571	1000	1000	3	0	3	0	89.28	0.00	89.28	0.00	True
PBT_CG_S1	136.22750	5.84777	136.22750	5.84777	1000	1000	1	0	1	0	83.39	3.40	83.39	3.40	True
PBT_CG_S8	145.65250	9.89896	145.65250	9.89896	1000	1000	3	0	3	0	89.28	0.00	89.28	0.00	True
PBT_CG_S2	135.17500	5.13449	135.17500	5.13449	1000	1000	2	0	2	0	85.00	0.00	85.00	0.00	True
PBT_R_S6	4.46168	5.54506	4.46168	5.54506	1000	1000	2	0	2	0	0.00	0.00	0.00	0.00	True
PBT_R_S2	0.54687	0.00723	0.54687	0.00723	1000	1000	2	0	2	0	0.00	0.00	0.00	0.00	True

Table D.74: M4 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-1.8307	0.4332	-7.1644	3.5029	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-3.7876	0.0685	-7.9682	0.3930	False
Random withConstantExtraction	Random withoutConstantExtraction	-2.8926	0.1941	-7.7344	1.9492	False
Coverage-Guided withBias	Coverage-Guided withoutBias	-4.5519	0.3693	-16.0283	6.9245	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-9.0044	0.0403	-17.4587	-0.5500	True
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	2.5394	0.6386	-9.5416	14.6203	False
Coverage-Guided	Random	-136.8012	0.0010	-142.3077	-131.2946	True

Table D.75: M4 - Tukey's test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-0.250	0.6844	-1.6179	1.1179	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-0.250	0.6844	-1.6179	1.1179	False
Random withConstantExtraction	Random withoutConstantExtraction	-1.250	0.0025	-1.8617	-0.6383	True
Coverage-Guided withBias	Coverage-Guided withoutBias	-0.500	0.4680	-2.0795	1.0795	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	0.000	0.9000	-1.6566	1.6566	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-1.500	0.0020	-2.2064	-0.7936	True
Coverage-Guided	Random	0.125	0.7742	-0.7526	1.0026	False

Table D.76: M4 - Tukey's test results concerning the *Coverage* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Coverage-Guided withBias	Coverage-Guided withoutBias	-0.3125	0.8785	-5.0149	4.3899	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-0.4925	0.8140	-5.1795	4.1945	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-4.5925	0.0010	-5.6710	-3.5140	True

D.5.1 Search Strategies

Table D.77: PBT_R_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1152.42500	2.44681	100	1000000	0	0.0	0.0	0.0
2	1234.60000	13.19166	100	1000000	0	0.0	0.0	0.0
3	1359.05000	12.84085	100	1000000	0	0.0	0.0	0.0
4	1440.17500	38.18857	100	1000000	0	0.0	0.0	0.0
5	0.56756	0.06284	1000	1	0	0.0	0.0	100.0

Table D.78: PBT_R_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1154.85000	5.14514	100	1000000	0	0.0	0.0	0.0
2	1272.82500	51.15454	100	1000000	0	0.0	0.0	0.0
3	1353.77500	6.94348	100	1000000	0	0.0	0.0	0.0
4	1402.30000	32.21506	100	1000000	0	0.0	0.0	0.0
5	0.54687	0.00723	1000	2	0	0.0	0.0	100.0

Table D.79: PBT_R_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1139.225	14.81560	100	1000000	0	0.0	0.0	0.0
2	1187.925	20.75023	100	1000000	0	0.0	0.0	0.0
3	1346.475	16.52700	100	1000000	0	0.0	0.0	0.0
4	1408.425	32.27246	100	1000000	0	0.0	0.0	0.0
5	1.966	0.09622	1000	3	0	0.0	0.0	100.0

Table D.80: PBT_R_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2320.00000	17.98235	1000	1000000	0	0.0	0.0	0.0
2	2247.30000	29.86997	1000	1000000	0	0.0	0.0	0.0
3	2358.17500	23.72208	1000	1000000	0	0.0	0.0	0.0
4	2402.45000	83.85048	1000	1000000	0	0.0	0.0	0.0
5	1.23025	0.00748	1000	2	1	0.0	0.0	100.0

Table D.81: PBT_R_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1132.62500	7.16532	100	1000000	0	0.0	0.0	0.0
2	1278.95000	67.59388	100	1000000	0	0.0	0.0	0.0
3	1342.62500	12.64879	100	1000000	0	0.0	0.0	0.0
4	1387.50000	35.60920	100	1000000	0	0.0	0.0	0.0
5	1.93605	0.05899	1000	3	0	0.0	0.0	100.0

Table D.82: PBT_R_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2263.50000	27.32005	1000	1000000	0	0.0	0.0	0.0
2	2300.22500	41.81647	1000	1000000	0	0.0	0.0	0.0
3	2400.40000	61.43969	1000	1000000	0	0.0	0.0	0.0
4	2445.42500	92.49528	1000	1000000	0	0.0	0.0	0.0
5	4.46168	5.54506	1000	2	0	0.0	0.0	100.0

Table D.83: PBT_R_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2334.17500	36.36636	1000	1000000	0	0.0	0.0	0.0
2	2266.30000	16.03356	1000	1000000	0	0.0	0.0	0.0
3	2349.50000	26.39669	1000	1000000	0	0.0	0.0	0.0
4	2423.55000	84.49889	1000	1000000	0	0.0	0.0	0.0
5	5.16632	1.35024	1000	3	0	0.0	0.0	100.0

Table D.84: PBT_R_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2337.90000	23.65819	1000	1000000	0	0.0	0.0	0.0
2	2273.70000	57.36972	1000	1000000	0	0.0	0.0	0.0
3	2315.00000	55.63785	1000	1000000	0	0.0	0.0	0.0
4	2337.82500	128.15544	1000	1000000	0	0.0	0.0	0.0
5	9.30852	3.47054	1000	3	0	0.0	0.0	100.0

Table D.85: PBT_CG_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14025.0000	100.38924	100	1000000	0	55.36	0.0	0.0
2	14887.2500	134.58524	100	1000000	0	75.00	0.0	0.0
3	16028.5000	122.28348	100	1000000	0	65.02	0.0	0.0
4	16213.0000	140.20877	100	1000000	0	82.19	0.0	0.0
5	136.2275	5.84777	1000	1	0	83.39	3.4	100.0

Table D.86: PBT_CG_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14080.750	86.20434	100	1000000	0	55.36	0.0	0.0
2	15002.500	59.06141	100	1000000	0	75.00	0.0	0.0
3	16019.000	114.90866	100	1000000	0	65.02	0.0	0.0
4	16292.750	150.32028	100	1000000	0	82.19	0.0	0.0
5	135.175	5.13449	1000	2	0	85.00	0.0	100.0

Table D.87: PBT_CG_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14094.5000	137.56544	100	1000000	0	55.36	0.0	0.0
2	14932.0000	84.06545	100	1000000	0	75.00	0.0	0.0
3	15900.2500	115.15289	100	1000000	0	65.02	0.0	0.0
4	16200.5000	88.05822	100	1000000	0	82.19	0.0	0.0
5	135.9775	6.74495	1000	3	0	89.28	0.0	100.0

Table D.88: PBT_CG_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	15190.5000	85.87345	1000	1000000	0	55.36	0.00	0.0
2	15857.5000	235.59340	1000	1000000	0	75.00	0.00	0.0
3	16829.2500	320.68082	1000	1000000	0	65.02	0.00	0.0
4	17149.0000	159.60420	1000	1000000	0	82.19	0.00	0.0
5	139.8075	5.69232	1000	1	0	85.36	3.92	100.0

Table D.89: PBT_CG_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14087.5000	101.14470	100	1000000	0	55.36	0.0	0.0
2	15010.7500	142.53837	100	1000000	0	75.00	0.0	0.0
3	15958.0000	101.05197	100	1000000	0	65.02	0.0	0.0
4	16251.7500	126.65776	100	1000000	0	82.19	0.0	0.0
5	134.4075	6.49717	1000	3	0	89.28	0.0	100.0

Table D.90: PBT_CG_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	15223.750	85.47916	1000	1000000	0	55.36	0.0	0.0
2	15859.500	231.96390	1000	1000000	0	75.00	0.0	0.0
3	16847.250	225.95616	1000	1000000	0	65.02	0.0	0.0
4	17125.000	243.09360	1000	1000000	0	82.19	0.0	0.0
5	153.665	8.32496	1000	2	0	85.00	0.0	100.0

Table D.91: PBT_CG_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	15277.75	148.01921	1000	1000000	0	55.36	0.0	0.0
2	15888.75	129.50169	1000	1000000	0	75.00	0.0	0.0
3	17001.75	196.71347	1000	1000000	0	65.02	0.0	0.0
4	17257.25	107.76682	1000	1000000	0	82.19	0.0	0.0
5	138.68	2.21571	1000	3	0	89.28	0.0	100.0

Table D.92: PBT_CG_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	15343.0000	94.65992	1000	1000000	0	55.36	0.0	0.0
2	16165.5000	217.85603	1000	1000000	0	75.00	0.0	0.0
3	16987.2500	142.51908	1000	1000000	0	65.02	0.0	0.0
4	17397.7500	23.62599	1000	1000000	0	82.19	0.0	0.0
5	145.6525	9.89896	1000	3	0	89.28	0.0	100.0

Appendix E

RERS Challenge 2012 Results

This appendix presents all the data associated with the results achieved for the `RERS Challenge 2012` benchmark, including each search strategy's results and all the data extracted for each property.

E.1 Search Strategies

Table E.1: PBT_R_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	393.9325	32.07710	100000	159	124	0.0	0.0	100.0
2	412.8925	32.22700	100000	1423	1713	0.0	0.0	100.0
3	403.8150	18.16233	100000	2085	640	0.0	0.0	100.0
4	411.6800	54.44969	100000	1693	1602	0.0	0.0	100.0
5	377.9925	27.32005	100000	1164	796	0.0	0.0	100.0
6	385.8100	44.62707	100000	2032	1520	0.0	0.0	100.0
7	447.5600	35.89580	100000	10758	7595	0.0	0.0	100.0
8	358.1750	21.61788	100000	2614	694	0.0	0.0	100.0
9	390.2400	10.55762	100000	4942	3446	0.0	0.0	100.0
10	460.3050	83.42366	100000	19114	13906	0.0	0.0	100.0
11	528.9725	138.97205	100000	50037	35085	0.0	0.0	100.0
12	667.4625	41.55455	100000	81706	14112	0.0	0.0	75.0
13	457.6550	50.52644	100000	33037	13223	0.0	0.0	100.0
14	688.7500	41.70118	100000	100000	0	0.0	0.0	100.0

Table E.2: PBT_R_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	380.1525	18.36412	100000	77	55	0.0	0.0	100.0
2	442.2600	32.42350	100000	2441	1465	0.0	0.0	100.0
3	400.9725	34.67144	100000	2903	1681	0.0	0.0	100.0
4	422.1750	17.55370	100000	1076	547	0.0	0.0	100.0
5	388.2125	7.84445	100000	951	605	0.0	0.0	100.0
6	374.8400	14.13720	100000	1075	871	0.0	0.0	100.0
7	432.6975	33.22803	100000	4430	3991	0.0	0.0	100.0
8	427.7775	54.64568	100000	12654	10314	0.0	0.0	100.0
9	438.6950	59.57937	100000	9251	8153	0.0	0.0	100.0
10	449.6325	39.92745	100000	15935	8322	0.0	0.0	100.0
11	582.5150	52.14665	100000	51183	13206	0.0	0.0	100.0
12	539.9125	120.09113	100000	42006	38156	0.0	0.0	75.0
13	562.1150	127.32386	100000	55350	36986	0.0	0.0	75.0
14	749.4375	26.56175	100000	100000	0	0.0	0.0	100.0

Table E.3: PBT_R_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	467.9375	9.87591	100000	167	120	0.0	0.0	100.0
2	517.9600	21.78375	100000	1854	1331	0.0	0.0	100.0
3	460.6450	15.23027	100000	2024	966	0.0	0.0	100.0
4	479.8725	12.72619	100000	2419	2251	0.0	0.0	100.0
5	456.0250	24.96654	100000	2071	1706	0.0	0.0	100.0
6	471.6500	28.64914	100000	757	681	0.0	0.0	100.0
7	484.4825	57.89764	100000	5319	3115	0.0	0.0	100.0
8	490.3025	35.34251	100000	6542	5025	0.0	0.0	100.0
9	568.6800	66.41922	100000	23289	14459	0.0	0.0	100.0
10	464.0775	21.45486	100000	4670	2125	0.0	0.0	100.0
11	635.4625	117.69989	100000	39154	18447	0.0	0.0	100.0
12	586.7250	84.54167	100000	47273	31180	0.0	0.0	75.0
13	531.9950	96.22079	100000	28969	34362	0.0	0.0	100.0
14	727.3500	12.41440	100000	100000	0	0.0	0.0	100.0

Table E.4: PBT_R_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	385.0975	3.08527	100000	171	152	0.0	0.0	100.0
2	436.6575	33.28765	100000	1878	1947	0.0	0.0	100.0
3	390.0700	9.03580	100000	1288	148	0.0	0.0	100.0
4	421.6350	7.60549	100000	1903	1433	0.0	0.0	100.0
5	394.6775	17.02406	100000	2434	1474	0.0	0.0	100.0
6	377.8500	15.43295	100000	2387	1771	0.0	0.0	100.0
7	474.4675	38.69733	100000	16635	9542	0.0	0.0	100.0
8	391.3875	25.37675	100000	4816	5180	0.0	0.0	100.0
9	433.8550	13.95068	100000	12088	3788	0.0	0.0	100.0
10	414.0475	28.80660	100000	6634	4552	0.0	0.0	100.0
11	604.4425	107.01968	100000	59731	31560	0.0	0.0	75.0
12	550.6925	21.48788	100000	54114	5118	0.0	0.0	100.0
13	597.9700	93.79242	100000	77532	26303	0.0	0.0	50.0
14	711.2650	6.92202	100000	100000	0	0.0	0.0	100.0

Table E.5: PBT_R_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	458.6750	8.28620	100000	59	43	0.0	0.0	100.0
2	509.8700	27.03682	100000	1928	1237	0.0	0.0	100.0
3	454.8350	16.00207	100000	2387	2837	0.0	0.0	100.0
4	485.0525	9.77133	100000	1673	1047	0.0	0.0	100.0
5	443.3925	10.92460	100000	2675	2268	0.0	0.0	100.0
6	469.0850	3.51639	100000	1516	724	0.0	0.0	100.0
7	502.8175	66.07355	100000	15045	15407	0.0	0.0	100.0
8	501.3275	38.23797	100000	9131	4668	0.0	0.0	100.0
9	491.4775	31.86412	100000	11445	7717	0.0	0.0	100.0
10	496.7000	56.51171	100000	15645	9450	0.0	0.0	100.0
11	690.8600	124.27227	100000	62917	35469	0.0	0.0	75.0
12	570.7700	88.11978	100000	46844	31212	0.0	0.0	100.0
13	571.0400	52.47686	100000	36929	21058	0.0	0.0	100.0
14	726.7125	24.55282	100000	100000	0	0.0	0.0	100.0

Table E.6: PBT_R_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	373.1325	11.18853	100000	172	101	0.0	0.0	100.0
2	419.3450	37.13545	100000	1524	987	0.0	0.0	100.0
3	398.6000	24.50212	100000	2263	1118	0.0	0.0	100.0
4	399.1975	38.08935	100000	2583	2145	0.0	0.0	100.0
5	375.4550	26.02378	100000	1697	1437	0.0	0.0	100.0
6	358.8400	9.13951	100000	1066	298	0.0	0.0	100.0
7	416.2650	25.87847	100000	4814	2411	0.0	0.0	100.0
8	412.4550	39.74699	100000	9723	3427	0.0	0.0	100.0
9	410.5775	13.88820	100000	6976	3463	0.0	0.0	100.0
10	461.1475	45.82050	100000	20666	7222	0.0	0.0	100.0
11	495.4525	48.99413	100000	34820	14118	0.0	0.0	100.0
12	604.8050	127.09808	100000	63615	32247	0.0	0.0	75.0
13	621.9750	81.60283	100000	87907	20945	0.0	0.0	25.0
14	702.4675	17.08435	100000	100000	0	0.0	0.0	100.0

Table E.7: PBT_R_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	459.1500	6.19081	100000	100	51	0.0	0.0	100.0
2	501.6425	13.07965	100000	1594	822	0.0	0.0	100.0
3	450.8725	13.33362	100000	1523	741	0.0	0.0	100.0
4	477.8950	14.06529	100000	1129	625	0.0	0.0	100.0
5	431.7475	10.11910	100000	1763	823	0.0	0.0	100.0
6	467.0800	17.42301	100000	1944	1219	0.0	0.0	100.0
7	509.8675	40.65263	100000	16958	11416	0.0	0.0	100.0
8	501.7125	17.26197	100000	12705	3662	0.0	0.0	100.0
9	468.0350	24.49254	100000	5037	1639	0.0	0.0	100.0
10	456.0625	11.73834	100000	11077	6080	0.0	0.0	100.0
11	636.7575	172.10953	100000	48746	43225	0.0	0.0	75.0
12	627.9400	57.43666	100000	60853	23139	0.0	0.0	75.0
13	617.0325	74.09574	100000	53819	28811	0.0	0.0	75.0
14	749.2000	25.31554	100000	100000	0	0.0	0.0	100.0

Table E.8: PBT_R_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	461.6150	3.79228	100000	138	113	0.0	0.0	100.0
2	503.8075	24.84724	100000	1680	1595	0.0	0.0	100.0
3	447.8250	10.30285	100000	1157	770	0.0	0.0	100.0
4	480.7500	3.03032	100000	1687	512	0.0	0.0	100.0
5	437.7400	13.92795	100000	1109	812	0.0	0.0	100.0
6	467.9275	8.62972	100000	879	908	0.0	0.0	100.0
7	538.7125	44.58544	100000	19632	6725	0.0	0.0	100.0
8	498.1700	21.70526	100000	12278	3486	0.0	0.0	100.0
9	496.1650	37.65684	100000	10324	6874	0.0	0.0	100.0
10	446.2100	16.12909	100000	4337	3453	0.0	0.0	100.0
11	663.4625	97.85277	100000	52514	28169	0.0	0.0	75.0
12	524.6300	141.21896	100000	32670	39981	0.0	0.0	75.0
13	577.0700	65.76352	100000	41820	23891	0.0	0.0	100.0
14	762.3000	25.53956	100000	100000	0	0.0	0.0	100.0

Table E.9: PBT_CG_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	1968.550	50.00582	100000	143	130	16.16	0.0	100.0
2	2253.675	546.10550	100000	2437	2077	16.16	0.0	100.0
3	2029.950	404.66158	100000	2353	1660	16.16	0.0	100.0
4	2590.400	102.70884	100000	4597	566	16.16	0.0	100.0
5	1885.325	258.15980	100000	1369	1231	16.16	0.0	100.0
6	2005.400	660.22688	100000	2112	2910	16.16	0.0	100.0
7	3121.100	957.98647	100000	6808	4163	16.16	0.0	100.0
8	7021.525	5742.88157	100000	23089	23679	16.16	0.0	100.0
9	3879.550	1370.53801	100000	9672	5212	16.16	0.0	100.0
10	2916.025	1036.42268	100000	5943	4608	16.16	0.0	100.0
11	24457.750	2496.47005	100000	94322	9834	16.16	0.0	25.0
12	10212.875	9095.30658	100000	35881	37130	16.16	0.0	75.0
13	24660.500	2375.07374	100000	94576	9393	16.16	0.0	25.0
14	27596.500	283.26180	100000	100000	0	16.48	0.0	100.0

Table E.10: PBT_CG_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	1992.225	22.29130	100000	94	42	16.16	0.0	100.0
2	2076.875	383.66308	100000	1957	1590	16.16	0.0	100.0
3	1536.075	77.63203	100000	352	313	16.16	0.0	100.0
4	1629.100	219.09998	100000	677	855	16.16	0.0	100.0
5	1937.100	223.16822	100000	1445	835	16.16	0.0	100.0
6	1777.000	236.31675	100000	1331	1054	16.16	0.0	100.0
7	7526.600	5355.41294	100000	25243	22076	16.16	0.0	100.0
8	4761.300	1433.89503	100000	13533	6095	16.16	0.0	100.0
9	6027.925	5535.39585	100000	18839	22577	16.16	0.0	100.0
10	7179.625	5347.09089	100000	23491	21945	16.16	0.0	100.0
11	22933.250	4856.88235	100000	88554	19824	16.16	0.0	25.0
12	13132.375	9417.05259	100000	47682	39025	16.16	0.0	75.0
13	22292.250	3715.90032	100000	85332	14786	16.16	0.0	50.0
14	27683.250	298.21081	100000	100000	0	16.48	0.0	100.0

Table E.11: PBT_CG_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	2041.725	29.85677	100000	44	22	16.16	0.0	100.0
2	2170.100	194.75128	100000	1888	847	16.16	0.0	100.0
3	1858.925	341.75977	100000	1461	1414	16.16	0.0	100.0
4	2393.325	327.88676	100000	3115	1228	16.16	0.0	100.0
5	1775.975	169.18725	100000	1123	582	16.16	0.0	100.0
6	1665.075	146.18460	100000	735	673	16.16	0.0	100.0
7	4557.725	1241.78670	100000	11749	5258	16.16	0.0	100.0
8	6998.100	6735.02401	100000	23003	28195	16.16	0.0	100.0
9	3467.050	756.04273	100000	8175	3169	16.16	0.0	100.0
10	9816.875	9374.83622	100000	33680	38728	16.16	0.0	75.0
11	16140.650	9677.42352	100000	60372	39628	16.16	0.0	50.0
12	23714.250	2174.40306	100000	91553	8832	16.16	0.0	50.0
13	17232.250	9331.10932	100000	64675	38900	16.16	0.0	75.0
14	27513.750	220.91217	100000	100000	0	16.48	0.0	100.0

Table E.12: PBT_CG_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	2008.175	57.77916	100000	78	64	16.16	0.0	100.0
2	2102.625	331.33368	100000	1919	1278	16.16	0.0	100.0
3	1823.150	241.47875	100000	1485	932	16.16	0.0	100.0
4	1577.450	80.39233	100000	466	315	16.16	0.0	100.0
5	1863.675	256.71076	100000	1181	875	16.16	0.0	100.0
6	1940.000	324.76986	100000	2008	1337	16.16	0.0	100.0
7	6699.100	5830.71533	100000	21400	23750	16.16	0.0	100.0
8	7368.000	5901.31988	100000	23970	24444	16.16	0.0	100.0
9	5063.100	2342.20808	100000	14724	9429	16.16	0.0	100.0
10	6168.550	5106.55480	100000	19021	20867	16.16	0.0	100.0
11	20092.975	10601.28368	100000	75309	42764	16.16	0.0	25.0
12	14431.600	10211.79244	100000	53160	41923	16.16	0.0	75.0
13	20478.925	10116.52775	100000	76461	40769	16.16	0.0	25.0
14	27500.250	245.58234	100000	100000	0	16.48	0.0	100.0

Table E.13: PBT_CG_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	2061.025	36.52112	100000	171	178	16.16	0.0	100.0
2	1965.675	304.84358	100000	1157	1304	16.16	0.0	100.0
3	1879.600	123.37388	100000	1504	447	16.16	0.0	100.0
4	2074.475	306.24182	100000	1718	1197	16.16	0.0	100.0
5	1609.700	61.59497	100000	456	206	16.16	0.0	100.0
6	1598.950	53.91783	100000	415	239	16.16	0.0	100.0
7	5559.025	5258.47766	100000	16168	21916	16.16	0.0	100.0
8	7101.275	4141.35481	100000	23196	16864	16.16	0.0	100.0
9	4260.975	1469.26625	100000	11363	6080	16.16	0.0	100.0
10	7933.275	7381.72228	100000	26383	30569	16.16	0.0	100.0
11	20736.300	8043.52543	100000	79079	32908	16.16	0.0	50.0
12	14396.900	11816.10371	100000	51964	48088	16.16	0.0	50.0
13	21830.750	3699.03226	100000	83721	15756	16.16	0.0	75.0
14	27462.250	225.85878	100000	100000	0	16.48	0.0	100.0

Table E.14: PBT_CG_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	1970.900	44.99205	100000	86	91	16.16	0.0	100.0
2	1970.775	236.88334	100000	1350	765	16.16	0.0	100.0
3	1952.750	252.54812	100000	1806	1001	16.16	0.0	100.0
4	1908.350	390.62798	100000	1807	1533	16.16	0.0	100.0
5	1854.750	209.40636	100000	1244	801	16.16	0.0	100.0
6	2291.600	534.03333	100000	3300	2365	16.16	0.0	100.0
7	7043.825	6446.43314	100000	23228	26820	16.16	0.0	100.0
8	10700.650	5812.52976	100000	37707	24292	16.16	0.0	100.0
9	8431.675	6912.17262	100000	29146	28277	16.16	0.0	100.0
10	3271.825	1392.62506	100000	7399	5909	16.16	0.0	100.0
11	17888.525	8905.18879	100000	66871	36562	16.16	0.0	50.0
12	19808.475	8720.56473	100000	75725	36123	16.16	0.0	50.0
13	17744.525	8953.86911	100000	66096	36189	16.16	0.0	50.0
14	27443.500	134.76368	100000	100000	0	16.48	0.0	100.0

Table E.15: PBT_CG_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	2118.300	100.44073	100000	339	415	16.16	0.0	100.0
2	2081.825	462.60906	100000	1776	1772	16.16	0.0	100.0
3	1975.650	241.88481	100000	2006	990	16.16	0.0	100.0
4	2007.050	233.07622	100000	1487	927	16.16	0.0	100.0
5	1900.575	86.79480	100000	1667	347	16.16	0.0	100.0
6	1847.700	205.74177	100000	1513	880	16.16	0.0	100.0
7	4467.525	1317.14550	100000	11729	5479	16.16	0.0	100.0
8	11850.200	7657.27807	100000	43301	31847	16.16	0.0	100.0
9	8582.475	7814.77489	100000	29457	32943	16.16	0.0	100.0
10	4207.450	1123.68389	100000	10898	5053	16.16	0.0	100.0
11	25672.500	212.97476	100000	100000	0	16.16	0.0	0.0
12	17479.075	7336.85371	100000	65991	30304	16.16	0.0	75.0
13	16731.650	9183.27871	100000	62795	38110	16.16	0.0	75.0
14	27514.750	155.58981	100000	100000	0	16.48	0.0	100.0

Table E.16: PBT_CG_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
1	2053.825	30.61873	100000	165	149	16.16	0.0	100.0
2	2146.125	548.26302	100000	2018	2236	16.16	0.0	100.0
3	1685.175	179.50799	100000	742	670	16.16	0.0	100.0
4	1923.600	302.80286	100000	1235	1205	16.16	0.0	100.0
5	2077.700	238.71545	100000	2401	1047	16.16	0.0	100.0
6	1807.775	427.37140	100000	1334	1833	16.16	0.0	100.0
7	10388.175	5276.67869	100000	36234	21623	16.16	0.0	100.0
8	3484.700	1538.19573	100000	8354	6459	16.16	0.0	100.0
9	3799.475	1227.48596	100000	9470	5269	16.16	0.0	100.0
10	3515.175	1657.20123	100000	7814	6847	16.16	0.0	100.0
11	21043.700	8447.35172	100000	80130	34415	16.16	0.0	25.0
12	23598.750	2435.06842	100000	90498	9641	16.16	0.0	50.0
13	24302.000	1674.64369	100000	93129	7657	16.16	0.0	50.0
14	27811.250	368.44564	100000	100000	0	16.48	0.0	100.0

E.2 Properties

E.2.1 P1

Table E.17: P1 - Search strategies results.

Strategy	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
PBT_R_S4	385.0975	3.08527	100000.0	171.0	152.0	0.00	0.0	100.0
PBT_R_S1	393.9325	32.07710	100000.0	159.0	124.0	0.00	0.0	100.0
PBT_R_S8	461.6150	3.79228	100000.0	138.0	113.0	0.00	0.0	100.0
PBT_CG_S4	2008.1750	57.77916	100000.0	78.0	64.0	16.16	0.0	100.0
PBT_CG_S6	1970.9000	44.99205	100000.0	86.0	91.0	16.16	0.0	100.0
PBT_CG_S3	2041.7250	29.85677	100000.0	44.0	22.0	16.16	0.0	100.0
PBT_R_S3	467.9375	9.87591	100000.0	167.0	120.0	0.00	0.0	100.0
PBT_R_S5	458.6750	8.28620	100000.0	59.0	43.0	0.00	0.0	100.0
PBT_R_S7	459.1500	6.19081	100000.0	100.0	51.0	0.00	0.0	100.0
PBT_CG_S5	2061.0250	36.52112	100000.0	171.0	178.0	16.16	0.0	100.0
PBT_CG_S7	2118.3000	100.44073	100000.0	339.0	415.0	16.16	0.0	100.0
PBT_CG_S1	1968.5500	50.00582	100000.0	143.0	130.0	16.16	0.0	100.0
PBT_CG_S8	2053.8250	30.61873	100000.0	165.0	149.0	16.16	0.0	100.0
PBT_CG_S2	1992.2250	22.29130	100000.0	94.0	42.0	16.16	0.0	100.0
PBT_R_S6	373.1325	11.18853	100000.0	172.0	101.0	0.00	0.0	100.0
PBT_R_S2	380.1525	18.36412	100000.0	77.0	55.0	0.00	0.0	100.0

Table E.18: P1 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	8.1356	0.8179	-71.0259	87.2972	False
Random withPairWiseTesting	Random withoutPairWiseTesting	5.4256	0.8753	-73.9672	84.8185	False
Random withConstantExtraction	Random withoutConstantExtraction	-78.7656	0.0010	-90.6664	-66.8649	True
Coverage-Guided withBias	Coverage-Guided withoutBias	14.6937	0.7315	-80.4034	109.7909	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-21.9188	0.5990	-115.6178	71.7803	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-83.7562	0.0050	-131.2807	-36.2318	True
Coverage-Guided	Random	-1604.3791	0.0010	-1655.0439	-1553.7142	True

Table E.19: P1 - Tukey's test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	37.750	0.2706	-38.3858	113.8858	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-29.750	0.3957	-109.3460	49.8460	False
Random withConstantExtraction	Random withoutConstantExtraction	28.750	0.4128	-51.2118	108.7118	False
Coverage-Guided withBias	Coverage-Guided withoutBias	22.000	0.7742	-148.5833	192.5833	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-54.000	0.4494	-217.3149	109.3149	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-79.500	0.2494	-232.0604	73.0604	False
Coverage-Guided	Random	-9.625	0.8027	-87.4624	68.2124	False

E.2.2 P2

Table E.20: P2 - Search strategies results.

Strategy	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
PBT_R_S4	436.6575	33.28765	100000.0	1878.0	1947.0	0.00	0.0	100.0
PBT_R_S1	412.8925	32.22700	100000.0	1423.0	1713.0	0.00	0.0	100.0
PBT_R_S8	503.8075	24.84724	100000.0	1680.0	1595.0	0.00	0.0	100.0
PBT_CG_S4	2102.6250	331.33368	100000.0	1919.0	1278.0	16.16	0.0	100.0
PBT_CG_S6	1970.7750	236.88334	100000.0	1350.0	765.0	16.16	0.0	100.0
PBT_CG_S3	2170.1000	194.75128	100000.0	1888.0	847.0	16.16	0.0	100.0
PBT_R_S3	517.9600	21.78375	100000.0	1854.0	1331.0	0.00	0.0	100.0
PBT_R_S5	509.8700	27.03682	100000.0	1928.0	1237.0	0.00	0.0	100.0
PBT_R_S7	501.6425	13.07965	100000.0	1594.0	822.0	0.00	0.0	100.0
PBT_CG_S5	1965.6750	304.84358	100000.0	1157.0	1304.0	16.16	0.0	100.0
PBT_CG_S7	2081.8250	462.60906	100000.0	1776.0	1772.0	16.16	0.0	100.0
PBT_CG_S1	2253.6750	546.10550	100000.0	2437.0	2077.0	16.16	0.0	100.0
PBT_CG_S8	2146.1250	548.26302	100000.0	2018.0	2236.0	16.16	0.0	100.0
PBT_CG_S2	2076.8750	383.66308	100000.0	1957.0	1590.0	16.16	0.0	100.0
PBT_R_S6	419.3450	37.13545	100000.0	1524.0	987.0	0.00	0.0	100.0
PBT_R_S2	442.2600	32.42350	100000.0	2441.0	1465.0	0.00	0.0	100.0

Table E.21: P2 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-1.5325	0.9000	-84.2325	81.1675	False
Random withPairWiseTesting	Random withoutPairWiseTesting	5.3825	0.8806	-77.1567	87.9217	False
Random withConstantExtraction	Random withoutConstantExtraction	-80.5312	0.0010	-99.7665	-61.2960	True
Coverage-Guided withBias	Coverage-Guided withoutBias	112.1938	0.1032	-30.7750	255.1625	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	41.2438	0.6004	-135.6844	218.1719	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	10.0563	0.8969	-171.3276	191.4401	False
Coverage-Guided	Random	-1627.9050	0.0010	-1708.8960	-1546.9140	True

Table E.22: P2 - Tukey's test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-206.0	0.4018	-764.7741	352.7741	False
Random withPairWiseTesting	Random withoutPairWiseTesting	242.5	0.3172	-301.4602	786.4602	False
Random withConstantExtraction	Random withoutConstantExtraction	52.5	0.8417	-540.6482	645.6482	False
Coverage-Guided withBias	Coverage-Guided withoutBias	384.5	0.1916	-254.9010	1023.9010	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	94.0	0.7774	-645.9631	833.9631	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	206.0	0.5099	-510.9492	922.9492	False
Coverage-Guided	Random	-22.5	0.9000	-409.7637	364.7637	False

E.2.3 P3

Table E.23: P3 - Search strategies results.

Strategy	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
PBT_R_S4	390.0700	9.03580	100000.0	1288.0	148.0	0.00	0.0	100.0
PBT_R_S1	403.8150	18.16233	100000.0	2085.0	640.0	0.00	0.0	100.0
PBT_R_S8	447.8250	10.30285	100000.0	1157.0	770.0	0.00	0.0	100.0
PBT_CG_S4	1823.1500	241.47875	100000.0	1485.0	932.0	16.16	0.0	100.0
PBT_CG_S6	1952.7500	252.54812	100000.0	1806.0	1001.0	16.16	0.0	100.0
PBT_CG_S3	1858.9250	341.75977	100000.0	1461.0	1414.0	16.16	0.0	100.0
PBT_R_S3	460.6450	15.23027	100000.0	2024.0	966.0	0.00	0.0	100.0
PBT_R_S5	454.8350	16.00207	100000.0	2387.0	2837.0	0.00	0.0	100.0
PBT_R_S7	450.8725	13.33362	100000.0	1523.0	741.0	0.00	0.0	100.0
PBT_CG_S5	1879.6000	123.37388	100000.0	1504.0	447.0	16.16	0.0	100.0
PBT_CG_S7	1975.6500	241.88481	100000.0	2006.0	990.0	16.16	0.0	100.0
PBT_CG_S1	2029.9500	404.66158	100000.0	2353.0	1660.0	16.16	0.0	100.0
PBT_CG_S8	1685.1750	179.50799	100000.0	742.0	670.0	16.16	0.0	100.0
PBT_CG_S2	1536.0750	77.63203	100000.0	352.0	313.0	16.16	0.0	100.0
PBT_R_S6	398.6000	24.50212	100000.0	2263.0	1118.0	0.00	0.0	100.0
PBT_R_S2	400.9725	34.67144	100000.0	2903.0	1681.0	0.00	0.0	100.0

Table E.24: P3 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.7925	0.9000	-55.2097	56.7947	False
Random withPairWiseTesting	Random withoutPairWiseTesting	8.2250	0.7416	-47.1768	63.6268	False
Random withConstantExtraction	Random withoutConstantExtraction	-55.1800	0.0010	-65.1018	-45.2582	True
Coverage-Guided withBias	Coverage-Guided withoutBias	158.5188	0.1863	-101.4313	418.4688	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-33.0438	0.8072	-335.6322	269.5447	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-14.3562	0.9000	-318.4018	289.6893	False
Coverage-Guided	Random	-1416.7050	0.0010	-1542.2831	-1291.1269	True

Table E.25: P3 - Tukey's test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-447.500	0.3226	-1463.6917	568.6917	False
Random withPairWiseTesting	Random withoutPairWiseTesting	792.000	0.0473	13.1957	1570.8043	True
Random withConstantExtraction	Random withoutConstantExtraction	362.000	0.4311	-687.6248	1411.6248	False
Coverage-Guided withBias	Coverage-Guided withoutBias	725.250	0.1189	-250.6944	1701.1944	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-92.250	0.8624	-1304.2138	1119.7138	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	70.750	0.8921	-1142.6557	1284.1557	False
Coverage-Guided	Random	490.125	0.1378	-177.8087	1158.0587	False

E.2.4 P4

Table E.26: P4 - Search strategies results.

Strategy	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
PBT_R_S4	421.6350	7.60549	100000.0	1903.0	1433.0	0.00	0.0	100.0
PBT_R_S1	411.6800	54.44969	100000.0	1693.0	1602.0	0.00	0.0	100.0
PBT_R_S8	480.7500	3.03032	100000.0	1687.0	512.0	0.00	0.0	100.0
PBT_CG_S4	1577.4500	80.39233	100000.0	466.0	315.0	16.16	0.0	100.0
PBT_CG_S6	1908.3500	390.62798	100000.0	1807.0	1533.0	16.16	0.0	100.0
PBT_CG_S3	2393.3250	327.88676	100000.0	3115.0	1228.0	16.16	0.0	100.0
PBT_R_S3	479.8725	12.72619	100000.0	2419.0	2251.0	0.00	0.0	100.0
PBT_R_S5	485.0525	9.77133	100000.0	1673.0	1047.0	0.00	0.0	100.0
PBT_R_S7	477.8950	14.06529	100000.0	1129.0	625.0	0.00	0.0	100.0
PBT_CG_S5	2074.4750	306.24182	100000.0	1718.0	1197.0	16.16	0.0	100.0
PBT_CG_S7	2007.0500	233.07622	100000.0	1487.0	927.0	16.16	0.0	100.0
PBT_CG_S1	2590.4000	102.70884	100000.0	4597.0	566.0	16.16	0.0	100.0
PBT_CG_S8	1923.6000	302.80286	100000.0	1235.0	1205.0	16.16	0.0	100.0
PBT_CG_S2	1629.1000	219.09998	100000.0	677.0	855.0	16.16	0.0	100.0
PBT_R_S6	399.1975	38.08935	100000.0	2583.0	2145.0	0.00	0.0	100.0
PBT_R_S2	422.1750	17.55370	100000.0	1076.0	547.0	0.00	0.0	100.0

Table E.27: P4 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.9769	0.9000	-67.5507	69.5045	False
Random withPairWiseTesting	Random withoutPairWiseTesting	4.8256	0.8716	-63.5392	73.1904	False
Random withConstantExtraction	Random withoutConstantExtraction	-67.2206	0.0010	-80.9274	-53.5139	True
Coverage-Guided withBias	Coverage-Guided withoutBias	258.1750	0.3277	-334.7367	851.0867	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	317.7125	0.2168	-245.6076	881.0326	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-173.2875	0.5252	-796.2586	449.6836	False
Coverage-Guided	Random	-1565.6866	0.0010	-1829.5061	-1301.8670	True

Table E.28: P4 - Tukey's test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	31.250	0.9000	-970.8515	1033.3515	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-110.250	0.8048	-1106.7701	886.2701	False
Random withConstantExtraction	Random withoutConstantExtraction	86.750	0.8445	-912.0854	1085.5854	False
Coverage-Guided withBias	Coverage-Guided withoutBias	1057.000	0.3062	-1255.8374	3369.8374	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	1278.000	0.2048	-920.6967	3476.6967	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-2.000	0.9000	-2544.4617	2540.4617	False
Coverage-Guided	Random	-117.375	0.8298	-1226.3002	991.5502	False

E.2.5 P5

Table E.29: P5 - Search strategies results.

Strategy	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
PBT_R_S4	394.6775	17.02406	100000.0	2434.0	1474.0	0.00	0.0	100.0
PBT_R_S1	377.9925	27.32005	100000.0	1164.0	796.0	0.00	0.0	100.0
PBT_R_S8	437.7400	13.92795	100000.0	1109.0	812.0	0.00	0.0	100.0
PBT_CG_S4	1863.6750	256.71076	100000.0	1181.0	875.0	16.16	0.0	100.0
PBT_CG_S6	1854.7500	209.40636	100000.0	1244.0	801.0	16.16	0.0	100.0
PBT_CG_S3	1775.9750	169.18725	100000.0	1123.0	582.0	16.16	0.0	100.0
PBT_R_S3	456.0250	24.96654	100000.0	2071.0	1706.0	0.00	0.0	100.0
PBT_R_S5	443.3925	10.92460	100000.0	2675.0	2268.0	0.00	0.0	100.0
PBT_R_S7	431.7475	10.11910	100000.0	1763.0	823.0	0.00	0.0	100.0
PBT_CG_S5	1609.7000	61.59497	100000.0	456.0	206.0	16.16	0.0	100.0
PBT_CG_S7	1900.5750	86.79480	100000.0	1667.0	347.0	16.16	0.0	100.0
PBT_CG_S1	1885.3250	258.15980	100000.0	1369.0	1231.0	16.16	0.0	100.0
PBT_CG_S8	2077.7000	238.71545	100000.0	2401.0	1047.0	16.16	0.0	100.0
PBT_CG_S2	1937.1000	223.16822	100000.0	1445.0	835.0	16.16	0.0	100.0
PBT_R_S6	375.4550	26.02378	100000.0	1697.0	1437.0	0.00	0.0	100.0
PBT_R_S2	388.2125	7.84445	100000.0	951.0	605.0	0.00	0.0	100.0

Table E.30: P5 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	3.9106	0.8812	-56.4117	64.2329	False
Random withPairWiseTesting	Random withoutPairWiseTesting	6.5006	0.8089	-53.5982	66.5995	False
Random withConstantExtraction	Random withoutConstantExtraction	-58.1419	0.0010	-74.8952	-41.3886	True
Coverage-Guided withBias	Coverage-Guided withoutBias	-13.4250	0.8997	-263.1014	236.2514	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-122.1500	0.2199	-340.3905	96.0905	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	44.2250	0.6895	-201.8774	290.3274	False
Coverage-Guided	Random	-1449.9447	0.0010	-1554.3206	-1345.5688	True

Table E.31: P5 - Tukey's test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	250.00	0.6296	-909.5682	1409.5682	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-35.50	0.9000	-1221.1261	1150.1261	False
Random withConstantExtraction	Random withoutConstantExtraction	-343.00	0.4878	-1478.5903	792.5903	False
Coverage-Guided withBias	Coverage-Guided withoutBias	-51.50	0.9000	-1074.8264	971.8264	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-525.00	0.1947	-1405.2263	355.2263	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-102.00	0.8224	-1121.5398	917.5398	False
Coverage-Guided	Random	372.25	0.2299	-263.7370	1008.2370	False

E.2.6 P6

Table E.32: P6 - Search strategies results.

Strategy	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
PBT_R_S4	377.8500	15.43295	100000.0	2387.0	1771.0	0.00	0.0	100.0
PBT_R_S1	385.8100	44.62707	100000.0	2032.0	1520.0	0.00	0.0	100.0
PBT_R_S8	467.9275	8.62972	100000.0	879.0	908.0	0.00	0.0	100.0
PBT_CG_S4	1940.0000	324.76986	100000.0	2008.0	1337.0	16.16	0.0	100.0
PBT_CG_S6	2291.6000	534.03333	100000.0	3300.0	2365.0	16.16	0.0	100.0
PBT_CG_S3	1665.0750	146.18460	100000.0	735.0	673.0	16.16	0.0	100.0
PBT_R_S3	471.6500	28.64914	100000.0	757.0	681.0	0.00	0.0	100.0
PBT_R_S5	469.0850	3.51639	100000.0	1516.0	724.0	0.00	0.0	100.0
PBT_R_S7	467.0800	17.42301	100000.0	1944.0	1219.0	0.00	0.0	100.0
PBT_CG_S5	1598.9500	53.91783	100000.0	415.0	239.0	16.16	0.0	100.0
PBT_CG_S7	1847.7000	205.74177	100000.0	1513.0	880.0	16.16	0.0	100.0
PBT_CG_S1	2005.4000	660.22688	100000.0	2112.0	2910.0	16.16	0.0	100.0
PBT_CG_S8	1807.7750	427.37140	100000.0	1334.0	1833.0	16.16	0.0	100.0
PBT_CG_S2	1777.0000	236.31675	100000.0	1331.0	1054.0	16.16	0.0	100.0
PBT_R_S6	358.8400	9.13951	100000.0	1066.0	298.0	0.00	0.0	100.0
PBT_R_S2	374.8400	14.13720	100000.0	1075.0	871.0	0.00	0.0	100.0

Table E.33: P6 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	7.9244	0.8505	-87.2885	103.1372	False
Random withPairWiseTesting	Random withoutPairWiseTesting	7.4219	0.8594	-87.8314	102.6751	False
Random withConstantExtraction	Random withoutConstantExtraction	-94.6006	0.0010	-108.6612	-80.5401	True
Coverage-Guided withBias	Coverage-Guided withoutBias	-4.2875	0.9000	-409.6533	401.0783	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-210.1625	0.1886	-556.9542	136.6292	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	273.6250	0.0667	-25.7522	573.0022	False
Coverage-Guided	Random	-1445.0522	0.0010	-1614.0465	-1276.0579	True

Table E.34: P6 - Tukey's test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	646.0	0.1392	-281.5786	1573.5786	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-224.0	0.6519	-1331.5964	883.5964	False
Random withConstantExtraction	Random withoutConstantExtraction	366.0	0.4343	-703.1894	1435.1894	False
Coverage-Guided withBias	Coverage-Guided withoutBias	-3.0	0.9000	-1677.7481	1671.7481	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-890.5	0.1755	-2309.4672	528.4672	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	1188.5	0.0490	7.3017	2369.6983	True
Coverage-Guided	Random	-136.5	0.7376	-956.2458	683.2458	False

E.2.7 P7

Table E.35: P7 - Search strategies results.

Strategy	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
PBT_R_S4	474.4675	38.69733	100000.0	16635.0	9542.0	0.00	0.0	100.0
PBT_R_S1	447.5600	35.89580	100000.0	10758.0	7595.0	0.00	0.0	100.0
PBT_R_S8	538.7125	44.58544	100000.0	19632.0	6725.0	0.00	0.0	100.0
PBT_CG_S4	6699.1000	5830.71533	100000.0	21400.0	23750.0	16.16	0.0	100.0
PBT_CG_S6	7043.8250	6446.43314	100000.0	23228.0	26820.0	16.16	0.0	100.0
PBT_CG_S3	4557.7250	1241.78670	100000.0	11749.0	5258.0	16.16	0.0	100.0
PBT_R_S3	484.4825	57.89764	100000.0	5319.0	3115.0	0.00	0.0	100.0
PBT_R_S5	502.8175	66.07355	100000.0	15045.0	15407.0	0.00	0.0	100.0
PBT_R_S7	509.8675	40.65263	100000.0	16958.0	11416.0	0.00	0.0	100.0
PBT_CG_S5	5559.0250	5258.47766	100000.0	16168.0	21916.0	16.16	0.0	100.0
PBT_CG_S7	4467.5250	1317.14550	100000.0	11729.0	5479.0	16.16	0.0	100.0
PBT_CG_S1	3121.1000	957.98647	100000.0	6808.0	4163.0	16.16	0.0	100.0
PBT_CG_S8	10388.1750	5276.67869	100000.0	36234.0	21623.0	16.16	0.0	100.0
PBT_CG_S2	7526.6000	5355.41294	100000.0	25243.0	22076.0	16.16	0.0	100.0
PBT_R_S6	416.2650	25.87847	100000.0	4814.0	2411.0	0.00	0.0	100.0
PBT_R_S2	432.6975	33.22803	100000.0	4430.0	3991.0	0.00	0.0	100.0

Table E.36: P7 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	6.4712	0.8501	-71.0410	83.9835	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-17.9387	0.5939	-93.6277	57.7502	False
Random withConstantExtraction	Random withoutConstantExtraction	-66.2225	0.0074	-107.1340	-25.3110	True
Coverage-Guided withBias	Coverage-Guided withoutBias	-2918.0438	0.0586	-5981.8873	145.7998	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-1958.5437	0.2484	-5707.7276	1790.6401	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-145.4562	0.9000	-4371.9312	4081.0187	False
Coverage-Guided	Random	-5694.5256	0.0010	-7410.7365	-3978.3148	True

Table E.37: P7 - Tukey's test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	1437.25	0.7803	-10036.5115	12911.0115	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-5621.75	0.2224	-15729.7151	4486.2151	False
Random withConstantExtraction	Random withoutConstantExtraction	-5079.25	0.2768	-15469.8245	5311.3245	False
Coverage-Guided withBias	Coverage-Guided withoutBias	-12296.75	0.0545	-24918.1698	324.6698	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-8155.75	0.2484	-23770.3543	7458.8543	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	199.75	0.9000	-17411.3935	17810.8935	False
Coverage-Guided	Random	-7371.00	0.0856	-15919.8033	1177.8033	False

E.2.8 P8

Table E.38: P8 - Search strategies results.

Strategy	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
PBT_R_S4	391.3875	25.37675	100000.0	4816.0	5180.0	0.00	0.0	100.0
PBT_R_S1	358.1750	21.61788	100000.0	2614.0	694.0	0.00	0.0	100.0
PBT_R_S8	498.1700	21.70526	100000.0	12278.0	3486.0	0.00	0.0	100.0
PBT_CG_S4	7368.0000	5901.31988	100000.0	23970.0	24444.0	16.16	0.0	100.0
PBT_CG_S6	10700.6500	5812.52976	100000.0	37707.0	24292.0	16.16	0.0	100.0
PBT_CG_S3	6998.1000	6735.02401	100000.0	23003.0	28195.0	16.16	0.0	100.0
PBT_R_S3	490.3025	35.34251	100000.0	6542.0	5025.0	0.00	0.0	100.0
PBT_R_S5	501.3275	38.23797	100000.0	9131.0	4668.0	0.00	0.0	100.0
PBT_R_S7	501.7125	17.26197	100000.0	12705.0	3662.0	0.00	0.0	100.0
PBT_CG_S5	7101.2750	4141.35481	100000.0	23196.0	16864.0	16.16	0.0	100.0
PBT_CG_S7	11850.2000	7657.27807	100000.0	43301.0	31847.0	16.16	0.0	100.0
PBT_CG_S1	7021.5250	5742.88157	100000.0	23089.0	23679.0	16.16	0.0	100.0
PBT_CG_S8	3484.7000	1538.19573	100000.0	8354.0	6459.0	16.16	0.0	100.0
PBT_CG_S2	4761.3000	1433.89503	100000.0	13533.0	6095.0	16.16	0.0	100.0
PBT_R_S6	412.4550	39.74699	100000.0	9723.0	3427.0	0.00	0.0	100.0
PBT_R_S2	427.7775	54.64568	100000.0	12654.0	10314.0	0.00	0.0	100.0

Table E.39: P8 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-24.5381	0.5965	-128.7742	79.6980	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-6.5356	0.8874	-113.4159	100.3447	False
Random withConstantExtraction	Random withoutConstantExtraction	-100.4294	0.0010	-137.8617	-62.9971	True
Coverage-Guided withBias	Coverage-Guided withoutBias	1797.4750	0.3984	-3041.1428	6636.0928	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-1880.3375	0.3755	-6687.4271	2926.7521	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	104.3000	0.9000	-5055.6889	5264.2889	False
Coverage-Guided	Random	-6963.0553	0.0010	-9057.6169	-4868.4937	True

Table E.40: P8 - Tukey's test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-4277.25	0.1176	-10007.3922	1452.8922	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-2145.25	0.4706	-8964.2304	4673.7304	False
Random withConstantExtraction	Random withoutConstantExtraction	-2712.25	0.3544	-9326.6347	3902.1347	False
Coverage-Guided withBias	Coverage-Guided withoutBias	7643.25	0.3849	-12313.7456	27600.2456	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-7627.75	0.3859	-27590.6625	12335.1625	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	111.25	0.9000	-21256.1479	21478.6479	False
Coverage-Guided	Random	-15711.25	0.0024	-24853.4916	-6569.0084	True

E.2.9 P9

Table E.41: P9 - Search strategies results.

Strategy	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
PBT_R_S4	433.8550	13.95068	100000.0	12088.0	3788.0	0.00	0.0	100.0
PBT_R_S1	390.2400	10.55762	100000.0	4942.0	3446.0	0.00	0.0	100.0
PBT_R_S8	496.1650	37.65684	100000.0	10324.0	6874.0	0.00	0.0	100.0
PBT_CG_S4	5063.1000	2342.20808	100000.0	14724.0	9429.0	16.16	0.0	100.0
PBT_CG_S6	8431.6750	6912.17262	100000.0	29146.0	28277.0	16.16	0.0	100.0
PBT_CG_S3	3467.0500	756.04273	100000.0	8175.0	3169.0	16.16	0.0	100.0
PBT_R_S3	568.6800	66.41922	100000.0	23289.0	14459.0	0.00	0.0	100.0
PBT_R_S5	491.4775	31.86412	100000.0	11445.0	7717.0	0.00	0.0	100.0
PBT_R_S7	468.0350	24.49254	100000.0	5037.0	1639.0	0.00	0.0	100.0
PBT_CG_S5	4260.9750	1469.26625	100000.0	11363.0	6080.0	16.16	0.0	100.0
PBT_CG_S7	8582.4750	7814.77489	100000.0	29457.0	32943.0	16.16	0.0	100.0
PBT_CG_S1	3879.5500	1370.53801	100000.0	9672.0	5212.0	16.16	0.0	100.0
PBT_CG_S8	3799.4750	1227.48596	100000.0	9470.0	5269.0	16.16	0.0	100.0
PBT_CG_S2	6027.9250	5535.39585	100000.0	18839.0	22577.0	16.16	0.0	100.0
PBT_R_S6	410.5775	13.88820	100000.0	6976.0	3463.0	0.00	0.0	100.0
PBT_R_S2	438.6950	59.57937	100000.0	9251.0	8153.0	0.00	0.0	100.0

Table E.42: P9 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	5.9738	0.8953	-100.0054	111.9529	False
Random withPairWiseTesting	Random withoutPairWiseTesting	20.1150	0.6673	-84.1127	124.3427	False
Random withConstantExtraction	Random withoutConstantExtraction	-87.7475	0.0116	-147.6120	-27.8830	True
Coverage-Guided withBias	Coverage-Guided withoutBias	-381.9688	0.8230	-4213.9535	3450.0160	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-2060.3062	0.1724	-5315.1133	1194.5008	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	823.0688	0.6243	-2939.0694	4585.2069	False
Coverage-Guided	Random	-4976.8125	0.0010	-6539.9368	-3413.6882	True

Table E.43: P9 - Tukey's test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	1840.00	0.7058	-8985.9118	12665.9118	False
Random withPairWiseTesting	Random withoutPairWiseTesting	3625.50	0.4250	-6740.8998	13991.8998	False
Random withConstantExtraction	Random withoutConstantExtraction	-4209.50	0.3491	-14353.2779	5934.2779	False
Coverage-Guided withBias	Coverage-Guided withoutBias	-1697.50	0.8139	-17839.6655	14444.6655	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-8687.00	0.1722	-22403.4040	5029.4040	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	3479.00	0.6232	-12375.5568	19333.5568	False
Coverage-Guided	Random	-5936.75	0.1316	-13888.1108	2014.6108	False

E.2.10 P10

Table E.44: P10 - Search strategies results.

Strategy	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
PBT_R_S4	414.0475	28.80660	100000.0	6634.0	4552.0	0.00	0.0	100.0
PBT_R_S1	460.3050	83.42366	100000.0	19114.0	13906.0	0.00	0.0	100.0
PBT_R_S8	446.2100	16.12909	100000.0	4337.0	3453.0	0.00	0.0	100.0
PBT_CG_S4	6168.5500	5106.55480	100000.0	19021.0	20867.0	16.16	0.0	100.0
PBT_CG_S6	3271.8250	1392.62506	100000.0	7399.0	5909.0	16.16	0.0	100.0
PBT_CG_S3	9816.8750	9374.83622	100000.0	33680.0	38728.0	16.16	0.0	75.0
PBT_R_S3	464.0775	21.45486	100000.0	4670.0	2125.0	0.00	0.0	100.0
PBT_R_S5	496.7000	56.51171	100000.0	15645.0	9450.0	0.00	0.0	100.0
PBT_R_S7	456.0625	11.73834	100000.0	11077.0	6080.0	0.00	0.0	100.0
PBT_CG_S5	7933.2750	7381.72228	100000.0	26383.0	30569.0	16.16	0.0	100.0
PBT_CG_S7	4207.4500	1123.68389	100000.0	10898.0	5053.0	16.16	0.0	100.0
PBT_CG_S1	2916.0250	1036.42268	100000.0	5943.0	4608.0	16.16	0.0	100.0
PBT_CG_S8	3515.1750	1657.20123	100000.0	7814.0	6847.0	16.16	0.0	100.0
PBT_CG_S2	7179.6250	5347.09089	100000.0	23491.0	21945.0	16.16	0.0	100.0
PBT_R_S6	461.1475	45.82050	100000.0	20666.0	7222.0	0.00	0.0	100.0
PBT_R_S2	449.6325	39.92745	100000.0	15935.0	8322.0	0.00	0.0	100.0

Table E.45: P10 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-14.7994	0.4015	-54.9138	25.3151	False
Random withPairWiseTesting	Random withoutPairWiseTesting	23.3119	0.1627	-12.5410	59.1648	False
Random withConstantExtraction	Random withoutConstantExtraction	-19.4794	0.2571	-57.5462	18.5874	False
Coverage-Guided withBias	Coverage-Guided withoutBias	302.2500	0.8827	-4423.8555	5028.3555	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	2670.7000	0.1459	-1242.0487	6583.4487	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-1484.1875	0.4502	-5981.8600	3013.4850	False
Coverage-Guided	Random	-5170.0772	0.0010	-7091.6992	-3248.4552	True

Table E.46: P10 - Tukey's test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-3772.000	0.4549	-15326.5773	7782.5773	False
Random withPairWiseTesting	Random withoutPairWiseTesting	3162.500	0.5397	-8573.1696	14898.1696	False
Random withConstantExtraction	Random withoutConstantExtraction	6655.000	0.1606	-3519.0088	16829.0088	False
Coverage-Guided withBias	Coverage-Guided withoutBias	1113.750	0.8930	-18176.7817	20404.2817	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	11091.250	0.1373	-4739.2714	26921.7714	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-5730.250	0.4762	-24185.4865	12724.9865	False
Coverage-Guided	Random	-4568.875	0.3080	-13830.9844	4693.2344	False

Table E.47: P10 - Tukey's test results concerning the *Satisfiability* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Coverage-Guided withBias	Coverage-Guided withoutBias	-6.250	0.3559	-21.5432	9.0432	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-6.250	0.3559	-21.5432	9.0432	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	6.250	0.3559	-9.0432	21.5432	False
Coverage-Guided	Random	3.125	0.3343	-3.5775	9.8275	False

E.2.11 P11

Table E.48: P11 - Search strategies results.

Strategy	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
PBT_R_S4	604.4425	107.01968	100000.0	59731.0	31560.0	0.00	0.0	75.0
PBT_R_S1	528.9725	138.97205	100000.0	50037.0	35085.0	0.00	0.0	100.0
PBT_R_S8	663.4625	97.85277	100000.0	52514.0	28169.0	0.00	0.0	75.0
PBT_CG_S4	20092.9750	10601.28368	100000.0	75309.0	42764.0	16.16	0.0	25.0
PBT_CG_S6	17888.5250	8905.18879	100000.0	66871.0	36562.0	16.16	0.0	50.0
PBT_CG_S3	16140.6500	9677.42352	100000.0	60372.0	39628.0	16.16	0.0	50.0
PBT_R_S3	635.4625	117.69989	100000.0	39154.0	18447.0	0.00	0.0	100.0
PBT_R_S5	690.8600	124.27227	100000.0	62917.0	35469.0	0.00	0.0	75.0
PBT_R_S7	636.7575	172.10953	100000.0	48746.0	43225.0	0.00	0.0	75.0
PBT_CG_S5	20736.3000	8043.52543	100000.0	79079.0	32908.0	16.16	0.0	50.0
PBT_CG_S7	25672.5000	212.97476	100000.0	100000.0	0.0	16.16	0.0	0.0
PBT_CG_S1	24457.7500	2496.47005	100000.0	94322.0	9834.0	16.16	0.0	25.0
PBT_CG_S8	21043.7000	8447.35172	100000.0	80130.0	34415.0	16.16	0.0	25.0
PBT_CG_S2	22933.2500	4856.88235	100000.0	88554.0	19824.0	16.16	0.0	25.0
PBT_R_S6	495.4525	48.99413	100000.0	34820.0	14118.0	0.00	0.0	100.0
PBT_R_S2	582.5150	52.14665	100000.0	51183.0	13206.0	0.00	0.0	100.0

Table E.49: P11 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-6.6637	0.8999	-130.8785	117.5510	False
Random withPairWiseTesting	Random withoutPairWiseTesting	9.4238	0.8626	-114.6125	133.4600	False
Random withConstantExtraction	Random withoutConstantExtraction	-103.7900	0.0101	-172.5208	-35.0592	True
Coverage-Guided withBias	Coverage-Guided withoutBias	940.5250	0.7235	-4962.0936	6843.1436	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-107.4375	0.9000	-6083.3988	5868.5238	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	444.8375	0.8648	-5515.5457	6405.2207	False
Coverage-Guided	Random	-20515.9656	0.0010	-22941.6497	-18090.2816	True

Table E.50: P11 - Tukey's test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-941.500	0.8998	-18487.5996	16604.5996	False
Random withPairWiseTesting	Random withoutPairWiseTesting	1870.000	0.8108	-15601.7093	19341.7093	False
Random withConstantExtraction	Random withoutConstantExtraction	-1890.000	0.8088	-19359.5617	15579.5617	False
Coverage-Guided withBias	Coverage-Guided withoutBias	3842.250	0.7302	-20896.3470	28580.8470	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	4.250	0.9000	-25030.3259	25038.8259	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	1368.750	0.8980	-23628.4593	26365.9593	False
Coverage-Guided	Random	-30691.875	0.0010	-43102.1009	-18281.6491	True

Table E.51: P11 - Tukey's test results concerning the *Satisfiability* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.00	0.9000	-24.9737	24.9737	False
Random withPairWiseTesting	Random withoutPairWiseTesting	12.50	0.2070	-9.1278	34.1278	False
Random withConstantExtraction	Random withoutConstantExtraction	12.50	0.2070	-9.1278	34.1278	False
Coverage-Guided withBias	Coverage-Guided withoutBias	-12.50	0.3559	-43.0864	18.0864	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	12.50	0.3559	-18.0864	43.0864	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	0.00	0.9000	-33.0371	33.0371	False
Coverage-Guided	Random	56.25	0.0010	39.4460	73.0540	True

E.2.12 P12

Table E.52: P12 - Search strategies results.

Strategy	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
PBT_R_S4	550.6925	21.48788	100000.0	54114.0	5118.0	0.00	0.0	100.0
PBT_R_S1	667.4625	41.55455	100000.0	81706.0	14112.0	0.00	0.0	75.0
PBT_R_S8	524.6300	141.21896	100000.0	32670.0	39981.0	0.00	0.0	75.0
PBT_CG_S4	14431.6000	10211.79244	100000.0	53160.0	41923.0	16.16	0.0	75.0
PBT_CG_S6	19808.4750	8720.56473	100000.0	75725.0	36123.0	16.16	0.0	50.0
PBT_CG_S3	23714.2500	2174.40306	100000.0	91553.0	8832.0	16.16	0.0	50.0
PBT_R_S3	586.7250	84.54167	100000.0	47273.0	31180.0	0.00	0.0	75.0
PBT_R_S5	570.7700	88.11978	100000.0	46844.0	31212.0	0.00	0.0	100.0
PBT_R_S7	627.9400	57.43666	100000.0	60853.0	23139.0	0.00	0.0	75.0
PBT_CG_S5	14396.9000	11816.10371	100000.0	51964.0	48088.0	16.16	0.0	50.0
PBT_CG_S7	17479.0750	7336.85371	100000.0	65991.0	30304.0	16.16	0.0	75.0
PBT_CG_S1	10212.8750	9095.30658	100000.0	35881.0	37130.0	16.16	0.0	75.0
PBT_CG_S8	23598.7500	2435.06842	100000.0	90498.0	9641.0	16.16	0.0	50.0
PBT_CG_S2	13132.3750	9417.05259	100000.0	47682.0	39025.0	16.16	0.0	75.0
PBT_R_S6	604.8050	127.09808	100000.0	63615.0	32247.0	0.00	0.0	75.0
PBT_R_S2	539.9125	120.09113	100000.0	42006.0	38156.0	0.00	0.0	75.0

Table E.53: P12 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	48.1756	0.1698	-27.4019	123.7532	False
Random withPairWiseTesting	Random withoutPairWiseTesting	14.2006	0.7215	-74.2683	102.6695	False
Random withConstantExtraction	Random withoutConstantExtraction	13.2019	0.7408	-75.4212	101.8250	False
Coverage-Guided withBias	Coverage-Guided withoutBias	-1274.6750	0.7569	-10425.0992	7875.7492	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-3465.3750	0.3604	-12030.8920	5100.1420	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-5400.9125	0.1285	-12900.4559	2098.6309	False
Coverage-Guided	Random	-16512.6703	0.0010	-20261.4399	-12763.9007	True

Table E.54: P12 - Tukey's test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	14702.750	0.1869	-9447.4795	38852.9795	False
Random withPairWiseTesting	Random withoutPairWiseTesting	1644.250	0.8922	-26573.6841	29862.1841	False
Random withConstantExtraction	Random withoutConstantExtraction	13450.250	0.2339	-11417.8226	38318.3226	False
Coverage-Guided withBias	Coverage-Guided withoutBias	-4821.000	0.7770	-42699.2155	33057.2155	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-14573.500	0.3514	-49872.3989	20725.3989	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-21889.500	0.1379	-53191.4193	9412.4193	False
Coverage-Guided	Random	-10421.625	0.2656	-29697.7061	8854.4561	False

Table E.55: P12 - Tukey's test results concerning the *Satisfiability* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.00	0.9000	-21.6278	21.6278	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.00	0.9000	-21.6278	21.6278	False
Random withConstantExtraction	Random withoutConstantExtraction	0.00	0.9000	-21.6278	21.6278	False
Coverage-Guided withBias	Coverage-Guided withoutBias	12.50	0.2070	-9.1278	34.1278	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	0.00	0.9000	-24.9737	24.9737	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	12.50	0.2070	-9.1278	34.1278	False
Coverage-Guided	Random	18.75	0.0096	5.3451	32.1549	True

E.2.13 P13

Table E.56: P13 - Search strategies results.

Strategy	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
PBT_R_S4	597.9700	93.79242	100000.0	77532.0	26303.0	0.00	0.0	50.0
PBT_R_S1	457.6550	50.52644	100000.0	33037.0	13223.0	0.00	0.0	100.0
PBT_R_S8	577.0700	65.76352	100000.0	41820.0	23891.0	0.00	0.0	100.0
PBT_CG_S4	20478.9250	10116.52775	100000.0	76461.0	40769.0	16.16	0.0	25.0
PBT_CG_S6	17744.5250	8953.86911	100000.0	66096.0	36189.0	16.16	0.0	50.0
PBT_CG_S3	17232.2500	9331.10932	100000.0	64675.0	38900.0	16.16	0.0	75.0
PBT_R_S3	531.9950	96.22079	100000.0	28969.0	34362.0	0.00	0.0	100.0
PBT_R_S5	571.0400	52.47686	100000.0	36929.0	21058.0	0.00	0.0	100.0
PBT_R_S7	617.0325	74.09574	100000.0	53819.0	28811.0	0.00	0.0	75.0
PBT_CG_S5	21830.7500	3699.03226	100000.0	83721.0	15756.0	16.16	0.0	75.0
PBT_CG_S7	16731.6500	9183.27871	100000.0	62795.0	38110.0	16.16	0.0	75.0
PBT_CG_S1	24660.5000	2375.07374	100000.0	94576.0	9393.0	16.16	0.0	25.0
PBT_CG_S8	24302.0000	1674.64369	100000.0	93129.0	7657.0	16.16	0.0	50.0
PBT_CG_S2	22292.2500	3715.90032	100000.0	85332.0	14786.0	16.16	0.0	50.0
PBT_R_S6	621.9750	81.60283	100000.0	87907.0	20945.0	0.00	0.0	25.0
PBT_R_S2	562.1150	127.32386	100000.0	55350.0	36986.0	0.00	0.0	75.0

Table E.57: P13 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-31.8869	0.4391	-126.0593	62.2855	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-72.8106	0.0391	-140.5811	-5.0401	True
Random withConstantExtraction	Random withoutConstantExtraction	-14.3556	0.7459	-112.7295	84.0183	False
Coverage-Guided withBias	Coverage-Guided withoutBias	-1766.5500	0.4692	-7363.1462	3830.0462	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	1689.6625	0.4897	-3930.5750	7309.9000	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	1269.8875	0.6195	-4459.5803	6999.3553	False
Coverage-Guided	Random	-20091.9997	0.0010	-22473.3934	-17710.6060	True

Table E.58: P13 - Tukey's test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-7162.25	0.6842	-46324.7558	32000.2558	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-26698.25	0.0691	-56254.8864	2858.3864	False
Random withConstantExtraction	Random withoutConstantExtraction	23072.25	0.1327	-9388.2326	55532.7326	False
Coverage-Guided withBias	Coverage-Guided withoutBias	-7442.75	0.4527	-30125.6942	15240.1942	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	7455.75	0.4518	-15222.9335	30134.4335	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	4536.25	0.6664	-18900.0397	27972.5397	False
Coverage-Guided	Random	-26427.75	0.0094	-45262.2614	-7593.2386	True

Table E.59: P13 - Tukey's test results concerning the *Satisfiability* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	6.25	0.7897	-45.9862	58.4862	False
Random withPairWiseTesting	Random withoutPairWiseTesting	31.25	0.1210	-11.0950	73.5950	False
Random withConstantExtraction	Random withoutConstantExtraction	-31.25	0.1210	-73.5950	11.0950	False
Coverage-Guided withBias	Coverage-Guided withoutBias	-6.25	0.7184	-44.7370	32.2370	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	6.25	0.7184	-32.2370	44.7370	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-31.25	0.0170	-54.6107	-7.8893	True
Coverage-Guided	Random	25.00	0.0632	-1.5694	51.5694	False

E.2.14 P14

Table E.60: P14 - Search strategies results.

Strategy	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Satisfiability (%)
PBT_R_S4	711.2650	6.92202	100000.0	100000.0	0.0	0.00	0.0	100.0
PBT_R_S1	688.7500	41.70118	100000.0	100000.0	0.0	0.00	0.0	100.0
PBT_R_S8	762.3000	25.53956	100000.0	100000.0	0.0	0.00	0.0	100.0
PBT_CG_S4	27500.2500	245.58234	100000.0	100000.0	0.0	16.48	0.0	100.0
PBT_CG_S6	27443.5000	134.76368	100000.0	100000.0	0.0	16.48	0.0	100.0
PBT_CG_S3	27513.7500	220.91217	100000.0	100000.0	0.0	16.48	0.0	100.0
PBT_R_S3	727.3500	12.41440	100000.0	100000.0	0.0	0.00	0.0	100.0
PBT_R_S5	726.7125	24.55282	100000.0	100000.0	0.0	0.00	0.0	100.0
PBT_R_S7	749.2000	25.31554	100000.0	100000.0	0.0	0.00	0.0	100.0
PBT_CG_S5	27462.2500	225.85878	100000.0	100000.0	0.0	16.48	0.0	100.0
PBT_CG_S7	27514.7500	155.58981	100000.0	100000.0	0.0	16.48	0.0	100.0
PBT_CG_S1	27596.5000	283.26180	100000.0	100000.0	0.0	16.48	0.0	100.0
PBT_CG_S8	27811.2500	368.44564	100000.0	100000.0	0.0	16.48	0.0	100.0
PBT_CG_S2	27683.2500	298.21081	100000.0	100000.0	0.0	16.48	0.0	100.0
PBT_R_S6	702.4675	17.08435	100000.0	100000.0	0.0	0.00	0.0	100.0
PBT_R_S2	749.4375	26.56175	100000.0	100000.0	0.0	0.00	0.0	100.0

Table E.61: P14 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-16.0881	0.4142	-60.9694	28.7931	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-8.2456	0.6963	-55.2005	38.7093	False
Random withConstantExtraction	Random withoutConstantExtraction	-28.4106	0.1195	-66.7139	9.8927	False
Coverage-Guided withBias	Coverage-Guided withoutBias	-68.7500	0.4816	-293.0371	155.5371	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-3.5000	0.9000	-238.0401	231.0401	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-19.6250	0.8493	-253.3705	214.1205	False
Coverage-Guided	Random	-26838.5022	0.0010	-26935.6240	-26741.3804	True

Appendix F

Triangle Classification Program Results

This appendix presents all the data associated with the results achieved for all the `Triangle Classification Program` benchmark variants, including each search strategy's results and all the data extracted concerning all the evaluation metrics.

F.1 M0

Table F.1: M0 - Search strategies results.

Strategy	Min Mean Time (ms)	Std Min Mean Time (ms)	Max Mean Time (ms)	Std Max Mean Time (ms)	Min Sample Size	Max Sample Size	Min Mean Test Cases	Std Min Mean Test Cases	Max Mean Test Cases	Std Max Mean Test Cases	Min Mean Coverage (%)	Std Min Mean Coverage (%)	Max Mean Coverage (%)	Std Max Mean Coverage (%)	Bug Found
PBT_R_S4	308.4775	17.57305	308.4775	17.57305	1000	1000	2	1	2	1	0.00	0.0	0.00	0.0	True
PBT_R_S1	330.7450	9.63550	330.7450	9.63550	1000	1000	3	2	3	2	0.00	0.0	0.00	0.0	True
PBT_R_S8	320.5425	21.79394	320.5425	21.79394	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_CG_S4	193.9850	11.81765	193.9850	11.81765	1000	1000	1	0	1	0	82.41	0.0	82.41	0.0	True
PBT_CG_S6	166.3475	10.19603	166.3475	10.19603	1000	1000	1	0	1	0	84.26	0.0	84.26	0.0	True
PBT_CG_S3	151.1775	7.37422	151.1775	7.37422	1000	1000	1	0	1	0	84.26	0.0	84.26	0.0	True
PBT_R_S3	335.8425	8.57494	335.8425	8.57494	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S5	324.1475	18.28314	324.1475	18.28314	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S7	309.1175	21.80078	309.1175	21.80078	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_CG_S5	152.8800	1.98526	152.8800	1.98526	1000	1000	1	0	1	0	84.26	0.0	84.26	0.0	True
PBT_CG_S7	170.9925	0.73213	170.9925	0.73213	1000	1000	1	0	1	0	84.26	0.0	84.26	0.0	True
PBT_CG_S1	148.9700	10.62322	148.9700	10.62322	1000	1000	2	3	2	3	83.57	2.0	83.57	2.0	True
PBT_CG_S8	162.9400	5.28683	162.9400	5.28683	1000	1000	1	0	1	0	84.26	0.0	84.26	0.0	True
PBT_CG_S2	153.8450	8.23506	153.8450	8.23506	1000	1000	1	0	1	0	84.26	0.0	84.26	0.0	True
PBT_R_S6	317.8000	21.82578	317.8000	21.82578	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S2	326.0925	16.32428	326.0925	16.32428	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True

Table F.2: M0 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-1.1000	0.8878	-19.1683	16.9683	False
Random withPairWiseTesting	Random withoutPairWiseTesting	15.2225	0.0090	5.4024	25.0426	True
Random withConstantExtraction	Random withoutConstantExtraction	-1.6338	0.8382	-19.6617	16.3942	False
Coverage-Guided withBias	Coverage-Guided withoutBias	7.2781	0.5380	-19.6290	34.1852	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-21.8481	0.0216	-39.1835	-4.5128	True
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	6.2894	0.6029	-20.8653	33.4441	False
Coverage-Guided	Random	158.9534	0.0010	145.4685	172.4384	True

Table F.3: M0 - Tukey's test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.75	0.1682	-0.4214	1.9214	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.25	0.6844	-1.1179	1.6179	False
Random withConstantExtraction	Random withoutConstantExtraction	0.75	0.1682	-0.4214	1.9214	False
Coverage-Guided withBias	Coverage-Guided withoutBias	0.25	0.3559	-0.3617	0.8617	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	0.25	0.3559	-0.3617	0.8617	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	0.25	0.3559	-0.3617	0.8617	False
Coverage-Guided	Random	0.25	0.4051	-0.3747	0.8747	False

Table F.4: M0 - Tukey's test results concerning the *Coverage* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Coverage-Guided withBias	Coverage-Guided withoutBias	-0.635	0.1959	-1.7029	0.4329	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	0.290	0.5887	-0.9178	1.4978	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-0.635	0.1959	-1.7029	0.4329	False

F.1.1 Search Strategies

Table F.5: PBT_R_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1120.10000	35.07086	100	1000000	0	0.0	0.0	0.0
2	1244.92500	82.38642	100	1000000	0	0.0	0.0	0.0
3	1299.10000	53.02867	100	1000000	0	0.0	0.0	0.0
4	330.74500	9.63550	1000	3	2	0.0	0.0	100.0
5	8.63955	1.10066	1000	1000	0	0.0	0.0	0.0

Table F.6: PBT_R_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1097.0500	27.61137	100	1000000	0	0.0	0.0	0.0
2	1255.4500	57.14300	100	1000000	0	0.0	0.0	0.0
3	1331.6250	16.93390	100	1000000	0	0.0	0.0	0.0
4	326.0925	16.32428	1000	1	0	0.0	0.0	100.0
5	9.4629	0.79609	1000	1000	0	0.0	0.0	0.0

Table F.7: PBT_R_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1123.37500	6.23393	100	1000000	0	0.0	0.0	0.0
2	1246.90000	74.57422	100	1000000	0	0.0	0.0	0.0
3	1288.02500	12.57823	100	1000000	0	0.0	0.0	0.0
4	335.84250	8.57494	1000	1	0	0.0	0.0	100.0
5	9.78845	0.12160	1000	1000	0	0.0	0.0	0.0

Table F.8: PBT_R_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2285.05000	5.67825	1000	1000000	0	0.0	0.0	0.0
2	2204.20000	32.13464	1000	1000000	0	0.0	0.0	0.0
3	2193.55000	76.59355	1000	1000000	0	0.0	0.0	0.0
4	308.47750	17.57305	1000	2	1	0.0	0.0	100.0
5	7.14285	1.53223	1000	1000	0	0.0	0.0	0.0

Table F.9: PBT_R_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1116.40000	9.54751	100	1000000	0	0.0	0.0	0.0
2	1248.57500	69.22393	100	1000000	0	0.0	0.0	0.0
3	1291.67500	5.30678	100	1000000	0	0.0	0.0	0.0
4	324.14750	18.28314	1000	1	0	0.0	0.0	100.0
5	9.62082	0.19972	1000	1000	0	0.0	0.0	0.0

Table F.10: PBT_R_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2286.62500	8.62101	1000	1000000	0	0.0	0.0	0.0
2	2218.85000	10.07236	1000	1000000	0	0.0	0.0	0.0
3	2272.05000	21.62736	1000	1000000	0	0.0	0.0	0.0
4	317.80000	21.82578	1000	1	0	0.0	0.0	100.0
5	8.10845	2.77237	1000	1000	0	0.0	0.0	0.0

Table F.11: PBT_R_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2293.4500	13.08864	1000	1000000	0	0.0	0.0	0.0
2	2254.5250	30.69718	1000	1000000	0	0.0	0.0	0.0
3	2319.2750	5.80318	1000	1000000	0	0.0	0.0	0.0
4	309.1175	21.80078	1000	1	0	0.0	0.0	100.0
5	9.4349	0.64407	1000	1000	0	0.0	0.0	0.0

Table F.12: PBT_R_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2259.12500	39.39266	1000	1000000	0	0.0	0.0	0.0
2	2287.97500	98.52189	1000	1000000	0	0.0	0.0	0.0
3	2327.47500	105.17420	1000	1000000	0	0.0	0.0	0.0
4	320.54250	21.79394	1000	1	0	0.0	0.0	100.0
5	6.81695	2.19733	1000	1000	0	0.0	0.0	0.0

Table F.13: PBT_CG_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14672.75	144.96788	100	1000000	0	49.86	0.0	0.0
2	15768.00	196.34536	100	1000000	0	55.11	0.0	0.0
3	16507.75	257.15596	100	1000000	0	74.53	0.0	0.0
4	148.97	10.62322	1000	2	3	83.57	2.0	100.0
5	146.85	13.81834	1000	1000	0	85.42	0.0	0.0

Table F.14: PBT_CG_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14663.750	155.61391	100	1000000	0	49.86	0.0	0.0
2	15932.000	271.85474	100	1000000	0	55.11	0.0	0.0
3	16690.000	297.77760	100	1000000	0	74.53	0.0	0.0
4	153.845	8.23506	1000	1	0	84.26	0.0	100.0
5	154.560	4.57549	1000	1000	0	85.42	0.0	0.0

Table F.15: PBT_CG_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14671.5000	112.27310	100	1000000	0	49.86	0.0	0.0
2	15839.0000	105.43956	100	1000000	0	55.11	0.0	0.0
3	16518.0000	195.84305	100	1000000	0	74.53	0.0	0.0
4	151.1775	7.37422	1000	1	0	84.26	0.0	100.0
5	148.0175	5.64053	1000	1000	0	85.42	0.0	0.0

Table F.16: PBT_CG_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	15971.5000	95.89187	1000	1000000	0	49.86	0.0	0.0
2	16670.5000	124.42970	1000	1000000	0	55.11	0.0	0.0
3	17669.0000	189.12033	1000	1000000	0	74.53	0.0	0.0
4	193.9850	11.81765	1000	1	0	82.41	0.0	100.0
5	156.4425	25.78647	1000	1000	0	85.42	0.0	0.0

Table F.17: PBT_CG_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14841.0000	135.86574	100	1000000	0	49.86	0.0	0.0
2	15859.7500	201.01042	100	1000000	0	55.11	0.0	0.0
3	16244.2500	68.60166	100	1000000	0	74.53	0.0	0.0
4	152.8800	1.98526	1000	1	0	84.26	0.0	100.0
5	144.5425	9.32637	1000	1000	0	85.42	0.0	0.0

Table F.18: PBT_CG_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	16419.0000	146.62367	1000	1000000	0	49.86	0.0	0.0
2	16895.5000	88.03834	1000	1000000	0	55.11	0.0	0.0
3	17527.2500	436.01340	1000	1000000	0	74.53	0.0	0.0
4	166.3475	10.19603	1000	1	0	84.26	0.0	100.0
5	153.0150	8.90410	1000	1000	0	85.42	0.0	0.0

Table F.19: PBT_CG_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	16029.0000	185.26063	1000	1000000	0	49.86	0.0	0.0
2	16842.0000	218.45823	1000	1000000	0	55.11	0.0	0.0
3	17414.0000	236.24352	1000	1000000	0	74.53	0.0	0.0
4	170.9925	0.73213	1000	1	0	84.26	0.0	100.0
5	158.9150	18.01413	1000	1000	0	85.42	0.0	0.0

Table F.20: PBT_CG_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	16359.75	331.27434	1000	1000000	0	49.86	0.0	0.0
2	16876.00	146.23782	1000	1000000	0	55.11	0.0	0.0
3	17578.75	154.62111	1000	1000000	0	74.53	0.0	0.0
4	162.94	5.28683	1000	1	0	84.26	0.0	100.0
5	147.79	3.59615	1000	1000	0	85.42	0.0	0.0

F.2 M1

Table F.21: M1 - Search strategies results.

Strategy	Min Mean Time (ms)	Std Min Mean Time (ms)	Max Mean Time (ms)	Std Max Mean Time (ms)	Min Sample Size	Max Sample Size	Min Mean Test Cases	Std Min Mean Test Cases	Max Mean Test Cases	Std Max Mean Test Cases	Min Mean Coverage (%)	Std Min Mean Coverage (%)	Max Mean Coverage (%)	Std Max Mean Coverage (%)	Bug Found
PBT_R_S4	333.8300	19.64739	1508.4500	2.73176	1000	1000	1	0	1502	1117	0.00	0.0	0.00	0.0	True
PBT_R_S1	340.5225	10.87533	542.0475	11.15218	100	1000	1	0	1504	1660	0.00	0.0	0.00	0.0	True
PBT_R_S8	279.6800	42.91711	1568.4750	29.16551	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_CG_S4	162.5975	13.39238	249.6075	9.88149	1000	1000	1	0	67	106	61.65	0.0	80.64	5.8	True
PBT_CG_S6	151.4550	3.83727	240.8475	5.14978	1000	1000	1	0	1	0	61.65	0.0	83.63	0.0	True
PBT_CG_S3	140.2800	13.23004	156.4675	15.11094	100	1000	1	0	2	0	61.65	0.0	83.63	0.0	True
PBT_R_S3	318.0475	17.74237	562.5275	41.53604	100	1000	1	0	2	0	0.00	0.0	0.00	0.0	True
PBT_R_S5	326.7175	20.44150	523.7600	16.16801	100	1000	1	0	2	0	0.00	0.0	0.00	0.0	True
PBT_R_S7	285.9300	37.62141	1598.7750	52.17286	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_CG_S5	153.2975	15.77681	166.8700	11.66206	100	1000	1	0	2	0	61.65	0.0	83.63	0.0	True
PBT_CG_S7	159.2900	0.81802	247.7050	7.38008	1000	1000	1	0	1	0	61.65	0.0	83.63	0.0	True
PBT_CG_S1	146.9875	15.72440	161.4250	15.19867	100	1000	1	0	2	1	61.65	0.0	73.94	5.8	True
PBT_CG_S8	167.7125	13.88307	253.7225	15.50911	1000	1000	1	0	1	0	61.65	0.0	83.63	0.0	True
PBT_CG_S2	138.3500	8.26712	153.0300	2.92866	100	1000	1	0	2	0	61.65	0.0	83.63	0.0	True
PBT_R_S6	297.9700	34.95886	1567.4500	68.91243	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S2	358.2325	13.22952	520.3475	9.79853	100	1000	1	0	2	0	0.00	0.0	0.00	0.0	True

Table F.22: M1 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	3.9325	0.8620	-47.5812	55.4462	False
Random withPairWiseTesting	Random withoutPairWiseTesting	36.5275	0.0502	-0.0462	73.1012	False
Random withConstantExtraction	Random withoutConstantExtraction	30.0450	0.1310	-12.0060	72.0960	False
Coverage-Guided withBias	Coverage-Guided withoutBias	-0.4150	0.9000	-19.8984	19.0684	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-15.5350	0.0180	-27.3225	-3.7475	True
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-5.2975	0.5180	-24.0530	13.4580	False
Coverage-Guided	Random	165.1200	0.0010	142.7157	187.5243	True

F.2.1 Search Strategies

Table F.23: PBT_R_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1145.02500	17.54827	100	1000000	0	0.0	0.0	0.0
2	542.04750	11.15218	100	14	20	0.0	0.0	100.0
3	517.89750	4.29399	100	1	0	0.0	0.0	100.0
4	340.52250	10.87533	1000	1504	1660	0.0	0.0	100.0
5	9.31677	0.31220	1000	1000	0	0.0	0.0	0.0

Table F.24: PBT_R_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1149.17500	12.44134	100	1000000	0	0.0	0.0	0.0
2	489.62500	13.65812	100	2	0	0.0	0.0	100.0
3	520.34750	9.79853	100	1	0	0.0	0.0	100.0
4	358.23250	13.22952	1000	1	0	0.0	0.0	100.0
5	9.20155	0.23254	1000	1000	0	0.0	0.0	0.0

Table F.25: PBT_R_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1187.55000	76.77013	100	1000000	0	0.0	0.0	0.0
2	519.20250	47.57777	100	2	0	0.0	0.0	100.0
3	562.52750	41.53604	100	1	0	0.0	0.0	100.0
4	318.04750	17.74237	1000	1	0	0.0	0.0	100.0
5	9.72283	0.20055	1000	1000	0	0.0	0.0	0.0

Table F.26: PBT_R_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2300.85000	11.00738	1000	1000000	0	0.0	0.0	0.0
2	1507.02500	4.94589	1000	1	0	0.0	0.0	100.0
3	1508.45000	2.73176	1000	1	0	0.0	0.0	100.0
4	333.83000	19.64739	1000	1502	1117	0.0	0.0	100.0
5	7.46338	0.95331	1000	1000	0	0.0	0.0	0.0

Table F.27: PBT_R_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1136.27500	9.71838	100	1000000	0	0.0	0.0	0.0
2	488.02000	30.99536	100	2	0	0.0	0.0	100.0
3	523.76000	16.16801	100	1	0	0.0	0.0	100.0
4	326.71750	20.44150	1000	1	0	0.0	0.0	100.0
5	9.61128	0.23229	1000	1000	0	0.0	0.0	0.0

Table F.28: PBT_R_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2260.00000	25.57469	1000	1000000	0	0.0	0.0	0.0
2	1567.45000	68.91243	1000	1	0	0.0	0.0	100.0
3	1547.07500	27.85295	1000	1	0	0.0	0.0	100.0
4	297.97000	34.95886	1000	1	0	0.0	0.0	100.0
5	8.60765	0.72181	1000	1000	0	0.0	0.0	0.0

Table F.29: PBT_R_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2259.27500	43.29355	1000	1000000	0	0.0	0.0	0.0
2	1543.85000	57.92519	1000	1	0	0.0	0.0	100.0
3	1598.77500	52.17286	1000	1	0	0.0	0.0	100.0
4	285.93000	37.62141	1000	1	0	0.0	0.0	100.0
5	9.20383	0.62330	1000	1000	0	0.0	0.0	0.0

Table F.30: PBT_R_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2262.4000	50.73091	1000	1000000	0	0.0	0.0	0.0
2	1552.9000	56.98934	1000	1	0	0.0	0.0	100.0
3	1568.4750	29.16551	1000	1	0	0.0	0.0	100.0
4	279.6800	42.91711	1000	1	0	0.0	0.0	100.0
5	8.3366	2.28172	1000	1000	0	0.0	0.0	0.0

Table F.31: PBT_CG_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14574.5000	73.36723	100	1000000	0	50.87	0.00	0.0
2	152.5575	10.84306	100	2	1	64.95	1.44	100.0
3	146.9875	15.72440	100	1	0	61.65	0.00	100.0
4	161.4250	15.19867	1000	1	0	73.94	5.80	100.0
5	155.0750	2.99850	1000	1000	0	84.53	0.00	0.0

Table F.32: PBT_CG_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14766.25	99.77318	100	1000000	0	50.87	0.0	0.0
2	153.03	2.92866	100	2	0	74.17	0.0	100.0
3	138.35	8.26712	100	1	0	61.65	0.0	100.0
4	150.71	8.51888	1000	1	0	83.63	0.0	100.0
5	145.90	9.76962	1000	1000	0	84.53	0.0	0.0

Table F.33: PBT_CG_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14713.7500	64.16531	100	1000000	0	50.87	0.0	0.0
2	147.2600	8.64716	100	2	0	74.17	0.0	100.0
3	140.2800	13.23004	100	1	0	61.65	0.0	100.0
4	156.4675	15.11094	1000	1	0	83.63	0.0	100.0
5	159.4050	24.31394	1000	1000	0	84.53	0.0	0.0

Table F.34: PBT_CG_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	16121.7500	396.33721	1000	1000000	0	50.87	0.00	0.0
2	249.6075	9.88149	1000	3	2	65.91	1.07	100.0
3	231.1900	7.83372	1000	1	0	61.65	0.00	100.0
4	162.5975	13.39238	1000	67	106	80.64	5.80	100.0
5	152.9525	13.76007	1000	1000	0	84.53	0.00	0.0

Table F.35: PBT_CG_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	15035.0000	448.95601	100	1000000	0	50.87	0.0	0.0
2	161.1750	21.87883	100	2	0	74.17	0.0	100.0
3	153.2975	15.77681	100	1	0	61.65	0.0	100.0
4	166.8700	11.66206	1000	1	0	83.63	0.0	100.0
5	157.7700	14.64322	1000	1000	0	84.53	0.0	0.0

Table F.36: PBT_CG_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	15963.7500	165.73982	1000	1000000	0	50.87	0.0	0.0
2	240.8475	5.14978	1000	1	0	71.28	0.0	100.0
3	231.6625	19.67431	1000	1	0	61.65	0.0	100.0
4	151.4550	3.83727	1000	1	0	83.63	0.0	100.0
5	161.4325	12.76561	1000	1000	0	84.53	0.0	0.0

Table F.37: PBT_CG_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	16045.7500	131.77514	1000	1000000	0	50.87	0.0	0.0
2	247.7050	7.38008	1000	1	0	71.28	0.0	100.0
3	237.5350	13.02743	1000	1	0	61.65	0.0	100.0
4	159.2900	0.81802	1000	1	0	83.63	0.0	100.0
5	150.2425	2.68288	1000	1000	0	84.53	0.0	0.0

Table F.38: PBT_CG_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	16060.0000	86.61697	1000	1000000	0	50.87	0.0	0.0
2	253.7225	15.50911	1000	1	0	71.28	0.0	100.0
3	241.4925	17.64517	1000	1	0	61.65	0.0	100.0
4	167.7125	13.88307	1000	1	0	83.63	0.0	100.0
5	149.6150	15.39536	1000	1000	0	84.53	0.0	0.0

F.3 M2

Table F.39: M2 - Search strategies results.

Strategy	Min Mean Time (ms)	Std Min Mean Time (ms)	Max Mean Time (ms)	Std Max Mean Time (ms)	Min Sample Size	Max Sample Size	Min Mean Test Cases	Std Min Mean Test Cases	Max Mean Test Cases	Std Max Mean Test Cases	Min Mean Coverage (%)	Std Min Mean Coverage (%)	Max Mean Coverage (%)	Std Max Mean Coverage (%)	Bug Found
PBT_R_S4	5.26522	2.16540	1603.7000	119.93605	1000	1000	1	0	2	1	0.00	0.0	0.00	0.00	True
PBT_R_S1	5.90392	3.11519	510.6975	23.29071	100	1000	1	0	32	40	0.00	0.0	0.00	0.00	True
PBT_R_S8	4.66490	0.12083	1501.3750	45.99948	1000	1000	1	0	2020	19	0.00	0.0	0.00	0.00	True
PBT_CG_S4	116.84500	7.55266	237.3975	10.26057	1000	1000	1	0	6	9	48.61	0.0	52.80	0.45	True
PBT_CG_S6	125.83750	5.79658	234.4525	11.15400	1000	1000	1	0	260	7	48.61	0.0	53.25	0.00	True
PBT_CG_S3	113.47725	13.69306	185.3375	7.32618	100	1000	1	0	1325	63	48.61	0.0	53.25	0.00	True
PBT_R_S3	4.79153	0.25978	512.0225	6.29637	100	1000	1	0	20500	717	0.00	0.0	0.00	0.00	True
PBT_R_S5	4.84397	0.10036	516.4900	26.70714	100	1000	1	0	10201	0	0.00	0.0	0.00	0.00	True
PBT_R_S7	4.65142	0.11791	1462.0750	25.54715	1000	1000	1	0	2184	181	0.00	0.0	0.00	0.00	True
PBT_CG_S5	111.20250	3.49808	159.4875	8.18281	100	1000	1	0	676	0	48.61	0.0	53.25	0.00	True
PBT_CG_S7	114.39250	6.83598	250.3650	12.71293	1000	1000	1	0	517	17	48.61	0.0	53.25	0.00	True
PBT_CG_S1	116.41750	5.59037	147.6275	11.68619	100	1000	1	0	364	357	48.61	0.0	53.25	0.00	True
PBT_CG_S8	120.54500	9.53953	234.0700	9.69642	1000	1000	1	0	505	0	48.61	0.0	53.25	0.00	True
PBT_CG_S2	111.18250	4.24162	162.6125	7.58307	100	1000	1	0	676	0	48.61	0.0	53.25	0.00	True
PBT_R_S6	4.36470	0.73698	1500.2250	78.80797	1000	1000	1	0	1007	4	0.00	0.0	0.00	0.00	True
PBT_R_S2	4.04133	0.11027	504.3275	5.04758	100	1000	1	0	10201	0	0.00	0.0	0.00	0.00	True

Table F.40: M2 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.6743	0.0893	-0.1407	1.4893	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.1586	0.7362	-0.8867	1.2040	False
Random withConstantExtraction	Random withoutConstantExtraction	0.1558	0.7407	-0.8899	1.2016	False
Coverage-Guided withBias	Coverage-Guided withoutBias	-1.9088	0.6391	-11.0030	7.1853	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-6.3351	0.0629	-13.1388	0.4687	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	2.6663	0.4913	-6.2357	11.5683	False
Coverage-Guided	Random	-111.4216	0.0010	-115.2162	-107.6270	True

F.3.1 Search Strategies

Table F.41: PBT_R_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1151.52500	6.20338	100	1000000	0	0.0	0.0	0.0
2	1344.42500	41.37580	100	1000000	0	0.0	0.0	0.0
3	510.69750	23.29071	100	32	40	0.0	0.0	100.0
4	325.82500	13.50087	1000	1	0	0.0	0.0	100.0
5	5.90392	3.11519	1000	1	0	0.0	0.0	100.0

Table F.42: PBT_R_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1126.50000	6.87314	100	1000000	0	0.0	0.0	0.0
2	1347.55000	9.37830	100	1000000	0	0.0	0.0	0.0
3	504.32750	5.04758	100	10201	0	0.0	0.0	100.0
4	342.45500	20.63989	1000	1	0	0.0	0.0	100.0
5	4.04133	0.11027	1000	1	0	0.0	0.0	100.0

Table F.43: PBT_R_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1127.67500	6.35861	100	1000000	0	0.0	0.0	0.0
2	1213.10000	43.66366	100	1000000	0	0.0	0.0	0.0
3	512.02250	6.29637	100	20500	717	0.0	0.0	100.0
4	330.15500	12.03274	1000	1	0	0.0	0.0	100.0
5	4.79153	0.25978	1000	1	0	0.0	0.0	100.0

Table F.44: PBT_R_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2312.85000	102.73925	1000	1000000	0	0.0	0.0	0.0
2	2296.65000	114.22702	1000	1000000	0	0.0	0.0	0.0
3	1603.70000	119.93605	1000	2	1	0.0	0.0	100.0
4	296.70750	17.34005	1000	1	0	0.0	0.0	100.0
5	5.26522	2.16540	1000	1	0	0.0	0.0	100.0

Table F.45: PBT_R_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1132.80000	35.59389	100	1000000	0	0.0	0.0	0.0
2	1312.07500	31.58230	100	1000000	0	0.0	0.0	0.0
3	516.49000	26.70714	100	10201	0	0.0	0.0	100.0
4	325.10750	18.36979	1000	1	0	0.0	0.0	100.0
5	4.84397	0.10036	1000	1	0	0.0	0.0	100.0

Table F.46: PBT_R_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2289.0750	17.24404	1000	1000000	0	0.0	0.0	0.0
2	2241.1500	36.12288	1000	1000000	0	0.0	0.0	0.0
3	1500.2250	78.80797	1000	1007	4	0.0	0.0	100.0
4	285.5900	23.54982	1000	1	0	0.0	0.0	100.0
5	4.3647	0.73698	1000	1	0	0.0	0.0	100.0

Table F.47: PBT_R_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2296.00000	1.60156	1000	1000000	0	0.0	0.0	0.0
2	2217.10000	17.47327	1000	1000000	0	0.0	0.0	0.0
3	1462.07500	25.54715	1000	2184	181	0.0	0.0	100.0
4	296.43750	21.59349	1000	1	0	0.0	0.0	100.0
5	4.65142	0.11791	1000	1	0	0.0	0.0	100.0

Table F.48: PBT_R_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2293.0000	5.66436	1000	1000000	0	0.0	0.0	0.0
2	2221.0500	5.50704	1000	1000000	0	0.0	0.0	0.0
3	1501.3750	45.99948	1000	2020	19	0.0	0.0	100.0
4	283.2750	21.70315	1000	1	0	0.0	0.0	100.0
5	4.6649	0.12083	1000	1	0	0.0	0.0	100.0

Table F.49: PBT_CG_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14521.2500	139.98103	100	1000000	0	48.61	0.00	0.0
2	15930.7500	209.60841	100	1000000	0	53.11	0.00	0.0
3	147.6275	11.68619	100	364	357	53.25	0.00	100.0
4	147.4075	2.75997	1000	4	5	52.12	0.47	100.0
5	116.4175	5.59037	1000	1	0	48.61	0.00	100.0

Table F.50: PBT_CG_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14498.5000	91.56009	100	1000000	0	48.61	0.0	0.0
2	15904.0000	94.94209	100	1000000	0	53.11	0.0	0.0
3	162.6125	7.58307	100	676	0	53.25	0.0	100.0
4	152.6825	9.05858	1000	1	0	51.66	0.0	100.0
5	111.1825	4.24162	1000	1	0	48.61	0.0	100.0

Table F.51: PBT_CG_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14574.00000	165.43730	100	1000000	0	48.61	0.0	0.0
2	15927.75000	363.93432	100	1000000	0	53.11	0.0	0.0
3	185.33750	7.32618	100	1325	63	53.25	0.0	100.0
4	157.97000	10.86238	1000	1	0	51.66	0.0	100.0
5	113.47725	13.69306	1000	1	0	48.61	0.0	100.0

Table F.52: PBT_CG_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	15897.7500	178.56704	1000	1000000	0	48.61	0.00	0.0
2	16632.2500	226.21823	1000	1000000	0	53.11	0.00	0.0
3	237.3975	10.26057	1000	6	9	52.80	0.45	100.0
4	155.1950	13.78829	1000	1	0	51.66	0.00	100.0
5	116.8450	7.55266	1000	1	0	48.61	0.00	100.0

Table F.53: PBT_CG_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14844.7500	200.67184	100	1000000	0	48.61	0.0	0.0
2	15773.2500	270.20582	100	1000000	0	53.11	0.0	0.0
3	159.4875	8.18281	100	676	0	53.25	0.0	100.0
4	142.0525	2.70297	1000	1	0	51.66	0.0	100.0
5	111.2025	3.49808	1000	1	0	48.61	0.0	100.0

Table F.54: PBT_CG_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	16058.5000	123.00508	1000	1000000	0	48.61	0.0	0.0
2	16817.0000	67.27927	1000	1000000	0	53.11	0.0	0.0
3	234.4525	11.15400	1000	260	7	53.25	0.0	100.0
4	161.9525	9.62532	1000	1	0	51.66	0.0	100.0
5	125.8375	5.79658	1000	1	0	48.61	0.0	100.0

Table F.55: PBT_CG_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	16077.5000	74.32530	1000	1000000	0	48.61	0.0	0.0
2	16685.7500	47.35174	1000	1000000	0	53.11	0.0	0.0
3	250.3650	12.71293	1000	517	17	53.25	0.0	100.0
4	153.5100	4.71367	1000	1	0	51.66	0.0	100.0
5	114.3925	6.83598	1000	1	0	48.61	0.0	100.0

Table F.56: PBT_CG_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	16254.2500	122.57115	1000	1000000	0	48.61	0.0	0.0
2	16705.5000	85.19830	1000	1000000	0	53.11	0.0	0.0
3	234.0700	9.69642	1000	505	0	53.25	0.0	100.0
4	163.7275	13.61882	1000	1	0	51.66	0.0	100.0
5	120.5450	9.53953	1000	1	0	48.61	0.0	100.0

F.4 M3

Table F.57: M3 - Search strategies results.

Strategy	Min Mean Time (ms)	Std Min Mean Time (ms)	Max Mean Time (ms)	Std Max Mean Time (ms)	Min Sample Size	Max Sample Size	Min Mean Test Cases	Std Min Mean Test Cases	Max Mean Test Cases	Std Max Mean Test Cases	Min Mean Coverage (%)	Std Min Mean Coverage (%)	Max Mean Coverage (%)	Std Max Mean Coverage (%)	Bug Found
PBT_R_S4	294.0450	36.30036	294.0450	36.30036	1000	1000	2	1	2	1	0.00	0.00	0.00	0.00	True
PBT_R_S1	335.3450	14.20074	335.3450	14.20074	1000	1000	3	2	3	2	0.00	0.00	0.00	0.00	True
PBT_R_S8	287.8450	24.84434	287.8450	24.84434	1000	1000	1	0	1	0	0.00	0.00	0.00	0.00	True
PBT_CG_S4	159.6225	7.79092	159.6225	7.79092	1000	1000	1	0	1	0	83.44	1.97	83.44	1.97	True
PBT_CG_S6	179.5725	19.74717	179.5725	19.74717	1000	1000	1	0	1	0	84.53	0.00	84.53	0.00	True
PBT_CG_S3	159.2275	5.47547	159.2275	5.47547	1000	1000	1	0	1	0	84.53	0.00	84.53	0.00	True
PBT_R_S3	333.9850	3.31835	333.9850	3.31835	1000	1000	1	0	1	0	0.00	0.00	0.00	0.00	True
PBT_R_S5	319.8100	20.61602	319.8100	20.61602	1000	1000	1	0	1	0	0.00	0.00	0.00	0.00	True
PBT_R_S7	290.9550	29.10620	290.9550	29.10620	1000	1000	1	0	1	0	0.00	0.00	0.00	0.00	True
PBT_CG_S5	153.0450	5.70725	153.0450	5.70725	1000	1000	1	0	1	0	84.53	0.00	84.53	0.00	True
PBT_CG_S7	185.4225	23.19865	185.4225	23.19865	1000	1000	1	0	1	0	84.53	0.00	84.53	0.00	True
PBT_CG_S1	145.1675	4.65111	145.1675	4.65111	1000	1000	2	2	2	2	84.58	2.28	84.58	2.28	True
PBT_CG_S8	164.5350	10.04052	164.5350	10.04052	1000	1000	1	0	1	0	84.53	0.00	84.53	0.00	True
PBT_CG_S2	156.4850	7.22733	156.4850	7.22733	1000	1000	1	0	1	0	84.53	0.00	84.53	0.00	True
PBT_R_S6	310.5850	29.27298	310.5850	29.27298	1000	1000	1	0	1	0	0.00	0.00	0.00	0.00	True
PBT_R_S2	348.9600	15.92017	348.9600	15.92017	1000	1000	1	0	1	0	0.00	0.00	0.00	0.00	True

Table F.58: M3 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-3.2175	0.8645	-46.2067	39.7717	False
Random withPairWiseTesting	Random withoutPairWiseTesting	38.6675	0.0026	19.5267	57.8083	True
Random withConstantExtraction	Random withoutConstantExtraction	14.0850	0.4301	-26.6634	54.8334	False
Coverage-Guided withBias	Coverage-Guided withoutBias	-1.0494	0.9000	-26.1491	24.0503	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-18.8069	0.0329	-35.4844	-2.1293	True
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-5.3456	0.6260	-29.8931	19.2019	False
Coverage-Guided	Random	152.3066	0.0010	132.0616	172.5516	True

Table F.59: M3 - Tukey's test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.75	0.1682	-0.4214	1.9214	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.25	0.6844	-1.1179	1.6179	False
Random withConstantExtraction	Random withoutConstantExtraction	0.75	0.1682	-0.4214	1.9214	False
Coverage-Guided withBias	Coverage-Guided withoutBias	0.25	0.3559	-0.3617	0.8617	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	0.25	0.3559	-0.3617	0.8617	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	0.25	0.3559	-0.3617	0.8617	False
Coverage-Guided	Random	0.25	0.4051	-0.3747	0.8747	False

Table F.60: M3 - Tukey's test results concerning the *Coverage* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Coverage-Guided withBias	Coverage-Guided withoutBias	-0.260	0.3840	-0.9376	0.4176	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	0.285	0.3364	-0.3825	0.9525	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-0.260	0.3840	-0.9376	0.4176	False

F.4.1 Search Strategies

Table F.61: PBT_R_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1143.55000	6.70242	100	1000000	0	0.0	0.0	0.0
2	1309.25000	73.28770	100	1000000	0	0.0	0.0	0.0
3	1357.22500	52.91051	100	1000000	0	0.0	0.0	0.0
4	335.34500	14.20074	1000	3	2	0.0	0.0	100.0
5	9.26548	0.36195	1000	1000	0	0.0	0.0	0.0

Table F.62: PBT_R_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1128.82500	16.57051	100	1000000	0	0.0	0.0	0.0
2	1296.82500	47.92903	100	1000000	0	0.0	0.0	0.0
3	1419.67500	113.16513	100	1000000	0	0.0	0.0	0.0
4	348.96000	15.92017	1000	1	0	0.0	0.0	100.0
5	8.97445	0.35134	1000	1000	0	0.0	0.0	0.0

Table F.63: PBT_R_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1128.95000	9.74846	100	1000000	0	0.0	0.0	0.0
2	1277.80000	55.28286	100	1000000	0	0.0	0.0	0.0
3	1299.25000	17.01242	100	1000000	0	0.0	0.0	0.0
4	333.98500	3.31835	1000	1	0	0.0	0.0	100.0
5	10.81223	1.46511	1000	1000	0	0.0	0.0	0.0

Table F.64: PBT_R_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2219.950	35.26659	1000	1000000	0	0.0	0.0	0.0
2	2278.600	82.83275	1000	1000000	0	0.0	0.0	0.0
3	2361.750	95.62166	1000	1000000	0	0.0	0.0	0.0
4	294.045	36.30036	1000	2	1	0.0	0.0	100.0
5	7.950	2.01169	1000	1000	0	0.0	0.0	0.0

Table F.65: PBT_R_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1125.47500	3.78971	100	1000000	0	0.0	0.0	0.0
2	1210.72500	55.90480	100	1000000	0	0.0	0.0	0.0
3	1288.22500	91.22797	100	1000000	0	0.0	0.0	0.0
4	319.81000	20.61602	1000	1	0	0.0	0.0	100.0
5	9.89305	0.38518	1000	1000	0	0.0	0.0	0.0

Table F.66: PBT_R_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2291.30000	6.50000	1000	1000000	0	0.0	0.0	0.0
2	2205.92500	36.16534	1000	1000000	0	0.0	0.0	0.0
3	2254.05000	36.34835	1000	1000000	0	0.0	0.0	0.0
4	310.58500	29.27298	1000	1	0	0.0	0.0	100.0
5	8.22518	2.54343	1000	1000	0	0.0	0.0	0.0

Table F.67: PBT_R_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2261.525	48.52156	1000	1000000	0	0.0	0.0	0.0
2	2268.300	84.61746	1000	1000000	0	0.0	0.0	0.0
3	2293.275	150.23336	1000	1000000	0	0.0	0.0	0.0
4	290.955	29.10620	1000	1	0	0.0	0.0	100.0
5	8.405	2.29233	1000	1000	0	0.0	0.0	0.0

Table F.68: PBT_R_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2247.17500	31.92588	1000	1000000	0	0.0	0.0	0.0
2	2355.60000	134.74635	1000	1000000	0	0.0	0.0	0.0
3	2343.27500	98.62597	1000	1000000	0	0.0	0.0	0.0
4	287.84500	24.84434	1000	1	0	0.0	0.0	100.0
5	9.51258	0.65697	1000	1000	0	0.0	0.0	0.0

Table F.69: PBT_CG_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14689.5000	107.92938	100	1000000	0	49.68	0.00	0.0
2	15831.7500	168.36326	100	1000000	0	54.77	0.00	0.0
3	16447.5000	56.53539	100	1000000	0	73.91	0.00	0.0
4	145.1675	4.65111	1000	2	2	84.58	2.28	100.0
5	151.5975	8.14984	1000	1000	0	85.72	0.00	0.0

Table F.70: PBT_CG_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14882.750	275.05670	100	1000000	0	49.68	0.0	0.0
2	16039.250	155.49819	100	1000000	0	54.77	0.0	0.0
3	16482.750	167.19655	100	1000000	0	73.91	0.0	0.0
4	156.485	7.22733	1000	1	0	84.53	0.0	100.0
5	144.490	10.02071	1000	1000	0	85.72	0.0	0.0

Table F.71: PBT_CG_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14612.2500	148.45264	100	1000000	0	49.68	0.0	0.0
2	16165.2500	191.22941	100	1000000	0	54.77	0.0	0.0
3	16428.2500	185.08697	100	1000000	0	73.91	0.0	0.0
4	159.2275	5.47547	1000	1	0	84.53	0.0	100.0
5	154.9100	8.02987	1000	1000	0	85.72	0.0	0.0

Table F.72: PBT_CG_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	16304.0000	176.77670	1000	1000000	0	49.68	0.00	0.0
2	16648.0000	110.55089	1000	1000000	0	54.77	0.00	0.0
3	17320.2500	297.91221	1000	1000000	0	73.91	0.00	0.0
4	159.6225	7.79092	1000	1	0	83.44	1.97	100.0
5	153.1625	7.49214	1000	1000	0	85.72	0.00	0.0

Table F.73: PBT_CG_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14929.000	166.03463	100	1000000	0	49.68	0.0	0.0
2	15973.250	198.67231	100	1000000	0	54.77	0.0	0.0
3	16582.250	141.81568	100	1000000	0	73.91	0.0	0.0
4	153.045	5.70725	1000	1	0	84.53	0.0	100.0
5	149.530	6.73983	1000	1000	0	85.72	0.0	0.0

Table F.74: PBT_CG_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	16220.5000	124.23466	1000	1000000	0	49.68	0.0	0.0
2	16797.5000	175.57691	1000	1000000	0	54.77	0.0	0.0
3	17727.0000	413.83994	1000	1000000	0	73.91	0.0	0.0
4	179.5725	19.74717	1000	1	0	84.53	0.0	100.0
5	161.2825	14.61928	1000	1000	0	85.72	0.0	0.0

Table F.75: PBT_CG_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	16207.7500	185.09102	1000	1000000	0	49.68	0.0	0.0
2	16781.0000	257.10698	1000	1000000	0	54.77	0.0	0.0
3	17651.2500	263.40309	1000	1000000	0	73.91	0.0	0.0
4	185.4225	23.19865	1000	1	0	84.53	0.0	100.0
5	169.7925	24.09489	1000	1000	0	85.72	0.0	0.0

Table F.76: PBT_CG_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	16342.500	76.36917	1000	1000000	0	49.68	0.0	0.0
2	16703.000	172.79612	1000	1000000	0	54.77	0.0	0.0
3	17394.750	247.67052	1000	1000000	0	73.91	0.0	0.0
4	164.535	10.04052	1000	1	0	84.53	0.0	100.0
5	175.285	17.56219	1000	1000	0	85.72	0.0	0.0

F.5 M4

Table F.77: M4 - Search strategies results.

Strategy	Min Mean Time (ms)	Std Min Mean Time (ms)	Max Mean Time (ms)	Std Max Mean Time (ms)	Min Sample Size	Max Sample Size	Min Mean Test Cases	Std Min Mean Test Cases	Max Mean Test Cases	Std Max Mean Test Cases	Min Mean Coverage (%)	Std Min Mean Coverage (%)	Max Mean Coverage (%)	Std Max Mean Coverage (%)	Bug Found
PBT_R_S4	277.9825	11.05383	277.9825	11.05383	1000	1000	6	7	6	7	0.00	0.0	0.00	0.0	True
PBT_R_S1	329.8850	19.28455	329.8850	19.28455	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S8	313.3200	27.52725	313.3200	27.52725	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_CG_S4	161.5175	8.58700	161.5175	8.58700	1000	1000	1	0	1	0	89.09	0.0	89.09	0.0	True
PBT_CG_S6	159.1575	5.31148	159.1575	5.31148	1000	1000	1	0	1	0	83.63	0.0	83.63	0.0	True
PBT_CG_S3	153.2725	5.89437	153.2725	5.89437	1000	1000	1	0	1	0	83.63	0.0	83.63	0.0	True
PBT_R_S3	319.4775	18.75142	319.4775	18.75142	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S5	327.1925	5.49787	327.1925	5.49787	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S7	305.9525	22.74157	305.9525	22.74157	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_CG_S5	153.0000	7.03030	153.0000	7.03030	1000	1000	1	0	1	0	83.63	0.0	83.63	0.0	True
PBT_CG_S7	166.5025	11.87631	166.5025	11.87631	1000	1000	1	0	1	0	83.63	0.0	83.63	0.0	True
PBT_CG_S1	162.4025	14.81964	162.4025	14.81964	1000	1000	1	0	1	0	89.09	0.0	89.09	0.0	True
PBT_CG_S8	162.1550	9.15799	162.1550	9.15799	1000	1000	1	0	1	0	83.63	0.0	83.63	0.0	True
PBT_CG_S2	156.9550	13.23615	156.9550	13.23615	1000	1000	1	0	1	0	83.63	0.0	83.63	0.0	True
PBT_R_S6	309.8625	26.97564	309.8625	26.97564	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S2	320.3475	16.41860	320.3475	16.41860	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True

Table F.78: M4 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-9.3562	0.4609	-38.4223	19.7098	False
Random withPairWiseTesting	Random withoutPairWiseTesting	22.4462	0.0380	1.7239	43.1686	True
Random withConstantExtraction	Random withoutConstantExtraction	-6.9663	0.5983	-36.6944	22.7619	False
Coverage-Guided withBias	Coverage-Guided withoutBias	3.1069	0.3929	-5.1538	11.3675	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-5.9256	0.0686	-12.4702	0.6190	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	1.2756	0.7456	-7.4563	10.0076	False
Coverage-Guided	Random	153.6322	0.0010	140.7367	166.5276	True

Table F.79: M4 - Tukey's test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	1.250	0.3559	-1.8086	4.3086	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-1.250	0.3559	-4.3086	1.8086	False
Random withConstantExtraction	Random withoutConstantExtraction	1.250	0.3559	-1.8086	4.3086	False
Coverage-Guided	Random	0.625	0.3343	-0.7155	1.9655	False

Table F.80: M4 - Tukey's test results concerning the *Coverage* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Coverage-Guided withBias	Coverage-Guided withoutBias	2.73	0.134	-1.1267	6.5867	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	0.00	0.900	-4.7235	4.7235	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	2.73	0.134	-1.1267	6.5867	False

F.5.1 Search Strategies

Table F.81: PBT_R_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1142.55000	13.26226	100	1000000	0	0.0	0.0	0.0
2	1332.07500	76.99196	100	1000000	0	0.0	0.0	0.0
3	1352.32500	40.43157	100	1000000	0	0.0	0.0	0.0
4	329.88500	19.28455	1000	1	0	0.0	0.0	100.0
5	9.10298	0.09837	1000	1000	0	0.0	0.0	0.0

Table F.82: PBT_R_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1144.50000	31.86840	100	1000000	0	0.0	0.0	0.0
2	1200.15000	27.64584	100	1000000	0	0.0	0.0	0.0
3	1319.62500	27.79248	100	1000000	0	0.0	0.0	0.0
4	320.34750	16.41860	1000	1	0	0.0	0.0	100.0
5	9.57485	0.97561	1000	1000	0	0.0	0.0	0.0

Table F.83: PBT_R_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1121.7750	8.25087	100	1000000	0	0.0	0.0	0.0
2	1178.4000	19.10419	100	1000000	0	0.0	0.0	0.0
3	1314.6000	19.86492	100	1000000	0	0.0	0.0	0.0
4	319.4775	18.75142	1000	1	0	0.0	0.0	100.0
5	9.8700	0.10430	1000	1000	0	0.0	0.0	0.0

Table F.84: PBT_R_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2251.10000	47.59333	1000	1000000	0	0.0	0.0	0.0
2	2241.17500	41.78872	1000	1000000	0	0.0	0.0	0.0
3	2291.10000	65.64945	1000	1000000	0	0.0	0.0	0.0
4	277.98250	11.05383	1000	6	7	0.0	0.0	100.0
5	8.85582	0.31657	1000	1000	0	0.0	0.0	0.0

Table F.85: PBT_R_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	1123.95000	11.59084	100	1000000	0	0.0	0.0	0.0
2	1203.15000	32.41686	100	1000000	0	0.0	0.0	0.0
3	1301.72500	15.40574	100	1000000	0	0.0	0.0	0.0
4	327.19250	5.49787	1000	1	0	0.0	0.0	100.0
5	9.74715	0.25779	1000	1000	0	0.0	0.0	0.0

Table F.86: PBT_R_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2280.05000	9.47958	1000	1000000	0	0.0	0.0	0.0
2	2191.60000	19.10654	1000	1000000	0	0.0	0.0	0.0
3	2228.57500	60.40929	1000	1000000	0	0.0	0.0	0.0
4	309.86250	26.97564	1000	1	0	0.0	0.0	100.0
5	8.04107	2.36203	1000	1000	0	0.0	0.0	0.0

Table F.87: PBT_R_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2282.45000	7.95660	1000	1000000	0	0.0	0.0	0.0
2	2198.62500	20.97300	1000	1000000	0	0.0	0.0	0.0
3	2241.15000	73.49410	1000	1000000	0	0.0	0.0	0.0
4	305.95250	22.74157	1000	1	0	0.0	0.0	100.0
5	8.33418	2.06682	1000	1000	0	0.0	0.0	0.0

Table F.88: PBT_R_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	2270.2750	36.10079	1000	1000000	0	0.0	0.0	0.0
2	2253.3250	126.53279	1000	1000000	0	0.0	0.0	0.0
3	2295.1250	94.10046	1000	1000000	0	0.0	0.0	0.0
4	313.3200	27.52725	1000	1	0	0.0	0.0	100.0
5	8.3753	2.39385	1000	1000	0	0.0	0.0	0.0

Table F.89: PBT_CG_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14613.7500	118.14266	100	1000000	0	49.68	0.0	0.0
2	15783.7500	105.50444	100	1000000	0	54.77	0.0	0.0
3	16520.0000	220.67963	100	1000000	0	73.91	0.0	0.0
4	162.4025	14.81964	1000	1	0	89.09	0.0	100.0
5	157.8725	10.20183	1000	1000	0	84.53	0.0	0.0

Table F.90: PBT_CG_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14626.750	66.30752	100	1000000	0	49.68	0.0	0.0
2	15950.500	105.58054	100	1000000	0	54.77	0.0	0.0
3	16448.250	126.11577	100	1000000	0	73.91	0.0	0.0
4	156.955	13.23615	1000	1	0	83.63	0.0	100.0
5	158.890	13.84989	1000	1000	0	84.53	0.0	0.0

Table F.91: PBT_CG_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14722.7500	167.15319	100	1000000	0	49.68	0.0	0.0
2	15921.0000	158.48186	100	1000000	0	54.77	0.0	0.0
3	16368.2500	245.42756	100	1000000	0	73.91	0.0	0.0
4	153.2725	5.89437	1000	1	0	83.63	0.0	100.0
5	152.6550	16.11627	1000	1000	0	84.53	0.0	0.0

Table F.92: PBT_CG_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	16025.2500	76.58778	1000	1000000	0	49.68	0.0	0.0
2	16606.0000	134.81654	1000	1000000	0	54.77	0.0	0.0
3	17556.0000	68.72772	1000	1000000	0	73.91	0.0	0.0
4	161.5175	8.58700	1000	1	0	89.09	0.0	100.0
5	154.8775	11.70061	1000	1000	0	84.53	0.0	0.0

Table F.93: PBT_CG_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	14706.50	194.86470	100	1000000	0	49.68	0.0	0.0
2	15933.00	184.96892	100	1000000	0	54.77	0.0	0.0
3	16443.75	263.54542	100	1000000	0	73.91	0.0	0.0
4	153.00	7.03030	1000	1	0	83.63	0.0	100.0
5	140.88	7.11766	1000	1000	0	84.53	0.0	0.0

Table F.94: PBT_CG_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	16085.2500	90.07046	1000	1000000	0	49.68	0.0	0.0
2	16711.2500	366.08290	1000	1000000	0	54.77	0.0	0.0
3	17601.7500	80.86524	1000	1000000	0	73.91	0.0	0.0
4	159.1575	5.31148	1000	1	0	83.63	0.0	100.0
5	159.8050	1.36155	1000	1000	0	84.53	0.0	0.0

Table F.95: PBT_CG_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	16104.0000	151.31920	1000	1000000	0	49.68	0.0	0.0
2	16878.0000	158.93552	1000	1000000	0	54.77	0.0	0.0
3	17591.5000	76.22828	1000	1000000	0	73.91	0.0	0.0
4	166.5025	11.87631	1000	1	0	83.63	0.0	100.0
5	152.1175	7.11106	1000	1000	0	84.53	0.0	0.0

Table F.96: PBT_CG_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	16263.500	158.30430	1000	1000000	0	49.68	0.0	0.0
2	16885.500	145.45532	1000	1000000	0	54.77	0.0	0.0
3	17662.250	195.45380	1000	1000000	0	73.91	0.0	0.0
4	162.155	9.15799	1000	1	0	83.63	0.0	100.0
5	157.155	11.15805	1000	1000	0	84.53	0.0	0.0

Appendix G

Zodiac Results

This appendix presents all the data associated with the results achieved for all the `Zodiac` benchmark variants, including each search strategy's results and all the data extracted concerning all the evaluation metrics.

G.1 M0

Table G.1: M0 - Search strategies results.

Strategy	Min Mean Time (ms)	Std Min Mean Time (ms)	Max Mean Time (ms)	Std Max Mean Time (ms)	Min Sample Size	Max Sample Size	Min Mean Test Cases	Std Min Mean Test Cases	Max Mean Test Cases	Std Max Mean Test Cases	Min Mean Coverage (%)	Std Min Mean Coverage (%)	Max Mean Coverage (%)	Std Max Mean Coverage (%)	Bug Found
PBT_R_S4	0.23921	0.04406	0.23921	0.04406	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S1	0.24031	0.02277	0.24031	0.02277	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S8	5.44670	0.20531	5.44670	0.20531	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_CG_S4	167.83750	2.73985	167.83750	2.73985	1000	1000	1	0	1	0	73.63	0.0	73.63	0.0	True
PBT_CG_S6	167.76750	0.81806	167.76750	0.81806	1000	1000	1	0	1	0	73.63	0.0	73.63	0.0	True
PBT_CG_S3	187.40250	23.27639	187.40250	23.27639	1000	1000	1	0	1	0	73.63	0.0	73.63	0.0	True
PBT_R_S3	5.15090	0.41124	5.15090	0.41124	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S5	5.28940	0.47565	5.28940	0.47565	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S7	5.36170	0.20051	5.36170	0.20051	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_CG_S5	173.76000	2.07573	173.76000	2.07573	1000	1000	1	0	1	0	73.63	0.0	73.63	0.0	True
PBT_CG_S7	175.76750	3.07982	175.76750	3.07982	1000	1000	1	0	1	0	73.63	0.0	73.63	0.0	True
PBT_CG_S1	164.97000	4.65157	164.97000	4.65157	1000	1000	1	0	1	0	73.63	0.0	73.63	0.0	True
PBT_CG_S8	196.90750	21.01064	196.90750	21.01064	1000	1000	1	0	1	0	73.63	0.0	73.63	0.0	True
PBT_CG_S2	169.77500	3.35044	169.77500	3.35044	1000	1000	1	0	1	0	73.63	0.0	73.63	0.0	True
PBT_R_S6	0.26451	0.03136	0.26451	0.03136	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S2	0.26510	0.03168	0.26510	0.03168	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True

Table G.2: M0 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-0.0684	0.9000	-5.1249	4.9881	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-0.0916	0.9000	-5.1477	4.9645	False
Random withConstantExtraction	Random withoutConstantExtraction	-5.0599	0.0010	-5.2142	-4.9056	True
Coverage-Guided withBias	Coverage-Guided withoutBias	-3.0581	0.7411	-23.6096	17.4933	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-3.0931	0.7382	-23.6393	17.4531	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-15.8719	0.0276	-29.2997	-2.4440	True
Coverage-Guided	Random	-172.7412	0.0010	-181.4178	-164.0647	True

G.1.1 Search Strategies

Table G.3: PBT_R_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	58.74725	7.37721	1000	31000	0	0.0	0.0	0.0
2	30.73425	2.92691	1000	12000	0	0.0	0.0	0.0
3	4.15515	0.42856	1000	10	0	0.0	0.0	0.0
4	0.23155	0.02369	1000	19	0	0.0	0.0	0.0
5	0.21102	0.00549	1000	12	0	0.0	0.0	0.0
6	0.20788	0.02375	1000	17	0	0.0	0.0	0.0
7	0.18996	0.01529	1000	12	0	0.0	0.0	0.0
8	0.20053	0.01917	1000	19	0	0.0	0.0	0.0
9	0.18874	0.01515	1000	12	0	0.0	0.0	0.0
10	0.19900	0.02027	1000	19	0	0.0	0.0	0.0
11	0.18506	0.02096	1000	11	0	0.0	0.0	0.0
12	0.20305	0.02346	1000	20	0	0.0	0.0	0.0
13	0.18217	0.02128	1000	11	0	0.0	0.0	0.0
14	0.20286	0.01722	1000	20	0	0.0	0.0	0.0
15	0.17944	0.02071	1000	10	0	0.0	0.0	0.0
16	0.24031	0.02277	1000	1	0	0.0	0.0	100.0
17	0.18519	0.02024	1000	9	0	0.0	0.0	0.0
18	0.21127	0.02160	1000	22	0	0.0	0.0	0.0
19	0.18342	0.02001	1000	9	0	0.0	0.0	0.0
20	0.21106	0.02016	1000	22	0	0.0	0.0	0.0
21	0.17179	0.01784	1000	8	0	0.0	0.0	0.0
22	0.21296	0.02044	1000	22	0	0.0	0.0	0.0
23	0.17564	0.01736	1000	9	0	0.0	0.0	0.0
24	0.21074	0.01774	1000	21	0	0.0	0.0	0.0
25	0.25989	0.15898	1000	9	0	0.0	0.0	0.0
26	0.35947	0.14421	1000	21	0	0.0	0.0	0.0

Table G.4: PBT_R_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	60.17475	4.48784	1000	31000	0	0.0	0.0	0.0
2	32.48500	1.80339	1000	12000	0	0.0	0.0	0.0
3	4.23323	0.44356	1000	10	0	0.0	0.0	0.0
4	0.27451	0.03788	1000	19	0	0.0	0.0	0.0
5	0.21338	0.02650	1000	12	0	0.0	0.0	0.0
6	0.22557	0.01883	1000	17	0	0.0	0.0	0.0
7	0.20709	0.02416	1000	12	0	0.0	0.0	0.0
8	0.21543	0.02302	1000	19	0	0.0	0.0	0.0
9	0.20739	0.02693	1000	12	0	0.0	0.0	0.0
10	0.22152	0.02526	1000	19	0	0.0	0.0	0.0
11	0.20816	0.03142	1000	11	0	0.0	0.0	0.0
12	0.23281	0.03937	1000	20	0	0.0	0.0	0.0
13	0.20144	0.02900	1000	11	0	0.0	0.0	0.0
14	0.26803	0.09543	1000	20	0	0.0	0.0	0.0
15	0.21870	0.04144	1000	10	0	0.0	0.0	0.0
16	0.26510	0.03168	1000	1	0	0.0	0.0	100.0
17	0.24301	0.09204	1000	9	0	0.0	0.0	0.0
18	0.26076	0.07301	1000	22	0	0.0	0.0	0.0
19	0.21988	0.05908	1000	9	0	0.0	0.0	0.0
20	0.26545	0.07694	1000	22	0	0.0	0.0	0.0
21	0.21493	0.06268	1000	8	0	0.0	0.0	0.0
22	0.23890	0.04330	1000	22	0	0.0	0.0	0.0
23	0.21035	0.05618	1000	9	0	0.0	0.0	0.0
24	0.29568	0.09604	1000	21	0	0.0	0.0	0.0
25	0.21903	0.06952	1000	9	0	0.0	0.0	0.0
26	0.35554	0.23969	1000	21	0	0.0	0.0	0.0

Table G.5: PBT_R_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	77.18075	8.19511	1000	31000	0	0.0	0.0	0.0
2	51.50300	4.56983	1000	12000	0	0.0	0.0	0.0
3	12.86550	1.06403	1000	10	0	0.0	0.0	0.0
4	6.60058	0.56007	1000	19	0	0.0	0.0	0.0
5	5.33230	0.45151	1000	12	0	0.0	0.0	0.0
6	5.46232	0.36956	1000	17	0	0.0	0.0	0.0
7	9.58778	0.77602	1000	12	0	0.0	0.0	0.0
8	5.26245	0.42622	1000	19	0	0.0	0.0	0.0
9	5.23372	0.39046	1000	12	0	0.0	0.0	0.0
10	5.51875	0.43408	1000	19	0	0.0	0.0	0.0
11	5.19648	0.36796	1000	11	0	0.0	0.0	0.0
12	5.24558	0.37757	1000	20	0	0.0	0.0	0.0
13	5.20300	0.40978	1000	11	0	0.0	0.0	0.0
14	5.35403	0.37727	1000	20	0	0.0	0.0	0.0
15	5.13232	0.36980	1000	10	0	0.0	0.0	0.0
16	5.15090	0.41124	1000	1	0	0.0	0.0	100.0
17	5.05735	0.33131	1000	9	0	0.0	0.0	0.0
18	5.10545	0.37043	1000	22	0	0.0	0.0	0.0
19	5.02430	0.35605	1000	9	0	0.0	0.0	0.0
20	5.04095	0.37514	1000	22	0	0.0	0.0	0.0
21	5.07498	0.29342	1000	8	0	0.0	0.0	0.0
22	5.10623	0.29405	1000	22	0	0.0	0.0	0.0
23	5.09165	0.32266	1000	9	0	0.0	0.0	0.0
24	5.05200	0.34386	1000	21	0	0.0	0.0	0.0
25	5.32990	0.35648	1000	9	0	0.0	0.0	0.0
26	5.10688	0.35523	1000	21	0	0.0	0.0	0.0

Table G.6: PBT_R_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	56.74500	5.04685	1000	31000	0	0.0	0.0	0.0
2	29.31200	2.49676	1000	12000	0	0.0	0.0	0.0
3	4.06430	0.37160	1000	10	0	0.0	0.0	0.0
4	0.28174	0.07921	1000	19	0	0.0	0.0	0.0
5	0.27485	0.13169	1000	12	0	0.0	0.0	0.0
6	0.22324	0.06298	1000	17	0	0.0	0.0	0.0
7	0.21346	0.05743	1000	12	0	0.0	0.0	0.0
8	0.22357	0.05860	1000	19	0	0.0	0.0	0.0
9	0.21444	0.06258	1000	12	0	0.0	0.0	0.0
10	0.23513	0.05739	1000	19	0	0.0	0.0	0.0
11	0.21326	0.05978	1000	11	0	0.0	0.0	0.0
12	0.23618	0.06023	1000	20	0	0.0	0.0	0.0
13	0.19704	0.04630	1000	11	0	0.0	0.0	0.0
14	0.28322	0.05591	1000	20	0	0.0	0.0	0.0
15	0.19646	0.03736	1000	10	0	0.0	0.0	0.0
16	0.23921	0.04406	1000	1	0	0.0	0.0	100.0
17	0.19288	0.02648	1000	9	0	0.0	0.0	0.0
18	0.22295	0.01327	1000	22	0	0.0	0.0	0.0
19	0.19900	0.01171	1000	9	0	0.0	0.0	0.0
20	0.22007	0.01639	1000	22	0	0.0	0.0	0.0
21	0.17844	0.01375	1000	8	0	0.0	0.0	0.0
22	0.21021	0.01336	1000	22	0	0.0	0.0	0.0
23	0.17843	0.01208	1000	9	0	0.0	0.0	0.0
24	0.20213	0.01374	1000	21	0	0.0	0.0	0.0
25	0.17449	0.01457	1000	9	0	0.0	0.0	0.0
26	0.52913	0.03763	1000	21	0	0.0	0.0	0.0

Table G.7: PBT_R_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	77.57900	6.74262	1000	31000	0	0.0	0.0	0.0
2	52.72200	4.98495	1000	12000	0	0.0	0.0	0.0
3	13.31000	1.22982	1000	10	0	0.0	0.0	0.0
4	6.85998	0.75317	1000	19	0	0.0	0.0	0.0
5	5.40813	0.49948	1000	12	0	0.0	0.0	0.0
6	5.63745	0.60625	1000	17	0	0.0	0.0	0.0
7	9.79615	0.98748	1000	12	0	0.0	0.0	0.0
8	5.39708	0.50656	1000	19	0	0.0	0.0	0.0
9	5.38723	0.47871	1000	12	0	0.0	0.0	0.0
10	5.66060	0.60617	1000	19	0	0.0	0.0	0.0
11	5.23238	0.55190	1000	11	0	0.0	0.0	0.0
12	5.28252	0.55460	1000	20	0	0.0	0.0	0.0
13	5.26325	0.36627	1000	11	0	0.0	0.0	0.0
14	5.39518	0.44954	1000	20	0	0.0	0.0	0.0
15	5.30810	0.50365	1000	10	0	0.0	0.0	0.0
16	5.28940	0.47565	1000	1	0	0.0	0.0	100.0
17	5.16330	0.50328	1000	9	0	0.0	0.0	0.0
18	5.15935	0.38496	1000	22	0	0.0	0.0	0.0
19	5.18082	0.38716	1000	9	0	0.0	0.0	0.0
20	5.19870	0.46388	1000	22	0	0.0	0.0	0.0
21	5.18105	0.42620	1000	8	0	0.0	0.0	0.0
22	5.15755	0.29620	1000	22	0	0.0	0.0	0.0
23	5.21927	0.31868	1000	9	0	0.0	0.0	0.0
24	5.13130	0.29742	1000	21	0	0.0	0.0	0.0
25	5.29028	0.26631	1000	9	0	0.0	0.0	0.0
26	5.25635	0.31062	1000	21	0	0.0	0.0	0.0

Table G.8: PBT_R_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	57.06975	5.36300	1000	31000	0	0.0	0.0	0.0
2	29.56475	2.37704	1000	12000	0	0.0	0.0	0.0
3	4.12967	0.36732	1000	10	0	0.0	0.0	0.0
4	0.35001	0.15240	1000	19	0	0.0	0.0	0.0
5	0.27917	0.12577	1000	12	0	0.0	0.0	0.0
6	0.28341	0.13614	1000	17	0	0.0	0.0	0.0
7	0.27899	0.12126	1000	12	0	0.0	0.0	0.0
8	0.29007	0.14486	1000	19	0	0.0	0.0	0.0
9	0.25649	0.12754	1000	12	0	0.0	0.0	0.0
10	0.27113	0.11571	1000	19	0	0.0	0.0	0.0
11	0.23334	0.07910	1000	11	0	0.0	0.0	0.0
12	0.25056	0.08523	1000	20	0	0.0	0.0	0.0
13	0.21538	0.05461	1000	11	0	0.0	0.0	0.0
14	0.30569	0.06532	1000	20	0	0.0	0.0	0.0
15	0.23287	0.04921	1000	10	0	0.0	0.0	0.0
16	0.26451	0.03136	1000	1	0	0.0	0.0	100.0
17	0.20344	0.02132	1000	9	0	0.0	0.0	0.0
18	0.23218	0.02034	1000	22	0	0.0	0.0	0.0
19	0.19954	0.01627	1000	9	0	0.0	0.0	0.0
20	0.23054	0.01824	1000	22	0	0.0	0.0	0.0
21	0.19143	0.01666	1000	8	0	0.0	0.0	0.0
22	0.22645	0.02089	1000	22	0	0.0	0.0	0.0
23	0.18907	0.02208	1000	9	0	0.0	0.0	0.0
24	0.22397	0.02105	1000	21	0	0.0	0.0	0.0
25	0.20112	0.01212	1000	9	0	0.0	0.0	0.0
26	0.31500	0.14175	1000	21	0	0.0	0.0	0.0

Table G.9: PBT_R_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	81.69975	4.83239	1000	31000	0	0.0	0.0	0.0
2	54.78425	3.46818	1000	12000	0	0.0	0.0	0.0
3	13.66000	0.65268	1000	10	0	0.0	0.0	0.0
4	7.19552	0.55694	1000	19	0	0.0	0.0	0.0
5	5.67135	0.40575	1000	12	0	0.0	0.0	0.0
6	5.84115	0.34951	1000	17	0	0.0	0.0	0.0
7	5.67600	0.52685	1000	12	0	0.0	0.0	0.0
8	5.75790	0.36050	1000	19	0	0.0	0.0	0.0
9	5.47140	0.39444	1000	12	0	0.0	0.0	0.0
10	5.92623	0.38664	1000	19	0	0.0	0.0	0.0
11	5.52603	0.39328	1000	11	0	0.0	0.0	0.0
12	5.50463	0.45558	1000	20	0	0.0	0.0	0.0
13	5.40023	0.43568	1000	11	0	0.0	0.0	0.0
14	10.21438	0.64497	1000	20	0	0.0	0.0	0.0
15	5.39123	0.30214	1000	10	0	0.0	0.0	0.0
16	5.36170	0.20051	1000	1	0	0.0	0.0	100.0
17	5.47108	0.21236	1000	9	0	0.0	0.0	0.0
18	5.36848	0.21793	1000	22	0	0.0	0.0	0.0
19	5.39447	0.35116	1000	9	0	0.0	0.0	0.0
20	5.44573	0.23184	1000	22	0	0.0	0.0	0.0
21	5.36780	0.19652	1000	8	0	0.0	0.0	0.0
22	5.46647	0.40997	1000	22	0	0.0	0.0	0.0
23	5.37052	0.30095	1000	9	0	0.0	0.0	0.0
24	5.39370	0.36021	1000	21	0	0.0	0.0	0.0
25	5.35980	0.26585	1000	9	0	0.0	0.0	0.0
26	5.43400	0.20009	1000	21	0	0.0	0.0	0.0

Table G.10: PBT_R_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	80.79950	1.16268	1000	31000	0	0.0	0.0	0.0
2	53.44325	0.84894	1000	12000	0	0.0	0.0	0.0
3	13.57150	0.27459	1000	10	0	0.0	0.0	0.0
4	6.94465	0.25809	1000	19	0	0.0	0.0	0.0
5	5.54315	0.10756	1000	12	0	0.0	0.0	0.0
6	5.64792	0.12606	1000	17	0	0.0	0.0	0.0
7	5.59158	0.08657	1000	12	0	0.0	0.0	0.0
8	5.46588	0.10631	1000	19	0	0.0	0.0	0.0
9	5.55505	0.10636	1000	12	0	0.0	0.0	0.0
10	5.74755	0.07387	1000	19	0	0.0	0.0	0.0
11	5.37865	0.03695	1000	11	0	0.0	0.0	0.0
12	5.36653	0.16966	1000	20	0	0.0	0.0	0.0
13	5.25930	0.10634	1000	11	0	0.0	0.0	0.0
14	10.07183	0.13930	1000	20	0	0.0	0.0	0.0
15	5.33225	0.10870	1000	10	0	0.0	0.0	0.0
16	5.44670	0.20531	1000	1	0	0.0	0.0	100.0
17	5.36728	0.17303	1000	9	0	0.0	0.0	0.0
18	5.26995	0.15602	1000	22	0	0.0	0.0	0.0
19	5.18675	0.11849	1000	9	0	0.0	0.0	0.0
20	5.27177	0.03366	1000	22	0	0.0	0.0	0.0
21	5.17975	0.08878	1000	8	0	0.0	0.0	0.0
22	5.20207	0.08582	1000	22	0	0.0	0.0	0.0
23	5.22167	0.09660	1000	9	0	0.0	0.0	0.0
24	5.32693	0.08088	1000	21	0	0.0	0.0	0.0
25	5.23475	0.09636	1000	9	0	0.0	0.0	0.0
26	5.25310	0.15467	1000	21	0	0.0	0.0	0.0

Table G.11: PBT_CG_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22539.2500	491.98901	1000	1000000	0	73.36	0.0	0.0
2	25708.0000	389.28845	1000	1000000	0	85.47	0.0	0.0
3	184.3075	8.14949	1000	10	0	33.33	0.0	0.0
4	174.2900	7.48168	1000	19	0	33.59	0.0	0.0
5	175.7650	4.54375	1000	12	0	37.35	0.0	0.0
6	169.0900	4.89574	1000	17	0	37.09	0.0	0.0
7	166.4775	3.60937	1000	12	0	40.85	0.0	0.0
8	164.6900	4.76664	1000	19	0	40.59	0.0	0.0
9	168.0300	12.17663	1000	12	0	44.36	0.0	0.0
10	168.6575	8.13339	1000	19	0	44.09	0.0	0.0
11	165.6300	2.67360	1000	11	0	47.86	0.0	0.0
12	164.9525	2.20189	1000	20	0	47.59	0.0	0.0
13	163.3700	3.90587	1000	11	0	51.36	0.0	0.0
14	172.7000	2.67106	1000	20	0	51.09	0.0	0.0
15	171.5775	2.94412	1000	10	0	54.86	0.0	0.0
16	164.9700	4.65157	1000	1	0	73.63	0.0	100.0
17	192.6925	17.43353	1000	9	0	57.84	0.0	0.0
18	172.3450	32.54353	1000	22	0	58.10	0.0	0.0
19	148.8800	8.22468	1000	9	0	61.86	0.0	0.0
20	142.1150	2.62100	1000	22	0	61.60	0.0	0.0
21	143.5025	6.26638	1000	8	0	65.36	0.0	0.0
22	143.7075	4.54553	1000	22	0	65.10	0.0	0.0
23	145.2875	8.59121	1000	9	0	68.86	0.0	0.0
24	139.2550	5.24772	1000	21	0	68.60	0.0	0.0
25	139.8200	6.31126	1000	9	0	72.37	0.0	0.0
26	156.9950	7.28161	1000	21	0	72.37	0.0	0.0

Table G.12: PBT_CG_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22816.5000	659.79144	1000	1000000	0	73.36	0.0	0.0
2	25831.2500	545.33677	1000	1000000	0	85.47	0.0	0.0
3	187.1250	5.07149	1000	10	0	33.33	0.0	0.0
4	178.1900	3.93573	1000	19	0	33.59	0.0	0.0
5	171.6950	0.35822	1000	12	0	37.35	0.0	0.0
6	167.8050	1.71560	1000	17	0	37.09	0.0	0.0
7	160.5350	2.87126	1000	12	0	40.85	0.0	0.0
8	162.4650	3.91675	1000	19	0	40.59	0.0	0.0
9	159.4850	1.90836	1000	12	0	44.36	0.0	0.0
10	166.1350	1.27143	1000	19	0	44.09	0.0	0.0
11	163.2225	4.34413	1000	11	0	47.86	0.0	0.0
12	163.9525	4.03610	1000	20	0	47.59	0.0	0.0
13	160.9275	3.51373	1000	11	0	51.36	0.0	0.0
14	173.1375	2.12317	1000	20	0	51.09	0.0	0.0
15	174.5750	1.53871	1000	10	0	54.86	0.0	0.0
16	169.7750	3.35044	1000	1	0	73.63	0.0	100.0
17	194.0975	20.97616	1000	9	0	57.84	0.0	0.0
18	176.8425	21.81247	1000	22	0	58.10	0.0	0.0
19	146.6100	7.17098	1000	9	0	61.86	0.0	0.0
20	142.9025	3.93748	1000	22	0	61.60	0.0	0.0
21	141.1725	4.55713	1000	8	0	65.36	0.0	0.0
22	139.9125	3.48413	1000	22	0	65.10	0.0	0.0
23	137.5600	4.11637	1000	9	0	68.86	0.0	0.0
24	133.6950	0.94299	1000	21	0	68.60	0.0	0.0
25	133.8350	0.67300	1000	9	0	72.37	0.0	0.0
26	148.6425	1.44873	1000	21	0	72.37	0.0	0.0

Table G.13: PBT_CG_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22911.2500	649.21120	1000	1000000	0	73.36	0.0	0.0
2	25982.0000	696.77364	1000	1000000	0	85.47	0.0	0.0
3	197.7850	2.10435	1000	10	0	33.33	0.0	0.0
4	188.5725	3.87882	1000	19	0	33.59	0.0	0.0
5	177.9375	2.66168	1000	12	0	37.35	0.0	0.0
6	173.9675	3.03319	1000	17	0	37.09	0.0	0.0
7	172.4200	0.47576	1000	12	0	40.85	0.0	0.0
8	164.0450	1.58494	1000	19	0	40.59	0.0	0.0
9	165.9950	1.36034	1000	12	0	44.36	0.0	0.0
10	168.8100	0.69004	1000	19	0	44.09	0.0	0.0
11	167.6725	0.53190	1000	11	0	47.86	0.0	0.0
12	170.9700	1.39478	1000	20	0	47.59	0.0	0.0
13	167.5175	0.46273	1000	11	0	51.36	0.0	0.0
14	175.8700	1.86647	1000	20	0	51.09	0.0	0.0
15	181.3250	1.00256	1000	10	0	54.86	0.0	0.0
16	187.4025	23.27639	1000	1	0	73.63	0.0	100.0
17	173.4625	2.21080	1000	9	0	57.84	0.0	0.0
18	198.1075	28.70432	1000	22	0	58.10	0.0	0.0
19	156.8325	9.81735	1000	9	0	61.86	0.0	0.0
20	152.1725	9.71543	1000	22	0	61.60	0.0	0.0
21	147.6850	4.98050	1000	8	0	65.36	0.0	0.0
22	148.8825	4.58418	1000	22	0	65.10	0.0	0.0
23	145.0000	3.59852	1000	9	0	68.86	0.0	0.0
24	143.7350	4.71128	1000	21	0	68.60	0.0	0.0
25	145.4775	7.62546	1000	9	0	72.37	0.0	0.0
26	158.7525	6.84545	1000	21	0	72.37	0.0	0.0

Table G.14: PBT_CG_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22721.7500	494.88351	1000	1000000	0	73.36	0.0	0.0
2	25606.0000	931.90692	1000	1000000	0	85.47	0.0	0.0
3	192.1300	3.87534	1000	10	0	33.33	0.0	0.0
4	177.2975	3.52567	1000	19	0	33.59	0.0	0.0
5	174.4400	1.78377	1000	12	0	37.35	0.0	0.0
6	166.4925	2.67001	1000	17	0	37.09	0.0	0.0
7	161.8475	1.89604	1000	12	0	40.85	0.0	0.0
8	159.2200	3.03817	1000	19	0	40.59	0.0	0.0
9	158.8350	1.65919	1000	12	0	44.36	0.0	0.0
10	163.8350	3.63225	1000	19	0	44.09	0.0	0.0
11	163.4475	0.87081	1000	11	0	47.86	0.0	0.0
12	163.5075	2.06273	1000	20	0	47.59	0.0	0.0
13	159.9025	2.08575	1000	11	0	51.36	0.0	0.0
14	165.7675	4.60808	1000	20	0	51.09	0.0	0.0
15	175.6475	4.87175	1000	10	0	54.86	0.0	0.0
16	167.8375	2.73985	1000	1	0	73.63	0.0	100.0
17	190.1575	11.87603	1000	9	0	57.84	0.0	0.0
18	166.7950	17.59487	1000	22	0	58.10	0.0	0.0
19	142.3425	2.80109	1000	9	0	61.86	0.0	0.0
20	138.4475	0.79534	1000	22	0	61.60	0.0	0.0
21	138.9125	1.94017	1000	8	0	65.36	0.0	0.0
22	141.8950	4.26265	1000	22	0	65.10	0.0	0.0
23	138.6025	0.83085	1000	9	0	68.86	0.0	0.0
24	135.2850	1.42796	1000	21	0	68.60	0.0	0.0
25	135.8875	1.77722	1000	9	0	72.37	0.0	0.0
26	149.9250	1.75955	1000	21	0	72.37	0.0	0.0

Table G.15: PBT_CG_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22282.0000	63.45077	1000	1000000	0	73.36	0.0	0.0
2	25336.0000	134.73492	1000	1000000	0	85.47	0.0	0.0
3	196.1825	1.33113	1000	10	0	33.33	0.0	0.0
4	191.1400	1.88186	1000	19	0	33.59	0.0	0.0
5	181.3825	1.95723	1000	12	0	37.35	0.0	0.0
6	175.3725	3.19021	1000	17	0	37.09	0.0	0.0
7	173.4575	1.33623	1000	12	0	40.85	0.0	0.0
8	167.5950	1.63726	1000	19	0	40.59	0.0	0.0
9	167.0275	1.32220	1000	12	0	44.36	0.0	0.0
10	170.9000	0.59405	1000	19	0	44.09	0.0	0.0
11	170.8225	1.36767	1000	11	0	47.86	0.0	0.0
12	170.8350	1.58281	1000	20	0	47.59	0.0	0.0
13	167.7675	1.33280	1000	11	0	51.36	0.0	0.0
14	175.9550	1.09422	1000	20	0	51.09	0.0	0.0
15	179.0550	1.74752	1000	10	0	54.86	0.0	0.0
16	173.7600	2.07573	1000	1	0	73.63	0.0	100.0
17	190.9000	18.24329	1000	9	0	57.84	0.0	0.0
18	179.1650	18.08907	1000	22	0	58.10	0.0	0.0
19	154.4875	10.90190	1000	9	0	61.86	0.0	0.0
20	148.9875	3.04061	1000	22	0	61.60	0.0	0.0
21	145.0825	2.44475	1000	8	0	65.36	0.0	0.0
22	146.9100	1.94747	1000	22	0	65.10	0.0	0.0
23	143.3150	2.75309	1000	9	0	68.86	0.0	0.0
24	143.8325	4.74101	1000	21	0	68.60	0.0	0.0
25	141.2400	1.72863	1000	9	0	72.37	0.0	0.0
26	155.4500	0.97553	1000	21	0	72.37	0.0	0.0

Table G.16: PBT_CG_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22412.7500	104.01773	1000	1000000	0	73.36	0.0	0.0
2	25379.2500	342.09821	1000	1000000	0	85.47	0.0	0.0
3	190.5425	5.76063	1000	10	0	33.33	0.0	0.0
4	176.4675	4.91217	1000	19	0	33.59	0.0	0.0
5	173.0850	3.07363	1000	12	0	37.35	0.0	0.0
6	164.9625	1.83530	1000	17	0	37.09	0.0	0.0
7	160.1750	2.86990	1000	12	0	40.85	0.0	0.0
8	158.1900	2.54824	1000	19	0	40.59	0.0	0.0
9	156.8375	0.79679	1000	12	0	44.36	0.0	0.0
10	161.8975	1.71958	1000	19	0	44.09	0.0	0.0
11	163.2300	0.92941	1000	11	0	47.86	0.0	0.0
12	163.5000	0.91575	1000	20	0	47.59	0.0	0.0
13	157.6600	1.44345	1000	11	0	51.36	0.0	0.0
14	166.2550	2.10338	1000	20	0	51.09	0.0	0.0
15	173.3975	1.23366	1000	10	0	54.86	0.0	0.0
16	167.7675	0.81806	1000	1	0	73.63	0.0	100.0
17	191.8475	13.62627	1000	9	0	57.84	0.0	0.0
18	164.6375	16.19363	1000	22	0	58.10	0.0	0.0
19	143.4150	4.17920	1000	9	0	61.86	0.0	0.0
20	140.4975	2.55784	1000	22	0	61.60	0.0	0.0
21	138.3800	1.13475	1000	8	0	65.36	0.0	0.0
22	137.8050	2.53662	1000	22	0	65.10	0.0	0.0
23	136.6650	2.39992	1000	9	0	68.86	0.0	0.0
24	133.2075	2.19216	1000	21	0	68.60	0.0	0.0
25	134.2275	3.10468	1000	9	0	72.37	0.0	0.0
26	149.1450	2.72204	1000	21	0	72.37	0.0	0.0

Table G.17: PBT_CG_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22956.5000	1331.55651	1000	1000000	0	73.36	0.0	0.0
2	25981.5000	1046.81720	1000	1000000	0	85.47	0.0	0.0
3	202.7875	6.28601	1000	10	0	33.33	0.0	0.0
4	192.3250	7.71498	1000	19	0	33.59	0.0	0.0
5	185.3125	6.20420	1000	12	0	37.35	0.0	0.0
6	175.3350	3.48441	1000	17	0	37.09	0.0	0.0
7	169.1475	4.06721	1000	12	0	40.85	0.0	0.0
8	168.4900	4.91336	1000	19	0	40.59	0.0	0.0
9	173.8425	12.17626	1000	12	0	44.36	0.0	0.0
10	176.4800	9.41712	1000	19	0	44.09	0.0	0.0
11	174.6500	9.52167	1000	11	0	47.86	0.0	0.0
12	174.3175	5.25734	1000	20	0	47.59	0.0	0.0
13	166.3450	2.98542	1000	11	0	51.36	0.0	0.0
14	181.4225	3.41663	1000	20	0	51.09	0.0	0.0
15	182.9075	4.93928	1000	10	0	54.86	0.0	0.0
16	175.7675	3.07982	1000	1	0	73.63	0.0	100.0
17	200.6100	15.42759	1000	9	0	57.84	0.0	0.0
18	180.3950	31.04871	1000	22	0	58.10	0.0	0.0
19	167.2250	35.86418	1000	9	0	61.86	0.0	0.0
20	150.8575	8.33905	1000	22	0	61.60	0.0	0.0
21	146.4400	3.11924	1000	8	0	65.36	0.0	0.0
22	147.0575	1.91855	1000	22	0	65.10	0.0	0.0
23	144.0850	2.24433	1000	9	0	68.86	0.0	0.0
24	141.4475	2.33951	1000	21	0	68.60	0.0	0.0
25	142.2025	2.07101	1000	9	0	72.37	0.0	0.0
26	156.6650	3.68210	1000	21	0	72.37	0.0	0.0

Table G.18: PBT_CG_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22578.7500	490.39850	1000	1000000	0	73.36	0.0	0.0
2	25767.0000	477.54424	1000	1000000	0	85.47	0.0	0.0
3	196.3450	4.05459	1000	10	0	33.33	0.0	0.0
4	185.6925	2.08203	1000	19	0	33.59	0.0	0.0
5	179.9025	2.28437	1000	12	0	37.35	0.0	0.0
6	176.2725	4.42074	1000	17	0	37.09	0.0	0.0
7	168.3075	1.55447	1000	12	0	40.85	0.0	0.0
8	166.8000	1.97516	1000	19	0	40.59	0.0	0.0
9	164.3800	2.35110	1000	12	0	44.36	0.0	0.0
10	169.9825	2.44882	1000	19	0	44.09	0.0	0.0
11	170.5950	3.61229	1000	11	0	47.86	0.0	0.0
12	170.3075	5.08260	1000	20	0	47.59	0.0	0.0
13	165.3950	1.19939	1000	11	0	51.36	0.0	0.0
14	183.4575	1.46532	1000	20	0	51.09	0.0	0.0
15	185.2500	2.88545	1000	10	0	54.86	0.0	0.0
16	196.9075	21.01064	1000	1	0	73.63	0.0	100.0
17	179.3775	23.67856	1000	9	0	57.84	0.0	0.0
18	166.1050	23.68106	1000	22	0	58.10	0.0	0.0
19	149.5800	8.79039	1000	9	0	61.86	0.0	0.0
20	151.1800	8.49702	1000	22	0	61.60	0.0	0.0
21	144.1300	4.03112	1000	8	0	65.36	0.0	0.0
22	145.3925	3.60465	1000	22	0	65.10	0.0	0.0
23	142.5800	2.60637	1000	9	0	68.86	0.0	0.0
24	141.2925	4.65285	1000	21	0	68.60	0.0	0.0
25	140.2475	2.80519	1000	9	0	72.37	0.0	0.0
26	156.7250	3.57302	1000	21	0	72.37	0.0	0.0

G.2 M1

Table G.19: M1 - Search strategies results.

Strategy	Min Mean Time (ms)	Std Min Mean Time (ms)	Max Mean Time (ms)	Std Max Mean Time (ms)	Min Sample Size	Max Sample Size	Min Mean Test Cases	Std Min Mean Test Cases	Max Mean Test Cases	Std Max Mean Test Cases	Min Mean Coverage (%)	Std Min Mean Coverage (%)	Max Mean Coverage (%)	Std Max Mean Coverage (%)	Bug Found
PBT_R_S4	0.17401	0.00622	20.43600	0.44378	1000	1000	1	0	1	0	0.00	0.0	0.00	0.00	True
PBT_R_S1	0.18725	0.01338	23.24900	4.65446	1000	1000	1	0	1	0	0.00	0.0	0.00	0.00	True
PBT_R_S8	5.19015	0.14839	40.15950	0.46876	1000	1000	1	0	2	0	0.00	0.0	0.00	0.00	True
PBT_CG_S4	129.42750	0.66815	316.64750	0.49398	1000	1000	1	0	2	0	36.55	0.0	66.08	16.75	True
PBT_CG_S6	128.70250	2.20349	318.51750	2.54583	1000	1000	1	0	1	0	36.55	0.0	37.07	0.00	True
PBT_CG_S3	131.69250	1.54918	341.56250	4.64457	1000	1000	1	0	2	0	36.55	0.0	75.75	0.00	True
PBT_R_S3	5.18637	0.16198	39.93325	0.77173	1000	1000	1	0	2	0	0.00	0.0	0.00	0.00	True
PBT_R_S5	5.34527	0.25844	38.60550	0.97233	1000	1000	1	0	2	0	0.00	0.0	0.00	0.00	True
PBT_R_S7	5.16635	0.04334	39.71925	0.53395	1000	1000	1	0	2	0	0.00	0.0	0.00	0.00	True
PBT_CG_S5	134.31000	2.79125	336.98250	7.27568	1000	1000	1	0	2	0	36.55	0.0	75.75	0.00	True
PBT_CG_S7	133.81750	0.73121	331.19500	2.30385	1000	1000	1	0	2	0	36.55	0.0	75.75	0.00	True
PBT_CG_S1	127.15250	1.44148	306.08000	4.64236	1000	1000	1	0	1	0	36.55	0.0	56.15	19.60	True
PBT_CG_S8	133.46000	0.39718	334.05500	6.12885	1000	1000	1	0	2	0	36.55	0.0	75.75	0.00	True
PBT_CG_S2	128.45000	1.49618	316.79500	3.48980	1000	1000	1	0	1	0	36.55	0.0	37.07	0.00	True
PBT_R_S6	0.18181	0.00475	21.16675	0.83136	1000	1000	1	0	1	0	0.00	0.0	0.00	0.00	True
PBT_R_S2	0.18391	0.00588	20.68325	0.29517	1000	1000	1	0	1	0	0.00	0.0	0.00	0.00	True

Table G.20: M1 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-0.0468	0.9000	-5.0826	4.9890	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.0476	0.9000	-4.9882	5.0834	False
Random withConstantExtraction	Random withoutConstantExtraction	-5.0403	0.0010	-5.1418	-4.9387	True
Coverage-Guided withBias	Coverage-Guided withoutBias	-0.7081	0.7604	-5.8679	4.4516	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-0.9506	0.6796	-6.0713	4.1701	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-4.8869	0.0010	-6.7012	-3.0725	True
Coverage-Guided	Random	-128.1747	0.0010	-131.1142	-125.2351	True

G.2.1 Search Strategies

Table G.21: PBT_R_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	23.24900	4.65446	1000	1	0	0.0	0.0	100.0
2	9.87270	2.06597	1000	1	0	0.0	0.0	100.0
3	0.39921	0.03736	1000	10	0	0.0	0.0	0.0
4	0.32887	0.02574	1000	19	0	0.0	0.0	0.0
5	0.26733	0.01956	1000	12	0	0.0	0.0	0.0
6	0.26939	0.02059	1000	17	0	0.0	0.0	0.0
7	0.20657	0.02168	1000	1	0	0.0	0.0	100.0
8	0.21990	0.01610	1000	1	0	0.0	0.0	100.0
9	0.22316	0.01922	1000	1	0	0.0	0.0	100.0
10	0.22203	0.01627	1000	1	0	0.0	0.0	100.0
11	0.20546	0.01712	1000	1	0	0.0	0.0	100.0
12	0.22075	0.01676	1000	1	0	0.0	0.0	100.0
13	0.20289	0.01905	1000	1	0	0.0	0.0	100.0
14	0.21565	0.02057	1000	1	0	0.0	0.0	100.0
15	0.20016	0.01837	1000	1	0	0.0	0.0	100.0
16	0.21318	0.01762	1000	1	0	0.0	0.0	100.0
17	0.19543	0.01754	1000	1	0	0.0	0.0	100.0
18	0.21251	0.02190	1000	1	0	0.0	0.0	100.0
19	0.19967	0.01180	1000	1	0	0.0	0.0	100.0
20	0.20656	0.01679	1000	1	0	0.0	0.0	100.0
21	0.18725	0.01338	1000	1	0	0.0	0.0	100.0
22	0.21911	0.01185	1000	1	0	0.0	0.0	100.0
23	0.18875	0.01339	1000	1	0	0.0	0.0	100.0
24	0.28928	0.14923	1000	1	0	0.0	0.0	100.0
25	0.27197	0.14707	1000	1	0	0.0	0.0	100.0
26	0.20105	0.01649	1000	1	0	0.0	0.0	100.0

Table G.22: PBT_R_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	20.68325	0.29517	1000	1	0	0.0	0.0	100.0
2	8.06388	0.38667	1000	1	0	0.0	0.0	100.0
3	0.40407	0.00520	1000	10	0	0.0	0.0	0.0
4	0.38067	0.01621	1000	19	0	0.0	0.0	0.0
5	0.30329	0.02785	1000	12	0	0.0	0.0	0.0
6	0.31705	0.03496	1000	17	0	0.0	0.0	0.0
7	0.26030	0.02647	1000	1	0	0.0	0.0	100.0
8	0.25789	0.02321	1000	1	0	0.0	0.0	100.0
9	0.27060	0.02326	1000	1	0	0.0	0.0	100.0
10	0.25758	0.02377	1000	1	0	0.0	0.0	100.0
11	0.24197	0.02245	1000	1	0	0.0	0.0	100.0
12	0.25491	0.01753	1000	1	0	0.0	0.0	100.0
13	0.24150	0.02077	1000	1	0	0.0	0.0	100.0
14	0.25877	0.02354	1000	1	0	0.0	0.0	100.0
15	0.24456	0.01386	1000	1	0	0.0	0.0	100.0
16	0.26094	0.02219	1000	1	0	0.0	0.0	100.0
17	0.22844	0.01555	1000	1	0	0.0	0.0	100.0
18	0.22414	0.00368	1000	1	0	0.0	0.0	100.0
19	0.20329	0.00493	1000	1	0	0.0	0.0	100.0
20	0.21482	0.01296	1000	1	0	0.0	0.0	100.0
21	0.18819	0.00551	1000	1	0	0.0	0.0	100.0
22	0.19996	0.00988	1000	1	0	0.0	0.0	100.0
23	0.18423	0.00422	1000	1	0	0.0	0.0	100.0
24	0.19373	0.00674	1000	1	0	0.0	0.0	100.0
25	0.18391	0.00588	1000	1	0	0.0	0.0	100.0
26	0.37051	0.16905	1000	1	0	0.0	0.0	100.0

Table G.23: PBT_R_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	39.93325	0.77173	1000	1	0	0.0	0.0	100.0
2	31.26450	0.42593	1000	2	0	0.0	0.0	100.0
3	9.81177	0.16506	1000	10	0	0.0	0.0	0.0
4	7.07925	0.09310	1000	19	0	0.0	0.0	0.0
5	5.79460	0.19035	1000	12	0	0.0	0.0	0.0
6	5.79327	0.06264	1000	17	0	0.0	0.0	0.0
7	10.09240	0.16547	1000	1	0	0.0	0.0	100.0
8	5.62928	0.09177	1000	1	0	0.0	0.0	100.0
9	5.86085	0.08320	1000	1	0	0.0	0.0	100.0
10	5.48770	0.05319	1000	1	0	0.0	0.0	100.0
11	5.47642	0.06053	1000	1	0	0.0	0.0	100.0
12	5.40337	0.11880	1000	1	0	0.0	0.0	100.0
13	5.41910	0.03946	1000	1	0	0.0	0.0	100.0
14	5.71882	0.06724	1000	1	0	0.0	0.0	100.0
15	5.45235	0.05227	1000	1	0	0.0	0.0	100.0
16	5.40638	0.08124	1000	1	0	0.0	0.0	100.0
17	5.25560	0.10166	1000	1	0	0.0	0.0	100.0
18	5.22013	0.03958	1000	1	0	0.0	0.0	100.0
19	5.29680	0.04920	1000	1	0	0.0	0.0	100.0
20	5.20925	0.07444	1000	1	0	0.0	0.0	100.0
21	5.20852	0.13500	1000	1	0	0.0	0.0	100.0
22	5.26608	0.14102	1000	1	0	0.0	0.0	100.0
23	5.19410	0.11929	1000	1	0	0.0	0.0	100.0
24	5.18637	0.16198	1000	1	0	0.0	0.0	100.0
25	5.33113	0.08961	1000	1	0	0.0	0.0	100.0
26	5.30535	0.12199	1000	1	0	0.0	0.0	100.0

Table G.24: PBT_R_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	20.43600	0.44378	1000	1	0	0.0	0.0	100.0
2	8.45673	0.15493	1000	1	0	0.0	0.0	100.0
3	0.38009	0.00802	1000	10	0	0.0	0.0	0.0
4	0.35618	0.03807	1000	19	0	0.0	0.0	0.0
5	0.27663	0.02135	1000	12	0	0.0	0.0	0.0
6	0.28658	0.02204	1000	17	0	0.0	0.0	0.0
7	0.22477	0.02020	1000	1	0	0.0	0.0	100.0
8	0.23381	0.02170	1000	1	0	0.0	0.0	100.0
9	0.24183	0.02360	1000	1	0	0.0	0.0	100.0
10	0.22803	0.01761	1000	1	0	0.0	0.0	100.0
11	0.22087	0.01630	1000	1	0	0.0	0.0	100.0
12	0.23104	0.01929	1000	1	0	0.0	0.0	100.0
13	0.21398	0.02124	1000	1	0	0.0	0.0	100.0
14	0.23220	0.01948	1000	1	0	0.0	0.0	100.0
15	0.21306	0.02251	1000	1	0	0.0	0.0	100.0
16	0.22000	0.01606	1000	1	0	0.0	0.0	100.0
17	0.19227	0.00454	1000	1	0	0.0	0.0	100.0
18	0.20650	0.00937	1000	1	0	0.0	0.0	100.0
19	0.19444	0.00655	1000	1	0	0.0	0.0	100.0
20	0.19623	0.00761	1000	1	0	0.0	0.0	100.0
21	0.17401	0.00622	1000	1	0	0.0	0.0	100.0
22	0.19537	0.00609	1000	1	0	0.0	0.0	100.0
23	0.17468	0.00602	1000	1	0	0.0	0.0	100.0
24	0.19287	0.01168	1000	1	0	0.0	0.0	100.0
25	0.34760	0.17210	1000	1	0	0.0	0.0	100.0
26	0.36295	0.17294	1000	1	0	0.0	0.0	100.0

Table G.25: PBT_R_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	38.60550	0.97233	1000	1	0	0.0	0.0	100.0
2	30.69125	0.30513	1000	2	0	0.0	0.0	100.0
3	9.66170	0.23979	1000	10	0	0.0	0.0	0.0
4	7.10290	0.10977	1000	19	0	0.0	0.0	0.0
5	5.69703	0.15484	1000	12	0	0.0	0.0	0.0
6	5.78983	0.09024	1000	17	0	0.0	0.0	0.0
7	10.03025	0.11837	1000	1	0	0.0	0.0	100.0
8	5.53130	0.10944	1000	1	0	0.0	0.0	100.0
9	5.89277	0.13078	1000	1	0	0.0	0.0	100.0
10	5.47738	0.05438	1000	1	0	0.0	0.0	100.0
11	5.42882	0.11798	1000	1	0	0.0	0.0	100.0
12	5.41672	0.05649	1000	1	0	0.0	0.0	100.0
13	5.40873	0.07423	1000	1	0	0.0	0.0	100.0
14	5.67640	0.06247	1000	1	0	0.0	0.0	100.0
15	5.37160	0.10435	1000	1	0	0.0	0.0	100.0
16	5.50413	0.16857	1000	1	0	0.0	0.0	100.0
17	5.43697	0.17387	1000	1	0	0.0	0.0	100.0
18	5.54812	0.20888	1000	1	0	0.0	0.0	100.0
19	5.80023	0.87724	1000	1	0	0.0	0.0	100.0
20	5.66883	0.67111	1000	1	0	0.0	0.0	100.0
21	5.35330	0.37251	1000	1	0	0.0	0.0	100.0
22	5.37027	0.26001	1000	1	0	0.0	0.0	100.0
23	5.34527	0.25844	1000	1	0	0.0	0.0	100.0
24	5.37365	0.22180	1000	1	0	0.0	0.0	100.0
25	5.52197	0.19027	1000	1	0	0.0	0.0	100.0
26	5.43200	0.23713	1000	1	0	0.0	0.0	100.0

Table G.26: PBT_R_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	21.16675	0.83136	1000	1	0	0.0	0.0	100.0
2	8.55195	0.60090	1000	1	0	0.0	0.0	100.0
3	0.39866	0.00183	1000	10	0	0.0	0.0	0.0
4	0.59554	0.21490	1000	19	0	0.0	0.0	0.0
5	0.45790	0.16532	1000	12	0	0.0	0.0	0.0
6	0.48847	0.15529	1000	17	0	0.0	0.0	0.0
7	0.35403	0.13463	1000	1	0	0.0	0.0	100.0
8	0.35936	0.13598	1000	1	0	0.0	0.0	100.0
9	0.35414	0.12561	1000	1	0	0.0	0.0	100.0
10	0.31133	0.07164	1000	1	0	0.0	0.0	100.0
11	0.26500	0.03945	1000	1	0	0.0	0.0	100.0
12	0.27886	0.03976	1000	1	0	0.0	0.0	100.0
13	0.25037	0.01711	1000	1	0	0.0	0.0	100.0
14	0.26336	0.02659	1000	1	0	0.0	0.0	100.0
15	0.23295	0.02121	1000	1	0	0.0	0.0	100.0
16	0.23990	0.01386	1000	1	0	0.0	0.0	100.0
17	0.21000	0.01110	1000	1	0	0.0	0.0	100.0
18	0.22129	0.01331	1000	1	0	0.0	0.0	100.0
19	0.19861	0.00446	1000	1	0	0.0	0.0	100.0
20	0.20268	0.00742	1000	1	0	0.0	0.0	100.0
21	0.18701	0.00559	1000	1	0	0.0	0.0	100.0
22	0.19911	0.00513	1000	1	0	0.0	0.0	100.0
23	0.18181	0.00475	1000	1	0	0.0	0.0	100.0
24	0.28082	0.14728	1000	1	0	0.0	0.0	100.0
25	0.21210	0.04037	1000	1	0	0.0	0.0	100.0
26	0.31183	0.13565	1000	1	0	0.0	0.0	100.0

Table G.27: PBT_R_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	39.71925	0.53395	1000	1	0	0.0	0.0	100.0
2	30.93350	0.49958	1000	2	0	0.0	0.0	100.0
3	9.59240	0.10856	1000	10	0	0.0	0.0	0.0
4	7.00873	0.09955	1000	19	0	0.0	0.0	0.0
5	5.61570	0.08704	1000	12	0	0.0	0.0	0.0
6	5.78883	0.17500	1000	17	0	0.0	0.0	0.0
7	5.65488	0.11481	1000	1	0	0.0	0.0	100.0
8	5.61615	0.06768	1000	1	0	0.0	0.0	100.0
9	5.93248	0.05579	1000	1	0	0.0	0.0	100.0
10	5.46033	0.07212	1000	1	0	0.0	0.0	100.0
11	5.53840	0.01742	1000	1	0	0.0	0.0	100.0
12	5.38455	0.08269	1000	1	0	0.0	0.0	100.0
13	5.40730	0.04379	1000	1	0	0.0	0.0	100.0
14	10.08238	0.12186	1000	1	0	0.0	0.0	100.0
15	5.35695	0.13161	1000	1	0	0.0	0.0	100.0
16	5.32265	0.04647	1000	1	0	0.0	0.0	100.0
17	5.27820	0.11994	1000	1	0	0.0	0.0	100.0
18	5.28995	0.08370	1000	1	0	0.0	0.0	100.0
19	5.26947	0.13075	1000	1	0	0.0	0.0	100.0
20	5.29282	0.03077	1000	1	0	0.0	0.0	100.0
21	5.19053	0.14493	1000	1	0	0.0	0.0	100.0
22	5.16635	0.04334	1000	1	0	0.0	0.0	100.0
23	5.17943	0.08122	1000	1	0	0.0	0.0	100.0
24	5.19535	0.10022	1000	1	0	0.0	0.0	100.0
25	5.27030	0.15637	1000	1	0	0.0	0.0	100.0
26	5.26785	0.17296	1000	1	0	0.0	0.0	100.0

Table G.28: PBT_R_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	40.15950	0.46876	1000	1	0	0.0	0.0	100.0
2	31.01825	0.45294	1000	2	0	0.0	0.0	100.0
3	9.60750	0.18347	1000	10	0	0.0	0.0	0.0
4	7.28180	0.26328	1000	19	0	0.0	0.0	0.0
5	5.76548	0.11168	1000	12	0	0.0	0.0	0.0
6	5.81940	0.08448	1000	17	0	0.0	0.0	0.0
7	5.52747	0.09882	1000	1	0	0.0	0.0	100.0
8	5.46717	0.13618	1000	1	0	0.0	0.0	100.0
9	5.82608	0.10617	1000	1	0	0.0	0.0	100.0
10	5.49035	0.10546	1000	1	0	0.0	0.0	100.0
11	5.43383	0.05106	1000	1	0	0.0	0.0	100.0
12	5.54177	0.13823	1000	1	0	0.0	0.0	100.0
13	5.43347	0.17153	1000	1	0	0.0	0.0	100.0
14	10.22050	0.09245	1000	1	0	0.0	0.0	100.0
15	5.44540	0.17506	1000	1	0	0.0	0.0	100.0
16	5.49503	0.13050	1000	1	0	0.0	0.0	100.0
17	5.34833	0.13800	1000	1	0	0.0	0.0	100.0
18	5.26410	0.12040	1000	1	0	0.0	0.0	100.0
19	5.33062	0.15942	1000	1	0	0.0	0.0	100.0
20	5.30535	0.09269	1000	1	0	0.0	0.0	100.0
21	5.29002	0.20941	1000	1	0	0.0	0.0	100.0
22	5.32290	0.09048	1000	1	0	0.0	0.0	100.0
23	5.19015	0.14839	1000	1	0	0.0	0.0	100.0
24	5.23093	0.11892	1000	1	0	0.0	0.0	100.0
25	5.28110	0.18874	1000	1	0	0.0	0.0	100.0
26	5.26435	0.16020	1000	1	0	0.0	0.0	100.0

Table G.29: PBT_CG_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	306.0800	4.64236	1000	1	0	36.55	0.0	100.0
2	180.7775	2.52607	1000	1	0	56.15	19.6	100.0
3	158.0875	3.29305	1000	10	0	33.32	0.0	0.0
4	145.6300	4.36745	1000	19	0	33.58	0.0	0.0
5	141.5225	1.86073	1000	12	0	37.07	0.0	0.0
6	134.0175	1.22387	1000	17	0	36.55	0.0	0.0
7	129.1600	0.21622	1000	1	0	36.55	0.0	100.0
8	127.1525	1.44148	1000	1	0	36.55	0.0	100.0
9	129.9775	0.18458	1000	1	0	36.55	0.0	100.0
10	132.7450	0.29030	1000	1	0	36.55	0.0	100.0
11	136.0050	3.19409	1000	1	0	36.55	0.0	100.0
12	132.6100	1.64059	1000	1	0	36.55	0.0	100.0
13	133.9425	1.10590	1000	1	0	36.55	0.0	100.0
14	142.0525	0.52256	1000	1	0	36.55	0.0	100.0
15	151.4250	6.87733	1000	1	0	36.55	0.0	100.0
16	140.9875	5.42389	1000	1	0	36.55	0.0	100.0
17	142.0950	1.30825	1000	1	0	36.55	0.0	100.0
18	135.4500	1.00869	1000	1	0	36.55	0.0	100.0
19	138.4575	1.96544	1000	1	0	36.55	0.0	100.0
20	142.1300	1.90713	1000	1	0	36.55	0.0	100.0
21	141.3525	1.54634	1000	1	0	36.55	0.0	100.0
22	206.4200	3.89163	1000	1	0	36.55	0.0	100.0
23	165.7875	3.63771	1000	1	0	36.55	0.0	100.0
24	151.5625	1.31456	1000	1	0	36.55	0.0	100.0
25	149.5650	2.79700	1000	1	0	36.55	0.0	100.0
26	159.2550	2.98380	1000	1	0	36.81	0.0	100.0

Table G.30: PBT_CG_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	316.7950	3.48980	1000	1	0	36.55	0.0	100.0
2	185.4100	4.79494	1000	1	0	37.07	0.0	100.0
3	157.0325	3.49934	1000	10	0	33.32	0.0	0.0
4	146.0775	3.27706	1000	19	0	33.58	0.0	0.0
5	141.9075	3.35121	1000	12	0	37.07	0.0	0.0
6	133.5800	0.91741	1000	17	0	36.55	0.0	0.0
7	128.4500	1.49618	1000	1	0	36.55	0.0	100.0
8	128.8175	1.33784	1000	1	0	36.55	0.0	100.0
9	130.1775	0.81146	1000	1	0	36.55	0.0	100.0
10	134.7825	2.39492	1000	1	0	36.55	0.0	100.0
11	134.1375	0.84381	1000	1	0	36.55	0.0	100.0
12	131.6375	0.66803	1000	1	0	36.55	0.0	100.0
13	132.4000	1.06478	1000	1	0	36.55	0.0	100.0
14	142.1350	0.97228	1000	1	0	36.55	0.0	100.0
15	146.7525	1.94975	1000	1	0	36.55	0.0	100.0
16	136.6725	0.93767	1000	1	0	36.55	0.0	100.0
17	155.1350	26.08855	1000	1	0	36.55	0.0	100.0
18	142.0975	9.97840	1000	1	0	36.55	0.0	100.0
19	140.4800	2.28855	1000	1	0	36.55	0.0	100.0
20	141.2025	3.00960	1000	1	0	36.55	0.0	100.0
21	140.5850	1.71415	1000	1	0	36.55	0.0	100.0
22	188.3225	29.41895	1000	1	0	36.55	0.0	100.0
23	156.8025	13.63338	1000	1	0	36.55	0.0	100.0
24	147.7550	8.32924	1000	1	0	36.55	0.0	100.0
25	146.3700	7.41211	1000	1	0	36.55	0.0	100.0
26	156.1775	7.05514	1000	1	0	36.81	0.0	100.0

Table G.31: PBT_CG_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	341.5625	4.64457	1000	1	0	36.55	0.0	100.0
2	206.7100	4.73570	1000	2	0	75.75	0.0	100.0
3	167.1725	2.30842	1000	10	0	33.32	0.0	0.0
4	159.1100	3.29473	1000	19	0	33.58	0.0	0.0
5	148.7450	4.67506	1000	12	0	37.07	0.0	0.0
6	140.5475	1.19160	1000	17	0	36.55	0.0	0.0
7	137.8675	1.16671	1000	1	0	36.55	0.0	100.0
8	131.6925	1.54918	1000	1	0	36.55	0.0	100.0
9	136.5275	1.03485	1000	1	0	36.55	0.0	100.0
10	137.8600	1.18821	1000	1	0	36.55	0.0	100.0
11	138.5775	0.69229	1000	1	0	36.55	0.0	100.0
12	134.7600	0.23292	1000	1	0	36.55	0.0	100.0
13	138.4400	2.65950	1000	1	0	36.55	0.0	100.0
14	146.9125	1.11925	1000	1	0	36.55	0.0	100.0
15	151.5600	1.90427	1000	1	0	36.55	0.0	100.0
16	141.6525	1.11327	1000	1	0	36.55	0.0	100.0
17	145.7200	1.52727	1000	1	0	36.55	0.0	100.0
18	141.2025	1.43079	1000	1	0	36.55	0.0	100.0
19	142.6700	1.32742	1000	1	0	36.55	0.0	100.0
20	142.5100	1.19262	1000	1	0	36.55	0.0	100.0
21	144.7450	2.13439	1000	1	0	36.55	0.0	100.0
22	207.0575	1.46162	1000	1	0	36.55	0.0	100.0
23	166.2350	0.50047	1000	1	0	36.55	0.0	100.0
24	154.6375	0.48535	1000	1	0	36.55	0.0	100.0
25	153.1925	0.96456	1000	1	0	36.55	0.0	100.0
26	167.0075	3.53745	1000	1	0	36.81	0.0	100.0

Table G.32: PBT_CG_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	316.6475	0.49398	1000	1	0	36.55	0.00	100.0
2	190.4025	6.83342	1000	2	0	66.08	16.75	100.0
3	158.7525	4.15900	1000	10	0	33.32	0.00	0.0
4	146.3175	2.66799	1000	19	0	33.58	0.00	0.0
5	141.1550	0.89882	1000	12	0	37.07	0.00	0.0
6	134.2775	1.21613	1000	17	0	36.55	0.00	0.0
7	129.4275	0.66815	1000	1	0	36.55	0.00	100.0
8	130.1625	1.96794	1000	1	0	36.55	0.00	100.0
9	130.0575	1.35025	1000	1	0	36.55	0.00	100.0
10	133.9675	1.80335	1000	1	0	36.55	0.00	100.0
11	134.1650	1.33012	1000	1	0	36.55	0.00	100.0
12	133.1125	1.21947	1000	1	0	36.55	0.00	100.0
13	132.7325	1.44030	1000	1	0	36.55	0.00	100.0
14	143.2775	1.91080	1000	1	0	36.55	0.00	100.0
15	148.0800	1.29362	1000	1	0	36.55	0.00	100.0
16	138.0600	0.66645	1000	1	0	36.55	0.00	100.0
17	140.8700	1.63641	1000	1	0	36.55	0.00	100.0
18	136.1250	0.16194	1000	1	0	36.55	0.00	100.0
19	138.5400	0.73529	1000	1	0	36.55	0.00	100.0
20	140.0200	1.99855	1000	1	0	36.55	0.00	100.0
21	140.8700	0.98372	1000	1	0	36.55	0.00	100.0
22	211.6150	7.81375	1000	1	0	36.55	0.00	100.0
23	165.1800	3.10657	1000	1	0	36.55	0.00	100.0
24	149.1475	0.34974	1000	1	0	36.55	0.00	100.0
25	147.7975	1.52231	1000	1	0	36.55	0.00	100.0
26	159.4700	3.82805	1000	1	0	36.81	0.00	100.0

Table G.33: PBT_CG_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	336.9825	7.27568	1000	1	0	36.55	0.0	100.0
2	209.5600	5.02290	1000	2	0	75.75	0.0	100.0
3	170.1200	5.79567	1000	10	0	33.32	0.0	0.0
4	157.3650	4.39984	1000	19	0	33.58	0.0	0.0
5	151.0050	5.15336	1000	12	0	37.07	0.0	0.0
6	142.3525	4.41607	1000	17	0	36.55	0.0	0.0
7	139.5600	2.90103	1000	1	0	36.55	0.0	100.0
8	134.3100	2.79125	1000	1	0	36.55	0.0	100.0
9	136.0675	0.94373	1000	1	0	36.55	0.0	100.0
10	141.1775	2.55093	1000	1	0	36.55	0.0	100.0
11	141.8525	2.19267	1000	1	0	36.55	0.0	100.0
12	138.1600	1.06932	1000	1	0	36.55	0.0	100.0
13	138.7375	3.82647	1000	1	0	36.55	0.0	100.0
14	148.5575	0.64755	1000	1	0	36.55	0.0	100.0
15	153.0625	0.66357	1000	1	0	36.55	0.0	100.0
16	142.5725	1.11965	1000	1	0	36.55	0.0	100.0
17	146.2525	0.20717	1000	1	0	36.55	0.0	100.0
18	142.3300	0.72121	1000	1	0	36.55	0.0	100.0
19	144.0625	1.20190	1000	1	0	36.55	0.0	100.0
20	143.5375	1.96614	1000	1	0	36.55	0.0	100.0
21	146.6175	1.90363	1000	1	0	36.55	0.0	100.0
22	216.8700	8.17595	1000	1	0	36.55	0.0	100.0
23	172.4100	8.57277	1000	1	0	36.55	0.0	100.0
24	156.5250	2.68068	1000	1	0	36.55	0.0	100.0
25	154.7925	2.04953	1000	1	0	36.55	0.0	100.0
26	164.8850	1.69867	1000	1	0	36.81	0.0	100.0

Table G.34: PBT_CG_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	318.5175	2.54583	1000	1	0	36.55	0.0	100.0
2	185.2475	2.71277	1000	1	0	37.07	0.0	100.0
3	157.8325	3.98105	1000	10	0	33.32	0.0	0.0
4	144.5550	2.50959	1000	19	0	33.58	0.0	0.0
5	143.8250	4.00945	1000	12	0	37.07	0.0	0.0
6	134.8375	2.18306	1000	17	0	36.55	0.0	0.0
7	129.8950	1.06397	1000	1	0	36.55	0.0	100.0
8	128.7025	2.20349	1000	1	0	36.55	0.0	100.0
9	131.4300	2.08920	1000	1	0	36.55	0.0	100.0
10	135.6900	4.58719	1000	1	0	36.55	0.0	100.0
11	135.9975	2.01616	1000	1	0	36.55	0.0	100.0
12	132.8775	1.79394	1000	1	0	36.55	0.0	100.0
13	133.5950	1.02395	1000	1	0	36.55	0.0	100.0
14	145.4250	5.02008	1000	1	0	36.55	0.0	100.0
15	150.3725	3.14152	1000	1	0	36.55	0.0	100.0
16	138.0075	1.16538	1000	1	0	36.55	0.0	100.0
17	141.9800	0.68589	1000	1	0	36.55	0.0	100.0
18	135.7400	0.49614	1000	1	0	36.55	0.0	100.0
19	140.1225	1.65284	1000	1	0	36.55	0.0	100.0
20	142.1525	1.41052	1000	1	0	36.55	0.0	100.0
21	142.4125	0.63231	1000	1	0	36.55	0.0	100.0
22	208.1150	8.61255	1000	1	0	36.55	0.0	100.0
23	162.4575	0.87913	1000	1	0	36.55	0.0	100.0
24	150.4450	1.38466	1000	1	0	36.55	0.0	100.0
25	148.1225	2.32643	1000	1	0	36.55	0.0	100.0
26	156.2400	0.57736	1000	1	0	36.81	0.0	100.0

Table G.35: PBT_CG_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	331.1950	2.30385	1000	1	0	36.55	0.0	100.0
2	207.1625	1.60233	1000	2	0	75.75	0.0	100.0
3	166.6225	2.77687	1000	10	0	33.32	0.0	0.0
4	156.4075	2.24724	1000	19	0	33.58	0.0	0.0
5	148.4750	3.12821	1000	12	0	37.07	0.0	0.0
6	140.8275	1.11430	1000	17	0	36.55	0.0	0.0
7	133.8175	0.73121	1000	1	0	36.55	0.0	100.0
8	135.2050	0.98817	1000	1	0	36.55	0.0	100.0
9	137.2175	1.27121	1000	1	0	36.55	0.0	100.0
10	142.3375	2.53746	1000	1	0	36.55	0.0	100.0
11	138.5500	0.88082	1000	1	0	36.55	0.0	100.0
12	135.7925	1.11735	1000	1	0	36.55	0.0	100.0
13	139.5625	1.42577	1000	1	0	36.55	0.0	100.0
14	159.3900	6.92631	1000	1	0	36.55	0.0	100.0
15	154.0250	2.41012	1000	1	0	36.55	0.0	100.0
16	144.0075	2.71344	1000	1	0	36.55	0.0	100.0
17	147.0450	1.57163	1000	1	0	36.55	0.0	100.0
18	142.0275	3.73237	1000	1	0	36.55	0.0	100.0
19	146.0900	2.37548	1000	1	0	36.55	0.0	100.0
20	145.1625	2.08143	1000	1	0	36.55	0.0	100.0
21	144.5500	2.19546	1000	1	0	36.55	0.0	100.0
22	223.1875	12.56934	1000	1	0	36.55	0.0	100.0
23	170.4825	3.72141	1000	1	0	36.55	0.0	100.0
24	155.4675	1.44507	1000	1	0	36.55	0.0	100.0
25	154.9975	2.68270	1000	1	0	36.55	0.0	100.0
26	167.2200	4.59999	1000	1	0	36.81	0.0	100.0

Table G.36: PBT_CG_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	334.0550	6.12885	1000	1	0	36.55	0.0	100.0
2	207.5075	3.94220	1000	2	0	75.75	0.0	100.0
3	167.1450	3.10353	1000	10	0	33.32	0.0	0.0
4	156.6625	3.11238	1000	19	0	33.58	0.0	0.0
5	149.8125	3.13766	1000	12	0	37.07	0.0	0.0
6	139.4050	1.47937	1000	17	0	36.55	0.0	0.0
7	134.5175	0.46251	1000	1	0	36.55	0.0	100.0
8	133.4600	0.39718	1000	1	0	36.55	0.0	100.0
9	136.6300	1.48039	1000	1	0	36.55	0.0	100.0
10	140.3000	1.61513	1000	1	0	36.55	0.0	100.0
11	139.2575	1.28962	1000	1	0	36.55	0.0	100.0
12	138.6925	0.87010	1000	1	0	36.55	0.0	100.0
13	137.8500	1.87751	1000	1	0	36.55	0.0	100.0
14	153.4800	0.42579	1000	1	0	36.55	0.0	100.0
15	156.6175	8.10653	1000	1	0	36.55	0.0	100.0
16	144.9400	4.23208	1000	1	0	36.55	0.0	100.0
17	147.9700	2.22042	1000	1	0	36.55	0.0	100.0
18	141.3150	1.10348	1000	1	0	36.55	0.0	100.0
19	143.7050	0.87274	1000	1	0	36.55	0.0	100.0
20	144.5250	1.37645	1000	1	0	36.55	0.0	100.0
21	149.0000	4.79463	1000	1	0	36.55	0.0	100.0
22	223.7225	11.90575	1000	1	0	36.55	0.0	100.0
23	169.3125	2.52663	1000	1	0	36.55	0.0	100.0
24	155.4875	1.26209	1000	1	0	36.55	0.0	100.0
25	156.0600	3.95926	1000	1	0	36.55	0.0	100.0
26	167.3950	1.49984	1000	1	0	36.81	0.0	100.0

G.3 M2

Table G.37: M2 - Search strategies results.

Strategy	Min Mean Time (ms)	Std Min Mean Time (ms)	Max Mean Time (ms)	Std Max Mean Time (ms)	Min Sample Size	Max Sample Size	Min Mean Test Cases	Std Min Mean Test Cases	Max Mean Test Cases	Std Max Mean Test Cases	Min Mean Coverage (%)	Std Min Mean Coverage (%)	Max Mean Coverage (%)	Std Max Mean Coverage (%)	Bug Found
PBT_R_S4	19.10125	1.20241	19.10125	1.20241	1000	1000	3753	830	3753	830	0.00	0.00	0.00	0.00	True
PBT_R_S1	24.19800	1.74349	24.19800	1.74349	1000	1000	7252	1089	7252	1089	0.00	0.00	0.00	0.00	True
PBT_R_S8	44.27325	0.66840	44.27325	0.66840	1000	1000	6002	0	6002	0	0.00	0.00	0.00	0.00	True
PBT_CG_S4	270.45000	36.35098	270.45000	36.35098	1000	1000	1315	1082	1315	1082	76.53	11.18	76.53	11.18	True
PBT_CG_S6	226.19000	9.69410	226.19000	9.69410	1000	1000	251	0	251	0	76.27	0.00	76.27	0.00	True
PBT_CG_S3	328.21250	21.05213	328.21250	21.05213	1000	1000	2064	369	2064	369	84.09	1.54	84.09	1.54	True
PBT_R_S3	47.94650	3.23166	47.94650	3.23166	1000	1000	8502	1658	8502	1658	0.00	0.00	0.00	0.00	True
PBT_R_S5	45.16400	0.13733	45.16400	0.13733	1000	1000	6002	0	6002	0	0.00	0.00	0.00	0.00	True
PBT_R_S7	45.04100	0.60429	45.04100	0.60429	1000	1000	7002	1000	7002	1000	0.00	0.00	0.00	0.00	True
PBT_CG_S5	297.20750	9.51023	297.20750	9.51023	1000	1000	1502	0	1502	0	81.74	0.00	81.74	0.00	True
PBT_CG_S7	320.14500	19.52241	320.14500	19.52241	1000	1000	2002	306	2002	306	83.83	1.28	83.83	1.28	True
PBT_CG_S1	255.68750	37.51412	255.68750	37.51412	1000	1000	878	974	878	974	78.94	4.01	78.94	4.01	True
PBT_CG_S8	296.14500	5.20117	296.14500	5.20117	1000	1000	1502	0	1502	0	81.74	0.00	81.74	0.00	True
PBT_CG_S2	230.40000	9.87711	230.40000	9.87711	1000	1000	251	0	251	0	76.27	0.00	76.27	0.00	True
PBT_R_S6	13.91175	0.22506	13.91175	0.22506	1000	1000	1001	0	1001	0	0.00	0.00	0.00	0.00	True
PBT_R_S2	14.06675	0.36500	14.06675	0.36500	1000	1000	1001	0	1001	0	0.00	0.00	0.00	0.00	True

Table G.38: M2 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	4.7177	0.7089	-23.3520	32.7875	False
Random withPairWiseTesting	Random withoutPairWiseTesting	2.2620	0.8563	-26.1108	30.6348	False
Random withConstantExtraction	Random withoutConstantExtraction	-27.7868	0.0010	-34.0829	-21.4906	True
Coverage-Guided withBias	Coverage-Guided withoutBias	31.1381	0.2879	-34.2199	96.4962	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-0.3556	0.9000	-72.7372	72.0260	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-64.7456	0.0028	-97.2424	-32.2488	True
Coverage-Guided	Random	-246.3419	0.0010	-277.9004	-214.7834	True

Table G.39: M2 - Tukey's test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	3125.75	0.1267	-1189.2383	7440.7383	False
Random withPairWiseTesting	Random withoutPairWiseTesting	1249.75	0.5866	-3928.1157	6427.6157	False
Random withConstantExtraction	Random withoutConstantExtraction	-3625.25	0.0636	-7530.8989	280.3989	False
Coverage-Guided withBias	Coverage-Guided withoutBias	688.25	0.1853	-437.3826	1813.8826	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-93.75	0.8705	-1409.4132	1221.9132	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-1093.75	0.0111	-1832.6464	-354.8536	True
Coverage-Guided	Random	3843.75	0.0024	1617.3262	6070.1738	True

Table G.40: M2 - Tukey's test results concerning the *Coverage* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Coverage-Guided withBias	Coverage-Guided withoutBias	1.8425	0.4792	-4.1349	7.8199	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	0.6675	0.8103	-5.5512	6.8862	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-5.8475	0.0010	-8.0826	-3.6124	True

G.3.1 Search Strategies

Table G.41: PBT_R_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	63.15725	2.87950	1000	31000	0	0.0	0.0	0.0
2	24.19800	1.74349	1000	7252	1089	0.0	0.0	100.0
3	1.76639	1.65298	1000	10	0	0.0	0.0	0.0
4	0.25180	0.01307	1000	19	0	0.0	0.0	0.0
5	0.20809	0.00616	1000	12	0	0.0	0.0	0.0
6	0.23031	0.01464	1000	17	0	0.0	0.0	0.0
7	0.20098	0.00457	1000	12	0	0.0	0.0	0.0
8	0.21896	0.00589	1000	19	0	0.0	0.0	0.0
9	0.19989	0.01065	1000	12	0	0.0	0.0	0.0
10	0.22522	0.01390	1000	19	0	0.0	0.0	0.0
11	0.23162	0.02603	1000	11	0	0.0	0.0	0.0
12	0.22291	0.00524	1000	20	0	0.0	0.0	0.0
13	0.19430	0.00301	1000	11	0	0.0	0.0	0.0
14	0.22535	0.00380	1000	20	0	0.0	0.0	0.0
15	0.19341	0.00552	1000	10	0	0.0	0.0	0.0
16	0.21831	0.00806	1000	22	0	0.0	0.0	0.0
17	0.18657	0.00369	1000	9	0	0.0	0.0	0.0
18	0.22026	0.00672	1000	22	0	0.0	0.0	0.0
19	0.18910	0.00727	1000	9	0	0.0	0.0	0.0
20	0.22104	0.01183	1000	22	0	0.0	0.0	0.0
21	0.17683	0.00319	1000	8	0	0.0	0.0	0.0
22	0.30744	0.16029	1000	22	0	0.0	0.0	0.0
23	0.18214	0.00479	1000	9	0	0.0	0.0	0.0
24	0.21422	0.01211	1000	21	0	0.0	0.0	0.0
25	0.26697	0.14945	1000	9	0	0.0	0.0	0.0
26	0.29960	0.14462	1000	21	0	0.0	0.0	0.0

Table G.42: PBT_R_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	62.82825	1.86304	1000	31000	0	0.0	0.0	0.0
2	14.06675	0.36500	1000	1001	0	0.0	0.0	100.0
3	0.37748	0.01058	1000	10	0	0.0	0.0	0.0
4	0.31614	0.03065	1000	19	0	0.0	0.0	0.0
5	0.25885	0.02295	1000	12	0	0.0	0.0	0.0
6	0.26758	0.01646	1000	17	0	0.0	0.0	0.0
7	0.24987	0.02574	1000	12	0	0.0	0.0	0.0
8	0.26861	0.01610	1000	19	0	0.0	0.0	0.0
9	0.24903	0.01750	1000	12	0	0.0	0.0	0.0
10	0.71767	0.05919	1000	19	0	0.0	0.0	0.0
11	0.25084	0.01652	1000	11	0	0.0	0.0	0.0
12	0.28689	0.02076	1000	20	0	0.0	0.0	0.0
13	0.24987	0.01584	1000	11	0	0.0	0.0	0.0
14	0.27568	0.02043	1000	20	0	0.0	0.0	0.0
15	0.24261	0.01855	1000	10	0	0.0	0.0	0.0
16	0.30993	0.03911	1000	22	0	0.0	0.0	0.0
17	0.22925	0.01783	1000	9	0	0.0	0.0	0.0
18	0.24890	0.02086	1000	22	0	0.0	0.0	0.0
19	0.21433	0.01723	1000	9	0	0.0	0.0	0.0
20	0.23453	0.00735	1000	22	0	0.0	0.0	0.0
21	0.22136	0.04837	1000	8	0	0.0	0.0	0.0
22	0.26264	0.06901	1000	22	0	0.0	0.0	0.0
23	0.22570	0.05608	1000	9	0	0.0	0.0	0.0
24	0.24127	0.06327	1000	21	0	0.0	0.0	0.0
25	0.35530	0.30000	1000	9	0	0.0	0.0	0.0
26	0.39408	0.15262	1000	21	0	0.0	0.0	0.0

Table G.43: PBT_R_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	78.37400	0.72537	1000	31000	0	0.0	0.0	0.0
2	47.94650	3.23166	1000	8502	1658	0.0	0.0	100.0
3	11.82275	1.71062	1000	10	0	0.0	0.0	0.0
4	6.90425	0.08932	1000	19	0	0.0	0.0	0.0
5	5.56265	0.11882	1000	12	0	0.0	0.0	0.0
6	5.65855	0.06937	1000	17	0	0.0	0.0	0.0
7	10.00502	0.06524	1000	12	0	0.0	0.0	0.0
8	5.56672	0.11533	1000	19	0	0.0	0.0	0.0
9	5.75445	0.03625	1000	12	0	0.0	0.0	0.0
10	5.50985	0.05291	1000	19	0	0.0	0.0	0.0
11	5.35450	0.03380	1000	11	0	0.0	0.0	0.0
12	5.31283	0.09787	1000	20	0	0.0	0.0	0.0
13	5.37805	0.14935	1000	11	0	0.0	0.0	0.0
14	5.55185	0.07457	1000	20	0	0.0	0.0	0.0
15	5.23275	0.05687	1000	10	0	0.0	0.0	0.0
16	5.21735	0.06804	1000	22	0	0.0	0.0	0.0
17	5.25235	0.12701	1000	9	0	0.0	0.0	0.0
18	5.20885	0.02485	1000	22	0	0.0	0.0	0.0
19	5.23880	0.11705	1000	9	0	0.0	0.0	0.0
20	5.28107	0.07345	1000	22	0	0.0	0.0	0.0
21	5.20185	0.09037	1000	8	0	0.0	0.0	0.0
22	5.22752	0.09777	1000	22	0	0.0	0.0	0.0
23	5.17192	0.08012	1000	9	0	0.0	0.0	0.0
24	5.51898	0.50674	1000	21	0	0.0	0.0	0.0
25	5.75282	0.51271	1000	9	0	0.0	0.0	0.0
26	5.55375	0.42831	1000	21	0	0.0	0.0	0.0

Table G.44: PBT_R_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	60.17725	0.31319	1000	31000	0	0.0	0.0	0.0
2	19.10125	1.20241	1000	3753	830	0.0	0.0	100.0
3	0.78680	0.01186	1000	10	0	0.0	0.0	0.0
4	0.29477	0.02038	1000	19	0	0.0	0.0	0.0
5	0.21957	0.01729	1000	12	0	0.0	0.0	0.0
6	0.22793	0.02570	1000	17	0	0.0	0.0	0.0
7	0.21584	0.02098	1000	12	0	0.0	0.0	0.0
8	0.23864	0.02377	1000	19	0	0.0	0.0	0.0
9	0.21190	0.02501	1000	12	0	0.0	0.0	0.0
10	0.23083	0.03407	1000	19	0	0.0	0.0	0.0
11	0.21024	0.02611	1000	11	0	0.0	0.0	0.0
12	0.23008	0.02212	1000	20	0	0.0	0.0	0.0
13	0.21159	0.02568	1000	11	0	0.0	0.0	0.0
14	0.29560	0.03180	1000	20	0	0.0	0.0	0.0
15	0.21071	0.02313	1000	10	0	0.0	0.0	0.0
16	0.24612	0.02708	1000	22	0	0.0	0.0	0.0
17	0.20538	0.02338	1000	9	0	0.0	0.0	0.0
18	0.23700	0.02582	1000	22	0	0.0	0.0	0.0
19	0.19961	0.01222	1000	9	0	0.0	0.0	0.0
20	0.22314	0.01295	1000	22	0	0.0	0.0	0.0
21	0.18882	0.01247	1000	8	0	0.0	0.0	0.0
22	0.22836	0.00355	1000	22	0	0.0	0.0	0.0
23	0.18146	0.00483	1000	9	0	0.0	0.0	0.0
24	0.29217	0.14513	1000	21	0	0.0	0.0	0.0
25	0.26217	0.14571	1000	9	0	0.0	0.0	0.0
26	0.29725	0.14771	1000	21	0	0.0	0.0	0.0

Table G.45: PBT_R_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	80.56950	0.47946	1000	31000	0	0.0	0.0	0.0
2	45.16400	0.13733	1000	6002	0	0.0	0.0	100.0
3	10.62275	0.11810	1000	10	0	0.0	0.0	0.0
4	6.81020	0.05709	1000	19	0	0.0	0.0	0.0
5	5.50220	0.08331	1000	12	0	0.0	0.0	0.0
6	5.69205	0.13764	1000	17	0	0.0	0.0	0.0
7	10.05300	0.13770	1000	12	0	0.0	0.0	0.0
8	5.46405	0.13782	1000	19	0	0.0	0.0	0.0
9	5.75485	0.06186	1000	12	0	0.0	0.0	0.0
10	5.41240	0.06040	1000	19	0	0.0	0.0	0.0
11	5.37358	0.09202	1000	11	0	0.0	0.0	0.0
12	5.57355	0.05628	1000	20	0	0.0	0.0	0.0
13	5.36850	0.05781	1000	11	0	0.0	0.0	0.0
14	5.58673	0.09318	1000	20	0	0.0	0.0	0.0
15	5.38878	0.10112	1000	10	0	0.0	0.0	0.0
16	5.38113	0.16037	1000	22	0	0.0	0.0	0.0
17	5.28418	0.11285	1000	9	0	0.0	0.0	0.0
18	5.35040	0.11083	1000	22	0	0.0	0.0	0.0
19	5.29002	0.10097	1000	9	0	0.0	0.0	0.0
20	5.27580	0.09158	1000	22	0	0.0	0.0	0.0
21	5.22050	0.11632	1000	8	0	0.0	0.0	0.0
22	5.31958	0.09809	1000	22	0	0.0	0.0	0.0
23	5.19840	0.11934	1000	9	0	0.0	0.0	0.0
24	5.39280	0.10300	1000	21	0	0.0	0.0	0.0
25	5.50380	0.11037	1000	9	0	0.0	0.0	0.0
26	5.34753	0.13442	1000	21	0	0.0	0.0	0.0

Table G.46: PBT_R_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	61.92700	0.42682	1000	31000	0	0.0	0.0	0.0
2	13.91175	0.22506	1000	1001	0	0.0	0.0	100.0
3	0.37692	0.00651	1000	10	0	0.0	0.0	0.0
4	0.31324	0.02539	1000	19	0	0.0	0.0	0.0
5	0.24199	0.00306	1000	12	0	0.0	0.0	0.0
6	0.24869	0.01417	1000	17	0	0.0	0.0	0.0
7	0.24760	0.03546	1000	12	0	0.0	0.0	0.0
8	0.24391	0.01077	1000	19	0	0.0	0.0	0.0
9	0.23017	0.01214	1000	12	0	0.0	0.0	0.0
10	0.66122	0.03582	1000	19	0	0.0	0.0	0.0
11	0.23542	0.01328	1000	11	0	0.0	0.0	0.0
12	0.25038	0.00911	1000	20	0	0.0	0.0	0.0
13	0.22434	0.01034	1000	11	0	0.0	0.0	0.0
14	0.25754	0.00578	1000	20	0	0.0	0.0	0.0
15	0.22088	0.00821	1000	10	0	0.0	0.0	0.0
16	0.28598	0.03224	1000	22	0	0.0	0.0	0.0
17	0.23523	0.02686	1000	9	0	0.0	0.0	0.0
18	0.24722	0.00457	1000	22	0	0.0	0.0	0.0
19	0.21092	0.00606	1000	9	0	0.0	0.0	0.0
20	0.23676	0.00941	1000	22	0	0.0	0.0	0.0
21	0.19649	0.00494	1000	8	0	0.0	0.0	0.0
22	0.22519	0.00817	1000	22	0	0.0	0.0	0.0
23	0.19244	0.00343	1000	9	0	0.0	0.0	0.0
24	0.31638	0.15040	1000	21	0	0.0	0.0	0.0
25	0.19591	0.00627	1000	9	0	0.0	0.0	0.0
26	0.32664	0.15829	1000	21	0	0.0	0.0	0.0

Table G.47: PBT_R_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	79.47875	0.62391	1000	31000	0	0.0	0.0	0.0
2	45.04100	0.60429	1000	7002	1000	0.0	0.0	100.0
3	10.25512	0.27136	1000	10	0	0.0	0.0	0.0
4	6.90710	0.12750	1000	19	0	0.0	0.0	0.0
5	5.56417	0.10801	1000	12	0	0.0	0.0	0.0
6	5.67087	0.11924	1000	17	0	0.0	0.0	0.0
7	5.63975	0.14649	1000	12	0	0.0	0.0	0.0
8	5.46280	0.03904	1000	19	0	0.0	0.0	0.0
9	5.79212	0.07332	1000	12	0	0.0	0.0	0.0
10	5.46985	0.12311	1000	19	0	0.0	0.0	0.0
11	5.37327	0.13102	1000	11	0	0.0	0.0	0.0
12	5.38925	0.11756	1000	20	0	0.0	0.0	0.0
13	5.34943	0.08286	1000	11	0	0.0	0.0	0.0
14	9.93450	0.06848	1000	20	0	0.0	0.0	0.0
15	5.27077	0.08299	1000	10	0	0.0	0.0	0.0
16	5.33268	0.02809	1000	22	0	0.0	0.0	0.0
17	5.24095	0.04554	1000	9	0	0.0	0.0	0.0
18	5.34267	0.16874	1000	22	0	0.0	0.0	0.0
19	5.19580	0.02682	1000	9	0	0.0	0.0	0.0
20	5.23452	0.06553	1000	22	0	0.0	0.0	0.0
21	5.15642	0.08894	1000	8	0	0.0	0.0	0.0
22	5.27628	0.07841	1000	22	0	0.0	0.0	0.0
23	5.24715	0.14614	1000	9	0	0.0	0.0	0.0
24	5.16245	0.11300	1000	21	0	0.0	0.0	0.0
25	5.23438	0.08618	1000	9	0	0.0	0.0	0.0
26	5.18307	0.04749	1000	21	0	0.0	0.0	0.0

Table G.48: PBT_R_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	79.80925	0.27408	1000	31000	0	0.0	0.0	0.0
2	44.27325	0.66840	1000	6002	0	0.0	0.0	100.0
3	10.50375	0.07785	1000	10	0	0.0	0.0	0.0
4	6.88463	0.09888	1000	19	0	0.0	0.0	0.0
5	5.56403	0.06464	1000	12	0	0.0	0.0	0.0
6	5.61967	0.06355	1000	17	0	0.0	0.0	0.0
7	5.52698	0.15720	1000	12	0	0.0	0.0	0.0
8	5.49910	0.11378	1000	19	0	0.0	0.0	0.0
9	5.79105	0.09167	1000	12	0	0.0	0.0	0.0
10	5.36675	0.15114	1000	19	0	0.0	0.0	0.0
11	5.34307	0.08160	1000	11	0	0.0	0.0	0.0
12	5.40465	0.08361	1000	20	0	0.0	0.0	0.0
13	5.27337	0.14873	1000	11	0	0.0	0.0	0.0
14	9.90760	0.09565	1000	20	0	0.0	0.0	0.0
15	5.38603	0.07563	1000	10	0	0.0	0.0	0.0
16	5.30980	0.09436	1000	22	0	0.0	0.0	0.0
17	5.25860	0.06478	1000	9	0	0.0	0.0	0.0
18	5.27202	0.09775	1000	22	0	0.0	0.0	0.0
19	5.23358	0.06583	1000	9	0	0.0	0.0	0.0
20	5.28142	0.06245	1000	22	0	0.0	0.0	0.0
21	5.18013	0.16169	1000	8	0	0.0	0.0	0.0
22	5.15590	0.08974	1000	22	0	0.0	0.0	0.0
23	5.21400	0.15917	1000	9	0	0.0	0.0	0.0
24	5.26443	0.13696	1000	21	0	0.0	0.0	0.0
25	5.21070	0.06381	1000	9	0	0.0	0.0	0.0
26	5.33788	0.19810	1000	21	0	0.0	0.0	0.0

Table G.49: PBT_CG_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22300.0000	62.77738	1000	1000000	0	73.27	0.00	0.0
2	255.6875	37.51412	1000	878	974	78.94	4.01	100.0
3	168.8650	4.41657	1000	10	0	33.32	0.00	0.0
4	154.3575	4.64072	1000	19	0	33.58	0.00	0.0
5	148.5450	2.59983	1000	12	0	37.33	0.00	0.0
6	142.0925	1.45881	1000	17	0	37.07	0.00	0.0
7	136.9575	1.32959	1000	12	0	40.82	0.00	0.0
8	148.2325	23.13580	1000	19	0	40.56	0.00	0.0
9	142.8150	13.65944	1000	12	0	44.32	0.00	0.0
10	142.5825	3.06743	1000	19	0	44.05	0.00	0.0
11	139.5375	3.13111	1000	11	0	47.81	0.00	0.0
12	139.1675	2.15800	1000	20	0	47.55	0.00	0.0
13	140.4375	3.35022	1000	11	0	51.30	0.00	0.0
14	150.3875	2.57230	1000	20	0	51.04	0.00	0.0
15	153.8375	5.02089	1000	10	0	54.80	0.00	0.0
16	141.7875	7.24773	1000	22	0	54.54	0.00	0.0
17	144.6250	5.47071	1000	9	0	58.29	0.00	0.0
18	141.2625	5.50111	1000	22	0	58.03	0.00	0.0
19	140.8700	6.95175	1000	9	0	61.78	0.00	0.0
20	146.4050	8.17943	1000	22	0	61.52	0.00	0.0
21	146.3375	6.43507	1000	8	0	65.28	0.00	0.0
22	147.7375	5.81204	1000	22	0	65.01	0.00	0.0
23	149.0300	6.31865	1000	9	0	68.77	0.00	0.0
24	149.6250	6.43747	1000	21	0	68.50	0.00	0.0
25	155.4175	10.11477	1000	9	0	72.26	0.00	0.0
26	163.1300	4.37340	1000	21	0	72.26	0.00	0.0

Table G.50: PBT_CG_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22305.2500	94.66355	1000	1000000	0	73.27	0.0	0.0
2	230.4000	9.87711	1000	251	0	76.27	0.0	100.0
3	174.8275	5.54147	1000	10	0	33.32	0.0	0.0
4	155.6600	6.61360	1000	19	0	33.58	0.0	0.0
5	151.4275	5.20646	1000	12	0	37.33	0.0	0.0
6	142.7850	2.43439	1000	17	0	37.07	0.0	0.0
7	136.9125	0.91330	1000	12	0	40.82	0.0	0.0
8	136.3400	1.81233	1000	19	0	40.56	0.0	0.0
9	135.8700	0.64452	1000	12	0	44.32	0.0	0.0
10	141.1400	1.13369	1000	19	0	44.05	0.0	0.0
11	141.7850	1.33230	1000	11	0	47.81	0.0	0.0
12	139.2300	1.69984	1000	20	0	47.55	0.0	0.0
13	138.7875	1.66082	1000	11	0	51.30	0.0	0.0
14	149.9525	1.82514	1000	20	0	51.04	0.0	0.0
15	154.0925	1.86244	1000	10	0	54.80	0.0	0.0
16	145.7325	1.80110	1000	22	0	54.54	0.0	0.0
17	147.2825	2.41056	1000	9	0	58.29	0.0	0.0
18	143.7050	1.66851	1000	22	0	58.03	0.0	0.0
19	146.3675	4.47824	1000	9	0	61.78	0.0	0.0
20	151.5575	2.44909	1000	22	0	61.52	0.0	0.0
21	148.9100	1.83457	1000	8	0	65.28	0.0	0.0
22	151.2100	1.56632	1000	22	0	65.01	0.0	0.0
23	152.7775	2.31523	1000	9	0	68.77	0.0	0.0
24	155.9450	2.40003	1000	21	0	68.50	0.0	0.0
25	156.9975	2.21718	1000	9	0	72.26	0.0	0.0
26	166.7800	3.56553	1000	21	0	72.26	0.0	0.0

Table G.51: PBT_CG_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22288.7500	134.81353	1000	1000000	0	73.27	0.00	0.0
2	328.2125	21.05213	1000	2064	369	84.09	1.54	100.0
3	187.2325	5.04106	1000	10	0	33.32	0.00	0.0
4	166.5775	3.50442	1000	19	0	33.58	0.00	0.0
5	157.9700	2.84479	1000	12	0	37.33	0.00	0.0
6	149.9250	2.38370	1000	17	0	37.07	0.00	0.0
7	148.4100	1.12532	1000	12	0	40.82	0.00	0.0
8	140.3700	2.37147	1000	19	0	40.56	0.00	0.0
9	144.5575	2.42302	1000	12	0	44.32	0.00	0.0
10	148.3175	2.31860	1000	19	0	44.05	0.00	0.0
11	148.3650	2.43171	1000	11	0	47.81	0.00	0.0
12	147.4450	2.01577	1000	20	0	47.55	0.00	0.0
13	145.9725	1.06140	1000	11	0	51.30	0.00	0.0
14	155.4975	0.91459	1000	20	0	51.04	0.00	0.0
15	160.2825	1.37955	1000	10	0	54.80	0.00	0.0
16	150.7050	1.55173	1000	22	0	54.54	0.00	0.0
17	154.8325	1.23194	1000	9	0	58.29	0.00	0.0
18	149.6825	1.15055	1000	22	0	58.03	0.00	0.0
19	148.0075	1.29801	1000	9	0	61.78	0.00	0.0
20	156.3800	0.91113	1000	22	0	61.52	0.00	0.0
21	155.4900	3.49353	1000	8	0	65.28	0.00	0.0
22	157.1700	1.01151	1000	22	0	65.01	0.00	0.0
23	158.0475	0.90731	1000	9	0	68.77	0.00	0.0
24	160.1425	3.21028	1000	21	0	68.50	0.00	0.0
25	160.5225	1.61706	1000	9	0	72.26	0.00	0.0
26	173.3400	1.92682	1000	21	0	72.26	0.00	0.0

Table G.52: PBT_CG_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22276.7500	107.31117	1000	1000000	0	73.27	0.00	0.0
2	270.4500	36.35098	1000	1315	1082	76.53	11.18	100.0
3	169.5225	8.63398	1000	10	0	33.32	0.00	0.0
4	150.9775	5.29744	1000	19	0	33.58	0.00	0.0
5	163.5750	18.61649	1000	12	0	37.33	0.00	0.0
6	150.6950	6.94960	1000	17	0	37.07	0.00	0.0
7	136.9425	1.27058	1000	12	0	40.82	0.00	0.0
8	136.4175	3.67781	1000	19	0	40.56	0.00	0.0
9	136.2050	2.30229	1000	12	0	44.32	0.00	0.0
10	141.5325	5.70057	1000	19	0	44.05	0.00	0.0
11	143.0150	8.10652	1000	11	0	47.81	0.00	0.0
12	139.0675	6.07675	1000	20	0	47.55	0.00	0.0
13	138.3775	2.08676	1000	11	0	51.30	0.00	0.0
14	150.2550	0.72092	1000	20	0	51.04	0.00	0.0
15	153.6000	3.31989	1000	10	0	54.80	0.00	0.0
16	142.5225	4.71621	1000	22	0	54.54	0.00	0.0
17	145.3050	6.32615	1000	9	0	58.29	0.00	0.0
18	143.5650	6.28518	1000	22	0	58.03	0.00	0.0
19	140.5975	5.00323	1000	9	0	61.78	0.00	0.0
20	146.8500	6.93589	1000	22	0	61.52	0.00	0.0
21	145.5200	5.24260	1000	8	0	65.28	0.00	0.0
22	151.1225	7.14333	1000	22	0	65.01	0.00	0.0
23	151.1275	3.47243	1000	9	0	68.77	0.00	0.0
24	150.9450	4.81471	1000	21	0	68.50	0.00	0.0
25	153.0275	4.91252	1000	9	0	72.26	0.00	0.0
26	161.3625	3.71281	1000	21	0	72.26	0.00	0.0

Table G.53: PBT_CG_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22204.7500	122.25460	1000	1000000	0	73.27	0.0	0.0
2	297.2075	9.51023	1000	1502	0	81.74	0.0	100.0
3	185.6425	2.00017	1000	10	0	33.32	0.0	0.0
4	165.0550	3.79971	1000	19	0	33.58	0.0	0.0
5	156.3175	5.44008	1000	12	0	37.33	0.0	0.0
6	150.8650	3.59455	1000	17	0	37.07	0.0	0.0
7	150.5775	2.20900	1000	12	0	40.82	0.0	0.0
8	140.0575	0.46927	1000	19	0	40.56	0.0	0.0
9	144.2850	1.05599	1000	12	0	44.32	0.0	0.0
10	148.8050	2.24849	1000	19	0	44.05	0.0	0.0
11	150.7700	4.91036	1000	11	0	47.81	0.0	0.0
12	146.2050	3.58931	1000	20	0	47.55	0.0	0.0
13	150.0075	2.67041	1000	11	0	51.30	0.0	0.0
14	156.0950	0.78862	1000	20	0	51.04	0.0	0.0
15	159.2675	1.78414	1000	10	0	54.80	0.0	0.0
16	154.1775	5.11771	1000	22	0	54.54	0.0	0.0
17	155.3000	1.63918	1000	9	0	58.29	0.0	0.0
18	149.9050	1.60659	1000	22	0	58.03	0.0	0.0
19	149.0350	2.47492	1000	9	0	61.78	0.0	0.0
20	158.7125	2.71364	1000	22	0	61.52	0.0	0.0
21	155.7425	2.51338	1000	8	0	65.28	0.0	0.0
22	157.8575	1.12420	1000	22	0	65.01	0.0	0.0
23	157.5725	0.12833	1000	9	0	68.77	0.0	0.0
24	158.4225	0.81106	1000	21	0	68.50	0.0	0.0
25	160.3750	0.73490	1000	9	0	72.26	0.0	0.0
26	169.3925	1.02814	1000	21	0	72.26	0.0	0.0

Table G.54: PBT_CG_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22461.0000	138.42507	1000	1000000	0	73.27	0.0	0.0
2	226.1900	9.69410	1000	251	0	76.27	0.0	100.0
3	176.9975	2.81389	1000	10	0	33.32	0.0	0.0
4	157.4675	3.89918	1000	19	0	33.58	0.0	0.0
5	149.6825	1.89911	1000	12	0	37.33	0.0	0.0
6	143.6975	2.16196	1000	17	0	37.07	0.0	0.0
7	136.7050	1.52489	1000	12	0	40.82	0.0	0.0
8	134.4325	0.84212	1000	19	0	40.56	0.0	0.0
9	139.6625	5.95222	1000	12	0	44.32	0.0	0.0
10	141.7575	1.08419	1000	19	0	44.05	0.0	0.0
11	143.3300	1.50321	1000	11	0	47.81	0.0	0.0
12	140.5450	0.89355	1000	20	0	47.55	0.0	0.0
13	139.8550	3.49782	1000	11	0	51.30	0.0	0.0
14	150.8700	0.42638	1000	20	0	51.04	0.0	0.0
15	154.9900	1.75353	1000	10	0	54.80	0.0	0.0
16	145.2500	1.18226	1000	22	0	54.54	0.0	0.0
17	147.7850	2.08061	1000	9	0	58.29	0.0	0.0
18	144.7050	0.73398	1000	22	0	58.03	0.0	0.0
19	144.3350	1.77298	1000	9	0	61.78	0.0	0.0
20	150.6400	1.16673	1000	22	0	61.52	0.0	0.0
21	149.0075	1.48646	1000	8	0	65.28	0.0	0.0
22	151.4650	1.06615	1000	22	0	65.01	0.0	0.0
23	153.1200	0.85241	1000	9	0	68.77	0.0	0.0
24	154.1175	2.21502	1000	21	0	68.50	0.0	0.0
25	155.5600	0.64300	1000	9	0	72.26	0.0	0.0
26	166.4000	2.19186	1000	21	0	72.26	0.0	0.0

Table G.55: PBT_CG_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22321.0000	177.21174	1000	1000000	0	73.27	0.00	0.0
2	320.1450	19.52241	1000	2002	306	83.83	1.28	100.0
3	187.3200	3.81500	1000	10	0	33.32	0.00	0.0
4	173.5375	5.27952	1000	19	0	33.58	0.00	0.0
5	160.5450	4.06590	1000	12	0	37.33	0.00	0.0
6	153.7050	4.33404	1000	17	0	37.07	0.00	0.0
7	146.0325	3.01317	1000	12	0	40.82	0.00	0.0
8	140.1325	0.93577	1000	19	0	40.56	0.00	0.0
9	143.1550	1.71806	1000	12	0	44.32	0.00	0.0
10	148.2925	0.68936	1000	19	0	44.05	0.00	0.0
11	149.4925	1.95194	1000	11	0	47.81	0.00	0.0
12	145.7675	2.15159	1000	20	0	47.55	0.00	0.0
13	146.4100	1.29987	1000	11	0	51.30	0.00	0.0
14	163.0250	1.87609	1000	20	0	51.04	0.00	0.0
15	164.1175	5.65420	1000	10	0	54.80	0.00	0.0
16	150.4500	3.53045	1000	22	0	54.54	0.00	0.0
17	155.2475	2.25261	1000	9	0	58.29	0.00	0.0
18	151.4800	1.96350	1000	22	0	58.03	0.00	0.0
19	152.1600	3.32696	1000	9	0	61.78	0.00	0.0
20	156.9050	2.39210	1000	22	0	61.52	0.00	0.0
21	154.8000	3.00865	1000	8	0	65.28	0.00	0.0
22	157.5175	1.58509	1000	22	0	65.01	0.00	0.0
23	158.6325	2.62600	1000	9	0	68.77	0.00	0.0
24	158.7050	0.68398	1000	21	0	68.50	0.00	0.0
25	161.1625	1.07362	1000	9	0	72.26	0.00	0.0
26	170.9725	2.63319	1000	21	0	72.26	0.00	0.0

Table G.56: PBT_CG_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22359.2500	119.03440	1000	1000000	0	73.27	0.0	0.0
2	296.1450	5.20117	1000	1502	0	81.74	0.0	100.0
3	182.6300	1.52039	1000	10	0	33.32	0.0	0.0
4	163.0375	1.41493	1000	19	0	33.58	0.0	0.0
5	154.4600	0.81526	1000	12	0	37.33	0.0	0.0
6	147.7475	2.43233	1000	17	0	37.07	0.0	0.0
7	143.9900	1.81094	1000	12	0	40.82	0.0	0.0
8	139.2000	0.72090	1000	19	0	40.56	0.0	0.0
9	144.5100	2.08546	1000	12	0	44.32	0.0	0.0
10	147.4800	3.06308	1000	19	0	44.05	0.0	0.0
11	150.7675	4.20336	1000	11	0	47.81	0.0	0.0
12	146.9750	2.36083	1000	20	0	47.55	0.0	0.0
13	149.9400	6.27444	1000	11	0	51.30	0.0	0.0
14	163.4850	1.75944	1000	20	0	51.04	0.0	0.0
15	160.6000	2.20380	1000	10	0	54.80	0.0	0.0
16	152.4200	0.68699	1000	22	0	54.54	0.0	0.0
17	153.9250	1.89067	1000	9	0	58.29	0.0	0.0
18	151.9200	1.99447	1000	22	0	58.03	0.0	0.0
19	150.6150	0.55410	1000	9	0	61.78	0.0	0.0
20	156.3375	0.25791	1000	22	0	61.52	0.0	0.0
21	157.6875	5.74015	1000	8	0	65.28	0.0	0.0
22	157.3000	0.94781	1000	22	0	65.01	0.0	0.0
23	157.2625	0.36982	1000	9	0	68.77	0.0	0.0
24	158.2850	1.05353	1000	21	0	68.50	0.0	0.0
25	159.4875	0.25898	1000	9	0	72.26	0.0	0.0
26	170.1600	1.69344	1000	21	0	72.26	0.0	0.0

G.4 M3

Table G.57: M3 - Search strategies results.

Strategy	Min Mean Time (ms)	Std Min Mean Time (ms)	Max Mean Time (ms)	Std Max Mean Time (ms)	Min Sample Size	Max Sample Size	Min Mean Test Cases	Std Min Mean Test Cases	Max Mean Test Cases	Std Max Mean Test Cases	Min Mean Coverage (%)	Std Min Mean Coverage (%)	Max Mean Coverage (%)	Std Max Mean Coverage (%)	Bug Found
PBT_R_S4	0.25761	0.02466	0.25761	0.02466	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S1	0.41980	0.16939	0.41980	0.16939	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S8	5.20515	0.07648	5.20515	0.07648	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_CG_S4	148.50750	2.57708	148.50750	2.57708	1000	1000	1	0	1	0	73.78	0.0	73.78	0.0	True
PBT_CG_S6	148.58250	2.85185	148.58250	2.85185	1000	1000	1	0	1	0	73.78	0.0	73.78	0.0	True
PBT_CG_S3	155.04750	1.61808	155.04750	1.61808	1000	1000	1	0	1	0	73.78	0.0	73.78	0.0	True
PBT_R_S3	5.34077	0.19095	5.34077	0.19095	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S5	5.40767	0.19750	5.40767	0.19750	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S7	5.40477	0.17325	5.40477	0.17325	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_CG_S5	154.40000	1.34186	154.40000	1.34186	1000	1000	1	0	1	0	73.78	0.0	73.78	0.0	True
PBT_CG_S7	156.38500	1.56337	156.38500	1.56337	1000	1000	1	0	1	0	73.78	0.0	73.78	0.0	True
PBT_CG_S1	149.69250	1.19753	149.69250	1.19753	1000	1000	1	0	1	0	73.78	0.0	73.78	0.0	True
PBT_CG_S8	155.92500	2.28959	155.92500	2.28959	1000	1000	1	0	1	0	73.78	0.0	73.78	0.0	True
PBT_CG_S2	148.62250	1.42157	148.62250	1.42157	1000	1000	1	0	1	0	73.78	0.0	73.78	0.0	True
PBT_R_S6	0.36197	0.14379	0.36197	0.14379	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True
PBT_R_S2	0.34531	0.15236	0.34531	0.15236	1000	1000	1	0	1	0	0.00	0.0	0.00	0.0	True

Table G.58: M3 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.0257	0.9000	-4.9644	5.0158	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.0710	0.9000	-4.9187	5.0607	False
Random withConstantExtraction	Random withoutConstantExtraction	-4.9934	0.0010	-5.1355	-4.8513	True
Coverage-Guided withBias	Coverage-Guided withoutBias	0.5256	0.8582	-6.1594	7.2107	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	-0.4094	0.8873	-7.1025	6.2838	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-6.5881	0.0010	-7.8741	-5.3022	True
Coverage-Guided	Random	-149.3024	0.0010	-152.6940	-145.9109	True

G.4.1 Search Strategies

Table G.59: PBT_R_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	62.37000	1.25101	1000	31000	0	0.0	0.0	0.0
2	32.32575	1.26315	1000	12000	0	0.0	0.0	0.0
3	4.39600	0.06689	1000	10	0	0.0	0.0	0.0
4	0.24966	0.00291	1000	19	0	0.0	0.0	0.0
5	0.21094	0.00820	1000	12	0	0.0	0.0	0.0
6	0.21823	0.00495	1000	17	0	0.0	0.0	0.0
7	0.19689	0.00180	1000	12	0	0.0	0.0	0.0
8	0.21306	0.00406	1000	19	0	0.0	0.0	0.0
9	0.19607	0.00333	1000	12	0	0.0	0.0	0.0
10	0.21692	0.00587	1000	19	0	0.0	0.0	0.0
11	0.19963	0.00169	1000	11	0	0.0	0.0	0.0
12	0.22209	0.01082	1000	20	0	0.0	0.0	0.0
13	0.19197	0.00378	1000	11	0	0.0	0.0	0.0
14	0.21090	0.00476	1000	20	0	0.0	0.0	0.0
15	0.19115	0.00777	1000	10	0	0.0	0.0	0.0
16	0.21696	0.00348	1000	22	0	0.0	0.0	0.0
17	0.18225	0.00170	1000	9	0	0.0	0.0	0.0
18	0.21298	0.00449	1000	22	0	0.0	0.0	0.0
19	0.18562	0.00201	1000	9	0	0.0	0.0	0.0
20	0.21653	0.00732	1000	22	0	0.0	0.0	0.0
21	0.17578	0.00747	1000	8	0	0.0	0.0	0.0
22	0.21086	0.00157	1000	22	0	0.0	0.0	0.0
23	0.18130	0.00352	1000	9	0	0.0	0.0	0.0
24	0.21068	0.00332	1000	21	0	0.0	0.0	0.0
25	0.35472	0.17713	1000	9	0	0.0	0.0	0.0
26	0.41980	0.16939	1000	1	0	0.0	0.0	100.0

Table G.60: PBT_R_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	61.09800	0.93255	1000	31000	0	0.0	0.0	0.0
2	31.90200	0.43083	1000	12000	0	0.0	0.0	0.0
3	4.38262	0.01372	1000	10	0	0.0	0.0	0.0
4	0.33993	0.08019	1000	19	0	0.0	0.0	0.0
5	0.26262	0.07232	1000	12	0	0.0	0.0	0.0
6	0.26631	0.07615	1000	17	0	0.0	0.0	0.0
7	0.25333	0.05769	1000	12	0	0.0	0.0	0.0
8	0.26445	0.07456	1000	19	0	0.0	0.0	0.0
9	0.24334	0.05950	1000	12	0	0.0	0.0	0.0
10	0.28061	0.10014	1000	19	0	0.0	0.0	0.0
11	0.24073	0.06458	1000	11	0	0.0	0.0	0.0
12	0.25998	0.06528	1000	20	0	0.0	0.0	0.0
13	0.22954	0.03797	1000	11	0	0.0	0.0	0.0
14	0.25293	0.03920	1000	20	0	0.0	0.0	0.0
15	0.22304	0.03718	1000	10	0	0.0	0.0	0.0
16	0.24238	0.02361	1000	22	0	0.0	0.0	0.0
17	0.20626	0.01509	1000	9	0	0.0	0.0	0.0
18	0.23685	0.01521	1000	22	0	0.0	0.0	0.0
19	0.21246	0.02570	1000	9	0	0.0	0.0	0.0
20	0.22539	0.00123	1000	22	0	0.0	0.0	0.0
21	0.19074	0.00573	1000	8	0	0.0	0.0	0.0
22	0.21620	0.00547	1000	22	0	0.0	0.0	0.0
23	0.18732	0.00610	1000	9	0	0.0	0.0	0.0
24	0.21107	0.00236	1000	21	0	0.0	0.0	0.0
25	0.18636	0.00598	1000	9	0	0.0	0.0	0.0
26	0.34531	0.15236	1000	1	0	0.0	0.0	100.0

Table G.61: PBT_R_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	79.67725	1.11161	1000	31000	0	0.0	0.0	0.0
2	53.43925	0.61083	1000	12000	0	0.0	0.0	0.0
3	13.37925	0.06769	1000	10	0	0.0	0.0	0.0
4	7.07185	0.38686	1000	19	0	0.0	0.0	0.0
5	5.57295	0.04619	1000	12	0	0.0	0.0	0.0
6	5.62820	0.14465	1000	17	0	0.0	0.0	0.0
7	9.93875	0.14339	1000	12	0	0.0	0.0	0.0
8	5.38325	0.12186	1000	19	0	0.0	0.0	0.0
9	5.70245	0.11163	1000	12	0	0.0	0.0	0.0
10	5.29330	0.13344	1000	19	0	0.0	0.0	0.0
11	5.37600	0.05008	1000	11	0	0.0	0.0	0.0
12	5.40760	0.11955	1000	20	0	0.0	0.0	0.0
13	5.24442	0.11158	1000	11	0	0.0	0.0	0.0
14	5.55175	0.10529	1000	20	0	0.0	0.0	0.0
15	5.26858	0.07806	1000	10	0	0.0	0.0	0.0
16	5.29937	0.08816	1000	22	0	0.0	0.0	0.0
17	5.21223	0.09549	1000	9	0	0.0	0.0	0.0
18	5.23355	0.06200	1000	22	0	0.0	0.0	0.0
19	5.15318	0.10216	1000	9	0	0.0	0.0	0.0
20	5.24257	0.07189	1000	22	0	0.0	0.0	0.0
21	5.17500	0.14682	1000	8	0	0.0	0.0	0.0
22	5.17405	0.08559	1000	22	0	0.0	0.0	0.0
23	5.12812	0.10702	1000	9	0	0.0	0.0	0.0
24	5.22535	0.12708	1000	21	0	0.0	0.0	0.0
25	5.34905	0.06855	1000	9	0	0.0	0.0	0.0
26	5.34077	0.19095	1000	1	0	0.0	0.0	100.0

Table G.62: PBT_R_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	60.03850	1.02091	1000	31000	0	0.0	0.0	0.0
2	31.29900	0.67967	1000	12000	0	0.0	0.0	0.0
3	4.22915	0.00842	1000	10	0	0.0	0.0	0.0
4	0.26307	0.00598	1000	19	0	0.0	0.0	0.0
5	0.19807	0.00672	1000	12	0	0.0	0.0	0.0
6	0.20217	0.01183	1000	17	0	0.0	0.0	0.0
7	0.21063	0.02400	1000	12	0	0.0	0.0	0.0
8	0.20509	0.01261	1000	19	0	0.0	0.0	0.0
9	0.20015	0.02125	1000	12	0	0.0	0.0	0.0
10	0.20978	0.01461	1000	19	0	0.0	0.0	0.0
11	0.19672	0.01682	1000	11	0	0.0	0.0	0.0
12	0.21414	0.02256	1000	20	0	0.0	0.0	0.0
13	0.19624	0.02049	1000	11	0	0.0	0.0	0.0
14	0.28270	0.01970	1000	20	0	0.0	0.0	0.0
15	0.20158	0.02238	1000	10	0	0.0	0.0	0.0
16	0.25026	0.03394	1000	22	0	0.0	0.0	0.0
17	0.19580	0.01488	1000	9	0	0.0	0.0	0.0
18	0.22451	0.02109	1000	22	0	0.0	0.0	0.0
19	0.19504	0.01763	1000	9	0	0.0	0.0	0.0
20	0.23561	0.03947	1000	22	0	0.0	0.0	0.0
21	0.19064	0.02361	1000	8	0	0.0	0.0	0.0
22	0.22348	0.02785	1000	22	0	0.0	0.0	0.0
23	0.18895	0.01727	1000	9	0	0.0	0.0	0.0
24	0.29616	0.12599	1000	21	0	0.0	0.0	0.0
25	0.45317	0.16950	1000	9	0	0.0	0.0	0.0
26	0.25761	0.02466	1000	1	0	0.0	0.0	100.0

Table G.63: PBT_R_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	79.47125	1.00014	1000	31000	0	0.0	0.0	0.0
2	53.33575	0.52635	1000	12000	0	0.0	0.0	0.0
3	13.44400	0.12483	1000	10	0	0.0	0.0	0.0
4	6.80570	0.09554	1000	19	0	0.0	0.0	0.0
5	5.59370	0.11745	1000	12	0	0.0	0.0	0.0
6	5.66825	0.07784	1000	17	0	0.0	0.0	0.0
7	9.83672	0.07461	1000	12	0	0.0	0.0	0.0
8	5.45210	0.09199	1000	19	0	0.0	0.0	0.0
9	5.76288	0.07579	1000	12	0	0.0	0.0	0.0
10	5.40798	0.10304	1000	19	0	0.0	0.0	0.0
11	5.32150	0.10512	1000	11	0	0.0	0.0	0.0
12	5.37065	0.11768	1000	20	0	0.0	0.0	0.0
13	5.35377	0.12308	1000	11	0	0.0	0.0	0.0
14	5.54138	0.09069	1000	20	0	0.0	0.0	0.0
15	5.26205	0.08310	1000	10	0	0.0	0.0	0.0
16	5.37235	0.14045	1000	22	0	0.0	0.0	0.0
17	5.25218	0.10299	1000	9	0	0.0	0.0	0.0
18	5.27527	0.06716	1000	22	0	0.0	0.0	0.0
19	5.19435	0.13541	1000	9	0	0.0	0.0	0.0
20	5.27787	0.06461	1000	22	0	0.0	0.0	0.0
21	5.28305	0.06031	1000	8	0	0.0	0.0	0.0
22	5.21783	0.08902	1000	22	0	0.0	0.0	0.0
23	5.23545	0.05600	1000	9	0	0.0	0.0	0.0
24	5.21710	0.12091	1000	21	0	0.0	0.0	0.0
25	5.37157	0.05775	1000	9	0	0.0	0.0	0.0
26	5.40767	0.19750	1000	1	0	0.0	0.0	100.0

Table G.64: PBT_R_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	60.21150	0.63320	1000	31000	0	0.0	0.0	0.0
2	30.99400	0.30917	1000	12000	0	0.0	0.0	0.0
3	4.32355	0.01430	1000	10	0	0.0	0.0	0.0
4	0.37754	0.15337	1000	19	0	0.0	0.0	0.0
5	0.31347	0.15344	1000	12	0	0.0	0.0	0.0
6	0.30771	0.13871	1000	17	0	0.0	0.0	0.0
7	0.29211	0.12698	1000	12	0	0.0	0.0	0.0
8	0.31122	0.13993	1000	19	0	0.0	0.0	0.0
9	0.27002	0.08512	1000	12	0	0.0	0.0	0.0
10	0.27803	0.08094	1000	19	0	0.0	0.0	0.0
11	0.25122	0.06861	1000	11	0	0.0	0.0	0.0
12	0.26878	0.04831	1000	20	0	0.0	0.0	0.0
13	0.23574	0.04947	1000	11	0	0.0	0.0	0.0
14	0.31685	0.03992	1000	20	0	0.0	0.0	0.0
15	0.22707	0.02428	1000	10	0	0.0	0.0	0.0
16	0.25672	0.02758	1000	22	0	0.0	0.0	0.0
17	0.21660	0.01956	1000	9	0	0.0	0.0	0.0
18	0.25009	0.01452	1000	22	0	0.0	0.0	0.0
19	0.21417	0.01917	1000	9	0	0.0	0.0	0.0
20	0.25446	0.02604	1000	22	0	0.0	0.0	0.0
21	0.21018	0.03192	1000	8	0	0.0	0.0	0.0
22	0.22773	0.02043	1000	22	0	0.0	0.0	0.0
23	0.20023	0.02331	1000	9	0	0.0	0.0	0.0
24	0.32150	0.15304	1000	21	0	0.0	0.0	0.0
25	0.20014	0.01795	1000	9	0	0.0	0.0	0.0
26	0.36197	0.14379	1000	1	0	0.0	0.0	100.0

Table G.65: PBT_R_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	78.93850	0.34677	1000	31000	0	0.0	0.0	0.0
2	53.04100	0.99193	1000	12000	0	0.0	0.0	0.0
3	13.33625	0.12969	1000	10	0	0.0	0.0	0.0
4	6.87017	0.33831	1000	19	0	0.0	0.0	0.0
5	5.50307	0.11931	1000	12	0	0.0	0.0	0.0
6	5.66850	0.18238	1000	17	0	0.0	0.0	0.0
7	5.41168	0.05157	1000	12	0	0.0	0.0	0.0
8	5.30373	0.06426	1000	19	0	0.0	0.0	0.0
9	5.85155	0.30331	1000	12	0	0.0	0.0	0.0
10	5.37660	0.09026	1000	19	0	0.0	0.0	0.0
11	5.30075	0.09597	1000	11	0	0.0	0.0	0.0
12	5.35047	0.05496	1000	20	0	0.0	0.0	0.0
13	5.33177	0.10133	1000	11	0	0.0	0.0	0.0
14	9.83200	0.11669	1000	20	0	0.0	0.0	0.0
15	5.22910	0.09393	1000	10	0	0.0	0.0	0.0
16	5.26170	0.05051	1000	22	0	0.0	0.0	0.0
17	5.26485	0.07150	1000	9	0	0.0	0.0	0.0
18	5.24120	0.08310	1000	22	0	0.0	0.0	0.0
19	5.14460	0.10499	1000	9	0	0.0	0.0	0.0
20	5.29683	0.08661	1000	22	0	0.0	0.0	0.0
21	5.16055	0.10199	1000	8	0	0.0	0.0	0.0
22	5.14905	0.10563	1000	22	0	0.0	0.0	0.0
23	5.17242	0.06373	1000	9	0	0.0	0.0	0.0
24	5.26323	0.06910	1000	21	0	0.0	0.0	0.0
25	5.18670	0.05750	1000	9	0	0.0	0.0	0.0
26	5.40477	0.17325	1000	1	0	0.0	0.0	100.0

Table G.66: PBT_R_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	82.07300	4.37215	1000	31000	0	0.0	0.0	0.0
2	53.60075	0.75518	1000	12000	0	0.0	0.0	0.0
3	13.45775	0.15975	1000	10	0	0.0	0.0	0.0
4	6.84527	0.05544	1000	19	0	0.0	0.0	0.0
5	5.47695	0.13469	1000	12	0	0.0	0.0	0.0
6	5.53210	0.09828	1000	17	0	0.0	0.0	0.0
7	5.36630	0.11060	1000	12	0	0.0	0.0	0.0
8	5.39002	0.12022	1000	19	0	0.0	0.0	0.0
9	5.82730	0.13672	1000	12	0	0.0	0.0	0.0
10	5.28855	0.06258	1000	19	0	0.0	0.0	0.0
11	5.36137	0.19097	1000	11	0	0.0	0.0	0.0
12	5.32595	0.12037	1000	20	0	0.0	0.0	0.0
13	5.22995	0.09177	1000	11	0	0.0	0.0	0.0
14	9.90057	0.07010	1000	20	0	0.0	0.0	0.0
15	5.27460	0.08189	1000	10	0	0.0	0.0	0.0
16	5.44580	0.12016	1000	22	0	0.0	0.0	0.0
17	5.22467	0.02986	1000	9	0	0.0	0.0	0.0
18	5.29043	0.07061	1000	22	0	0.0	0.0	0.0
19	5.19968	0.08495	1000	9	0	0.0	0.0	0.0
20	5.25243	0.10088	1000	22	0	0.0	0.0	0.0
21	5.15212	0.11433	1000	8	0	0.0	0.0	0.0
22	5.15815	0.09394	1000	22	0	0.0	0.0	0.0
23	5.19303	0.10568	1000	9	0	0.0	0.0	0.0
24	5.21402	0.10374	1000	21	0	0.0	0.0	0.0
25	5.08995	0.05742	1000	9	0	0.0	0.0	0.0
26	5.20515	0.07648	1000	1	0	0.0	0.0	100.0

Table G.67: PBT_CG_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22165.2500	264.00509	1000	1000000	0	73.27	0.0	0.0
2	25079.2500	117.03712	1000	1000000	0	85.50	0.0	0.0
3	186.7450	4.38801	1000	10	0	33.32	0.0	0.0
4	173.7975	2.06523	1000	19	0	33.58	0.0	0.0
5	169.0900	2.82442	1000	12	0	37.33	0.0	0.0
6	165.6600	0.79350	1000	17	0	37.07	0.0	0.0
7	161.5350	0.91707	1000	12	0	40.82	0.0	0.0
8	159.0850	2.02142	1000	19	0	40.56	0.0	0.0
9	157.5025	1.29955	1000	12	0	44.32	0.0	0.0
10	162.9450	1.51912	1000	19	0	44.05	0.0	0.0
11	165.1925	1.64847	1000	11	0	47.81	0.0	0.0
12	163.8125	2.57437	1000	20	0	47.55	0.0	0.0
13	158.3475	1.61374	1000	11	0	51.30	0.0	0.0
14	163.4175	1.23453	1000	20	0	51.04	0.0	0.0
15	174.8650	0.86228	1000	10	0	54.80	0.0	0.0
16	166.6875	0.73018	1000	22	0	54.54	0.0	0.0
17	189.9300	21.32643	1000	9	0	58.29	0.0	0.0
18	177.3950	17.80116	1000	22	0	58.03	0.0	0.0
19	146.4425	6.53338	1000	9	0	61.78	0.0	0.0
20	141.3175	1.68774	1000	22	0	61.52	0.0	0.0
21	139.1125	1.72772	1000	8	0	65.28	0.0	0.0
22	141.2425	1.12190	1000	22	0	65.01	0.0	0.0
23	137.7050	0.98475	1000	9	0	68.77	0.0	0.0
24	136.4725	1.58695	1000	21	0	68.50	0.0	0.0
25	137.0000	1.16151	1000	9	0	72.26	0.0	0.0
26	149.6925	1.19753	1000	1	0	73.78	0.0	100.0

Table G.68: PBT_CG_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22306.0000	251.63764	1000	1000000	0	73.27	0.0	0.0
2	25152.7500	81.70488	1000	1000000	0	85.50	0.0	0.0
3	191.7275	6.04347	1000	10	0	33.32	0.0	0.0
4	176.1025	2.27998	1000	19	0	33.58	0.0	0.0
5	174.3100	3.98062	1000	12	0	37.33	0.0	0.0
6	167.6675	1.72478	1000	17	0	37.07	0.0	0.0
7	162.0400	1.75384	1000	12	0	40.82	0.0	0.0
8	159.2225	0.79222	1000	19	0	40.56	0.0	0.0
9	158.3675	1.78005	1000	12	0	44.32	0.0	0.0
10	162.8200	1.69016	1000	19	0	44.05	0.0	0.0
11	165.5750	1.98860	1000	11	0	47.81	0.0	0.0
12	163.2525	1.44994	1000	20	0	47.55	0.0	0.0
13	158.3175	1.01657	1000	11	0	51.30	0.0	0.0
14	168.2300	1.77440	1000	20	0	51.04	0.0	0.0
15	176.5300	1.62026	1000	10	0	54.80	0.0	0.0
16	168.2300	1.28792	1000	22	0	54.54	0.0	0.0
17	194.9075	15.16958	1000	9	0	58.29	0.0	0.0
18	165.7850	17.11249	1000	22	0	58.03	0.0	0.0
19	142.4950	4.98333	1000	9	0	61.78	0.0	0.0
20	140.0650	2.92553	1000	22	0	61.52	0.0	0.0
21	139.8175	1.63194	1000	8	0	65.28	0.0	0.0
22	138.3975	1.19751	1000	22	0	65.01	0.0	0.0
23	136.4900	0.76033	1000	9	0	68.77	0.0	0.0
24	136.0575	1.16988	1000	21	0	68.50	0.0	0.0
25	135.9000	1.72785	1000	9	0	72.26	0.0	0.0
26	148.6225	1.42157	1000	1	0	73.78	0.0	100.0

Table G.69: PBT_CG_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22467.0000	318.95376	1000	1000000	0	73.27	0.0	0.0
2	25500.7500	169.86668	1000	1000000	0	85.50	0.0	0.0
3	215.3650	24.20095	1000	10	0	33.32	0.0	0.0
4	204.8625	24.55078	1000	19	0	33.58	0.0	0.0
5	180.2875	12.95451	1000	12	0	37.33	0.0	0.0
6	168.8050	16.43698	1000	17	0	37.07	0.0	0.0
7	164.8450	14.67109	1000	12	0	40.82	0.0	0.0
8	160.8950	15.69023	1000	19	0	40.56	0.0	0.0
9	158.1125	14.51690	1000	12	0	44.32	0.0	0.0
10	180.7175	11.24769	1000	19	0	44.05	0.0	0.0
11	171.6575	5.21547	1000	11	0	47.81	0.0	0.0
12	161.8025	13.29490	1000	20	0	47.55	0.0	0.0
13	158.8375	9.99360	1000	11	0	51.30	0.0	0.0
14	171.3275	13.09291	1000	20	0	51.04	0.0	0.0
15	173.3100	15.05675	1000	10	0	54.80	0.0	0.0
16	176.1725	31.22268	1000	22	0	54.54	0.0	0.0
17	183.4075	32.68132	1000	9	0	58.29	0.0	0.0
18	150.5125	9.51133	1000	22	0	58.03	0.0	0.0
19	143.8150	4.59749	1000	9	0	61.78	0.0	0.0
20	143.2150	4.14545	1000	22	0	61.52	0.0	0.0
21	141.6875	3.36377	1000	8	0	65.28	0.0	0.0
22	142.1250	2.09046	1000	22	0	65.01	0.0	0.0
23	141.2550	1.07895	1000	9	0	68.77	0.0	0.0
24	140.9050	0.56256	1000	21	0	68.50	0.0	0.0
25	140.3425	1.36361	1000	9	0	72.26	0.0	0.0
26	155.0475	1.61808	1000	1	0	73.78	0.0	100.0

Table G.70: PBT_CG_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22136.0000	126.04166	1000	1000000	0	73.27	0.0	0.0
2	25134.0000	156.69556	1000	1000000	0	85.50	0.0	0.0
3	187.5225	3.06051	1000	10	0	33.32	0.0	0.0
4	173.6600	4.19490	1000	19	0	33.58	0.0	0.0
5	170.4900	1.09209	1000	12	0	37.33	0.0	0.0
6	167.9175	0.80763	1000	17	0	37.07	0.0	0.0
7	162.9500	2.65613	1000	12	0	40.82	0.0	0.0
8	160.5000	4.31435	1000	19	0	40.56	0.0	0.0
9	158.1600	2.42995	1000	12	0	44.32	0.0	0.0
10	163.7725	3.74832	1000	19	0	44.05	0.0	0.0
11	165.6625	1.54810	1000	11	0	47.81	0.0	0.0
12	164.1850	1.43598	1000	20	0	47.55	0.0	0.0
13	158.2025	0.64569	1000	11	0	51.30	0.0	0.0
14	165.8325	3.15655	1000	20	0	51.04	0.0	0.0
15	174.9825	1.09673	1000	10	0	54.80	0.0	0.0
16	166.9050	1.34001	1000	22	0	54.54	0.0	0.0
17	178.9825	11.44709	1000	9	0	58.29	0.0	0.0
18	176.8075	20.59974	1000	22	0	58.03	0.0	0.0
19	145.9325	5.21090	1000	9	0	61.78	0.0	0.0
20	139.7200	1.47843	1000	22	0	61.52	0.0	0.0
21	138.4700	0.72805	1000	8	0	65.28	0.0	0.0
22	136.0100	0.46395	1000	22	0	65.01	0.0	0.0
23	135.4650	1.39432	1000	9	0	68.77	0.0	0.0
24	134.4875	1.36643	1000	21	0	68.50	0.0	0.0
25	134.9075	1.82459	1000	9	0	72.26	0.0	0.0
26	148.5075	2.57708	1000	1	0	73.78	0.0	100.0

Table G.71: PBT_CG_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22400.5000	268.38824	1000	1000000	0	73.27	0.0	0.0
2	25694.7500	394.13283	1000	1000000	0	85.50	0.0	0.0
3	201.8925	8.87058	1000	10	0	33.32	0.0	0.0
4	192.7875	5.53976	1000	19	0	33.58	0.0	0.0
5	185.6800	2.66600	1000	12	0	37.33	0.0	0.0
6	174.9325	2.89176	1000	17	0	37.07	0.0	0.0
7	175.4000	2.12078	1000	12	0	40.82	0.0	0.0
8	170.3000	2.86031	1000	19	0	40.56	0.0	0.0
9	165.4100	1.42802	1000	12	0	44.32	0.0	0.0
10	174.2800	5.50850	1000	19	0	44.05	0.0	0.0
11	174.0250	1.51436	1000	11	0	47.81	0.0	0.0
12	167.2975	0.86343	1000	20	0	47.55	0.0	0.0
13	166.4350	0.96183	1000	11	0	51.30	0.0	0.0
14	178.9900	1.70949	1000	20	0	51.04	0.0	0.0
15	181.8650	1.45878	1000	10	0	54.80	0.0	0.0
16	181.9125	16.48287	1000	22	0	54.54	0.0	0.0
17	192.5825	25.15591	1000	9	0	58.29	0.0	0.0
18	170.4950	20.63424	1000	22	0	58.03	0.0	0.0
19	149.4100	6.06906	1000	9	0	61.78	0.0	0.0
20	146.7700	0.88504	1000	22	0	61.52	0.0	0.0
21	144.4875	1.49817	1000	8	0	65.28	0.0	0.0
22	143.6625	1.74101	1000	22	0	65.01	0.0	0.0
23	142.4100	1.05278	1000	9	0	68.77	0.0	0.0
24	140.9525	0.87987	1000	21	0	68.50	0.0	0.0
25	140.1475	0.82211	1000	9	0	72.26	0.0	0.0
26	154.4000	1.34186	1000	1	0	73.78	0.0	100.0

Table G.72: PBT_CG_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22172.7500	106.19881	1000	1000000	0	73.27	0.0	0.0
2	25253.0000	118.21379	1000	1000000	0	85.50	0.0	0.0
3	190.5500	1.09275	1000	10	0	33.32	0.0	0.0
4	179.2150	5.02804	1000	19	0	33.58	0.0	0.0
5	174.9150	4.00080	1000	12	0	37.33	0.0	0.0
6	171.8500	3.81960	1000	17	0	37.07	0.0	0.0
7	162.6850	1.29276	1000	12	0	40.82	0.0	0.0
8	159.3675	0.51363	1000	19	0	40.56	0.0	0.0
9	157.9225	1.44581	1000	12	0	44.32	0.0	0.0
10	165.9950	3.23956	1000	19	0	44.05	0.0	0.0
11	166.0475	1.30574	1000	11	0	47.81	0.0	0.0
12	163.0825	1.59750	1000	20	0	47.55	0.0	0.0
13	158.4875	0.80865	1000	11	0	51.30	0.0	0.0
14	168.5400	1.10469	1000	20	0	51.04	0.0	0.0
15	176.7400	1.81065	1000	10	0	54.80	0.0	0.0
16	169.4875	0.79200	1000	22	0	54.54	0.0	0.0
17	176.5175	11.41647	1000	9	0	58.29	0.0	0.0
18	191.4575	24.13487	1000	22	0	58.03	0.0	0.0
19	149.9775	6.86883	1000	9	0	61.78	0.0	0.0
20	142.1175	3.39553	1000	22	0	61.52	0.0	0.0
21	138.7225	1.40560	1000	8	0	65.28	0.0	0.0
22	140.3775	2.62439	1000	22	0	65.01	0.0	0.0
23	138.2400	1.40824	1000	9	0	68.77	0.0	0.0
24	136.1900	2.62545	1000	21	0	68.50	0.0	0.0
25	135.4250	2.87895	1000	9	0	72.26	0.0	0.0
26	148.5825	2.85185	1000	1	0	73.78	0.0	100.0

Table G.73: PBT_CG_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22277.2500	161.49981	1000	1000000	0	73.27	0.0	0.0
2	25350.7500	76.78338	1000	1000000	0	85.50	0.0	0.0
3	195.2450	3.16220	1000	10	0	33.32	0.0	0.0
4	191.8375	3.67027	1000	19	0	33.58	0.0	0.0
5	182.1300	3.10086	1000	12	0	37.33	0.0	0.0
6	174.6800	3.70726	1000	17	0	37.07	0.0	0.0
7	172.6875	2.16618	1000	12	0	40.82	0.0	0.0
8	169.0250	3.78273	1000	19	0	40.56	0.0	0.0
9	166.3000	0.89686	1000	12	0	44.32	0.0	0.0
10	169.1600	1.54169	1000	19	0	44.05	0.0	0.0
11	170.3825	2.55439	1000	11	0	47.81	0.0	0.0
12	168.7350	1.06777	1000	20	0	47.55	0.0	0.0
13	164.8925	1.16607	1000	11	0	51.30	0.0	0.0
14	182.2000	0.71456	1000	20	0	51.04	0.0	0.0
15	183.4150	1.69960	1000	10	0	54.80	0.0	0.0
16	198.6000	26.31938	1000	22	0	54.54	0.0	0.0
17	190.8600	23.73240	1000	9	0	58.29	0.0	0.0
18	156.7950	9.98906	1000	22	0	58.03	0.0	0.0
19	145.3550	4.11364	1000	9	0	61.78	0.0	0.0
20	146.4750	3.66919	1000	22	0	61.52	0.0	0.0
21	142.9300	1.73046	1000	8	0	65.28	0.0	0.0
22	143.7525	2.25050	1000	22	0	65.01	0.0	0.0
23	142.2225	1.75739	1000	9	0	68.77	0.0	0.0
24	140.4700	1.99247	1000	21	0	68.50	0.0	0.0
25	140.1650	1.20753	1000	9	0	72.26	0.0	0.0
26	156.3850	1.56337	1000	1	0	73.78	0.0	100.0

Table G.74: PBT_CG_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22447.7500	264.66240	1000	1000000	0	73.27	0.0	0.0
2	25571.2500	300.04698	1000	1000000	0	85.50	0.0	0.0
3	199.5950	1.79001	1000	10	0	33.32	0.0	0.0
4	188.6425	1.83447	1000	19	0	33.58	0.0	0.0
5	182.7125	3.63035	1000	12	0	37.33	0.0	0.0
6	174.2925	1.35280	1000	17	0	37.07	0.0	0.0
7	170.3650	2.93612	1000	12	0	40.82	0.0	0.0
8	166.5975	1.99949	1000	19	0	40.56	0.0	0.0
9	169.5600	4.39450	1000	12	0	44.32	0.0	0.0
10	173.9575	3.82820	1000	19	0	44.05	0.0	0.0
11	175.4425	5.90090	1000	11	0	47.81	0.0	0.0
12	168.4525	0.96991	1000	20	0	47.55	0.0	0.0
13	165.4625	2.42185	1000	11	0	51.30	0.0	0.0
14	184.7725	0.31140	1000	20	0	51.04	0.0	0.0
15	182.7100	0.77450	1000	10	0	54.80	0.0	0.0
16	186.9025	23.70069	1000	22	0	54.54	0.0	0.0
17	189.9250	20.95807	1000	9	0	58.29	0.0	0.0
18	167.7850	23.30521	1000	22	0	58.03	0.0	0.0
19	148.6325	5.55799	1000	9	0	61.78	0.0	0.0
20	146.8475	3.28435	1000	22	0	61.52	0.0	0.0
21	144.5175	1.82335	1000	8	0	65.28	0.0	0.0
22	144.8850	2.32033	1000	22	0	65.01	0.0	0.0
23	143.0575	3.06660	1000	9	0	68.77	0.0	0.0
24	144.6850	7.09849	1000	21	0	68.50	0.0	0.0
25	141.8425	3.13430	1000	9	0	72.26	0.0	0.0
26	155.9250	2.28959	1000	1	0	73.78	0.0	100.0

G.5 M4

Table G.75: M4 - Search strategies results.

Strategy	Min Mean Time (ms)	Std Min Mean Time (ms)	Max Mean Time (ms)	Std Max Mean Time (ms)	Min Sample Size	Max Sample Size	Min Mean Test Cases	Std Min Mean Test Cases	Max Mean Test Cases	Std Max Mean Test Cases	Min Mean Coverage (%)	Std Min Mean Coverage (%)	Max Mean Coverage (%)	Std Max Mean Coverage (%)	Bug Found
PBT_R_S4	0.23709	0.02225	0.23709	0.02225	1000	1000	3	1	3	1	0.00	0.0	0.00	0.0	True
PBT_R_S1	0.24897	0.00881	0.24897	0.00881	1000	1000	5	1	5	1	0.00	0.0	0.00	0.0	True
PBT_R_S8	5.38940	0.12391	5.38940	0.12391	1000	1000	2	0	2	0	0.00	0.0	0.00	0.0	True
PBT_CG_S4	170.43750	22.06524	170.43750	22.06524	1000	1000	5	2	5	2	75.49	0.0	75.49	0.0	True
PBT_CG_S6	182.23500	23.11880	182.23500	23.11880	1000	1000	3	0	3	0	75.49	0.0	75.49	0.0	True
PBT_CG_S3	198.48500	14.53054	198.48500	14.53054	1000	1000	2	0	2	0	75.49	0.0	75.49	0.0	True
PBT_R_S3	5.24895	0.09228	5.24895	0.09228	1000	1000	2	0	2	0	0.00	0.0	0.00	0.0	True
PBT_R_S5	5.25223	0.11788	5.25223	0.11788	1000	1000	2	0	2	0	0.00	0.0	0.00	0.0	True
PBT_R_S7	5.37355	0.10629	5.37355	0.10629	1000	1000	2	0	2	0	0.00	0.0	0.00	0.0	True
PBT_CG_S5	200.99000	20.98800	200.99000	20.98800	1000	1000	2	0	2	0	75.49	0.0	75.49	0.0	True
PBT_CG_S7	188.46750	20.33549	188.46750	20.33549	1000	1000	2	0	2	0	75.49	0.0	75.49	0.0	True
PBT_CG_S1	192.58000	16.34087	192.58000	16.34087	1000	1000	6	1	6	1	75.49	0.0	75.49	0.0	True
PBT_CG_S8	195.13750	19.72838	195.13750	19.72838	1000	1000	2	0	2	0	75.49	0.0	75.49	0.0	True
PBT_CG_S2	188.48500	20.53375	188.48500	20.53375	1000	1000	3	0	3	0	75.49	0.0	75.49	0.0	True
PBT_R_S6	0.27210	0.02742	0.27210	0.02742	1000	1000	3	0	3	0	0.00	0.0	0.00	0.0	True
PBT_R_S2	0.26121	0.00504	0.26121	0.00504	1000	1000	3	0	3	0	0.00	0.0	0.00	0.0	True

Table G.76: M4 - Tukey's test results concerning the *Time* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	-0.0166	0.9000	-5.0733	5.0401	False
Random withPairWiseTesting	Random withoutPairWiseTesting	-0.0652	0.9000	-5.1215	4.9911	False
Random withConstantExtraction	Random withoutConstantExtraction	-5.0612	0.0010	-5.1558	-4.9666	True
Coverage-Guided withBias	Coverage-Guided withoutBias	-4.2194	0.5943	-22.0413	13.6026	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	11.0656	0.1131	-3.5357	25.6669	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	-12.3356	0.0675	-25.8833	1.2121	False
Coverage-Guided	Random	-186.8168	0.0010	-194.5256	-179.1079	True

Table G.77: M4 - Tukey's test results concerning the *Number of Test Cases* evaluation metric.

Group1	Group2	Mean Diff	P-Adj	Lower	Upper	Reject
Random withBias	Random withoutBias	0.500	0.5429	-1.3689	2.3689	False
Random withPairWiseTesting	Random withoutPairWiseTesting	0.500	0.5429	-1.3689	2.3689	False
Random withConstantExtraction	Random withoutConstantExtraction	1.500	0.0240	0.2765	2.7235	True
Coverage-Guided withBias	Coverage-Guided withoutBias	1.250	0.2872	-1.3693	3.8693	False
Coverage-Guided withPairWiseTesting	Coverage-Guided withoutPairWiseTesting	0.250	0.8451	-2.6409	3.1409	False
Coverage-Guided withConstantExtraction	Coverage-Guided withoutConstantExtraction	2.250	0.0240	0.4148	4.0852	True
Coverage-Guided	Random	-0.375	0.5879	-1.7900	1.0400	False

G.5.1 Search Strategies

Table G.78: PBT_R_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	64.11200	1.36028	1000	31000	0	0.0	0.0	0.0
2	33.54900	0.60257	1000	12000	0	0.0	0.0	0.0
3	4.31785	0.04465	1000	10	0	0.0	0.0	0.0
4	0.24556	0.00631	1000	19	0	0.0	0.0	0.0
5	0.20964	0.00407	1000	12	0	0.0	0.0	0.0
6	0.22346	0.01789	1000	17	0	0.0	0.0	0.0
7	0.19823	0.00331	1000	12	0	0.0	0.0	0.0
8	0.21347	0.00508	1000	19	0	0.0	0.0	0.0
9	0.19746	0.00374	1000	12	0	0.0	0.0	0.0
10	0.21083	0.00318	1000	19	0	0.0	0.0	0.0
11	0.19292	0.00146	1000	11	0	0.0	0.0	0.0
12	0.21820	0.00776	1000	20	0	0.0	0.0	0.0
13	0.19090	0.00265	1000	11	0	0.0	0.0	0.0
14	0.21502	0.00679	1000	20	0	0.0	0.0	0.0
15	0.18836	0.00329	1000	10	0	0.0	0.0	0.0
16	0.21549	0.00503	1000	22	0	0.0	0.0	0.0
17	0.24897	0.00881	1000	5	1	0.0	0.0	100.0
18	0.23164	0.00736	1000	22	0	0.0	0.0	0.0
19	0.19206	0.00231	1000	9	0	0.0	0.0	0.0
20	0.23033	0.00209	1000	22	0	0.0	0.0	0.0
21	0.18188	0.00241	1000	8	0	0.0	0.0	0.0
22	0.22043	0.00560	1000	22	0	0.0	0.0	0.0
23	0.18417	0.00259	1000	9	0	0.0	0.0	0.0
24	0.30102	0.14713	1000	21	0	0.0	0.0	0.0
25	0.18286	0.00194	1000	9	0	0.0	0.0	0.0
26	0.32630	0.17141	1000	21	0	0.0	0.0	0.0

Table G.79: PBT_R_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	60.90625	1.67065	1000	31000	0	0.0	0.0	0.0
2	31.20175	0.42285	1000	12000	0	0.0	0.0	0.0
3	4.39320	0.07372	1000	10	0	0.0	0.0	0.0
4	0.29704	0.00948	1000	19	0	0.0	0.0	0.0
5	0.21873	0.00612	1000	12	0	0.0	0.0	0.0
6	0.22542	0.00226	1000	17	0	0.0	0.0	0.0
7	0.22997	0.01964	1000	12	0	0.0	0.0	0.0
8	0.22873	0.00932	1000	19	0	0.0	0.0	0.0
9	0.20904	0.00465	1000	12	0	0.0	0.0	0.0
10	0.22532	0.00754	1000	19	0	0.0	0.0	0.0
11	0.20395	0.00525	1000	11	0	0.0	0.0	0.0
12	0.22541	0.00676	1000	20	0	0.0	0.0	0.0
13	0.20771	0.00505	1000	11	0	0.0	0.0	0.0
14	0.24035	0.01218	1000	20	0	0.0	0.0	0.0
15	0.20228	0.00487	1000	10	0	0.0	0.0	0.0
16	0.23167	0.00137	1000	22	0	0.0	0.0	0.0
17	0.26121	0.00504	1000	3	0	0.0	0.0	100.0
18	0.24424	0.00970	1000	22	0	0.0	0.0	0.0
19	0.19607	0.00311	1000	9	0	0.0	0.0	0.0
20	0.23200	0.00561	1000	22	0	0.0	0.0	0.0
21	0.19569	0.00243	1000	8	0	0.0	0.0	0.0
22	0.22654	0.00342	1000	22	0	0.0	0.0	0.0
23	0.19219	0.00196	1000	9	0	0.0	0.0	0.0
24	0.22820	0.00809	1000	21	0	0.0	0.0	0.0
25	0.27869	0.14974	1000	9	0	0.0	0.0	0.0
26	0.22335	0.00228	1000	21	0	0.0	0.0	0.0

Table G.80: PBT_R_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	79.45850	0.68545	1000	31000	0	0.0	0.0	0.0
2	53.03350	0.50283	1000	12000	0	0.0	0.0	0.0
3	13.39825	0.04693	1000	10	0	0.0	0.0	0.0
4	6.97677	0.07410	1000	19	0	0.0	0.0	0.0
5	5.58253	0.06675	1000	12	0	0.0	0.0	0.0
6	5.74052	0.09578	1000	17	0	0.0	0.0	0.0
7	9.94165	0.14357	1000	12	0	0.0	0.0	0.0
8	5.45510	0.10164	1000	19	0	0.0	0.0	0.0
9	5.80178	0.08696	1000	12	0	0.0	0.0	0.0
10	5.36078	0.10770	1000	19	0	0.0	0.0	0.0
11	5.36477	0.10325	1000	11	0	0.0	0.0	0.0
12	5.37078	0.03744	1000	20	0	0.0	0.0	0.0
13	5.30805	0.05022	1000	11	0	0.0	0.0	0.0
14	5.68030	0.07762	1000	20	0	0.0	0.0	0.0
15	5.35503	0.08622	1000	10	0	0.0	0.0	0.0
16	5.27572	0.10502	1000	22	0	0.0	0.0	0.0
17	5.24895	0.09228	1000	2	0	0.0	0.0	100.0
18	5.34192	0.12613	1000	22	0	0.0	0.0	0.0
19	5.27053	0.14378	1000	9	0	0.0	0.0	0.0
20	5.22487	0.04939	1000	22	0	0.0	0.0	0.0
21	5.17165	0.09286	1000	8	0	0.0	0.0	0.0
22	5.25685	0.13559	1000	22	0	0.0	0.0	0.0
23	5.23198	0.11342	1000	9	0	0.0	0.0	0.0
24	5.15990	0.10715	1000	21	0	0.0	0.0	0.0
25	5.38632	0.10312	1000	9	0	0.0	0.0	0.0
26	5.24500	0.06478	1000	21	0	0.0	0.0	0.0

Table G.81: PBT_R_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	59.71750	0.87524	1000	31000	0	0.0	0.0	0.0
2	31.84450	0.68360	1000	12000	0	0.0	0.0	0.0
3	4.21312	0.05256	1000	10	0	0.0	0.0	0.0
4	0.33283	0.12192	1000	19	0	0.0	0.0	0.0
5	0.25237	0.10093	1000	12	0	0.0	0.0	0.0
6	0.26098	0.11779	1000	17	0	0.0	0.0	0.0
7	0.25045	0.08755	1000	12	0	0.0	0.0	0.0
8	0.25616	0.10341	1000	19	0	0.0	0.0	0.0
9	0.23542	0.09713	1000	12	0	0.0	0.0	0.0
10	0.24864	0.09822	1000	19	0	0.0	0.0	0.0
11	0.22102	0.06020	1000	11	0	0.0	0.0	0.0
12	0.23652	0.05693	1000	20	0	0.0	0.0	0.0
13	0.19898	0.03829	1000	11	0	0.0	0.0	0.0
14	0.29945	0.04743	1000	20	0	0.0	0.0	0.0
15	0.20132	0.03141	1000	10	0	0.0	0.0	0.0
16	0.22221	0.02392	1000	22	0	0.0	0.0	0.0
17	0.23709	0.02225	1000	3	1	0.0	0.0	100.0
18	0.22755	0.00663	1000	22	0	0.0	0.0	0.0
19	0.18739	0.00483	1000	9	0	0.0	0.0	0.0
20	0.21859	0.00184	1000	22	0	0.0	0.0	0.0
21	0.17571	0.00942	1000	8	0	0.0	0.0	0.0
22	0.21085	0.00509	1000	22	0	0.0	0.0	0.0
23	0.17490	0.00435	1000	9	0	0.0	0.0	0.0
24	0.20593	0.00511	1000	21	0	0.0	0.0	0.0
25	0.17375	0.00506	1000	9	0	0.0	0.0	0.0
26	0.36506	0.15427	1000	21	0	0.0	0.0	0.0

Table G.82: PBT_R_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	79.74625	1.00523	1000	31000	0	0.0	0.0	0.0
2	53.56425	1.51619	1000	12000	0	0.0	0.0	0.0
3	13.40025	0.13297	1000	10	0	0.0	0.0	0.0
4	7.03620	0.40014	1000	19	0	0.0	0.0	0.0
5	5.60825	0.11644	1000	12	0	0.0	0.0	0.0
6	5.80297	0.15746	1000	17	0	0.0	0.0	0.0
7	9.89930	0.22537	1000	12	0	0.0	0.0	0.0
8	5.52700	0.19151	1000	19	0	0.0	0.0	0.0
9	5.86220	0.14187	1000	12	0	0.0	0.0	0.0
10	5.54220	0.07055	1000	19	0	0.0	0.0	0.0
11	5.38220	0.12849	1000	11	0	0.0	0.0	0.0
12	5.51143	0.12107	1000	20	0	0.0	0.0	0.0
13	5.45012	0.14575	1000	11	0	0.0	0.0	0.0
14	5.78450	0.09247	1000	20	0	0.0	0.0	0.0
15	5.30210	0.12572	1000	10	0	0.0	0.0	0.0
16	5.35290	0.18085	1000	22	0	0.0	0.0	0.0
17	5.25223	0.11788	1000	2	0	0.0	0.0	100.0
18	5.29480	0.08394	1000	22	0	0.0	0.0	0.0
19	5.28472	0.13025	1000	9	0	0.0	0.0	0.0
20	5.28290	0.13300	1000	22	0	0.0	0.0	0.0
21	5.25375	0.18254	1000	8	0	0.0	0.0	0.0
22	5.23500	0.19724	1000	22	0	0.0	0.0	0.0
23	5.36610	0.12196	1000	9	0	0.0	0.0	0.0
24	5.29345	0.13351	1000	21	0	0.0	0.0	0.0
25	5.50310	0.07188	1000	9	0	0.0	0.0	0.0
26	5.38292	0.11471	1000	21	0	0.0	0.0	0.0

Table G.83: PBT_R_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	60.94375	0.95491	1000	31000	0	0.0	0.0	0.0
2	31.38450	0.64670	1000	12000	0	0.0	0.0	0.0
3	4.29823	0.02456	1000	10	0	0.0	0.0	0.0
4	0.37619	0.14029	1000	19	0	0.0	0.0	0.0
5	0.32131	0.16934	1000	12	0	0.0	0.0	0.0
6	0.29952	0.12660	1000	17	0	0.0	0.0	0.0
7	0.29325	0.10702	1000	12	0	0.0	0.0	0.0
8	0.29811	0.11080	1000	19	0	0.0	0.0	0.0
9	0.28173	0.10769	1000	12	0	0.0	0.0	0.0
10	0.28670	0.09999	1000	19	0	0.0	0.0	0.0
11	0.25317	0.06743	1000	11	0	0.0	0.0	0.0
12	0.26521	0.06661	1000	20	0	0.0	0.0	0.0
13	0.23674	0.04157	1000	11	0	0.0	0.0	0.0
14	0.33350	0.05094	1000	20	0	0.0	0.0	0.0
15	0.24914	0.04563	1000	10	0	0.0	0.0	0.0
16	0.26606	0.03518	1000	22	0	0.0	0.0	0.0
17	0.27210	0.02742	1000	3	0	0.0	0.0	100.0
18	0.24280	0.01222	1000	22	0	0.0	0.0	0.0
19	0.21792	0.01719	1000	9	0	0.0	0.0	0.0
20	0.24916	0.02754	1000	22	0	0.0	0.0	0.0
21	0.20084	0.01829	1000	8	0	0.0	0.0	0.0
22	0.23776	0.02333	1000	22	0	0.0	0.0	0.0
23	0.20930	0.03353	1000	9	0	0.0	0.0	0.0
24	0.23572	0.02024	1000	21	0	0.0	0.0	0.0
25	0.28744	0.13409	1000	9	0	0.0	0.0	0.0
26	0.40906	0.16581	1000	21	0	0.0	0.0	0.0

Table G.84: PBT_R_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	79.86475	0.52695	1000	31000	0	0.0	0.0	0.0
2	54.16275	0.99037	1000	12000	0	0.0	0.0	0.0
3	13.34875	0.06681	1000	10	0	0.0	0.0	0.0
4	7.09320	0.31524	1000	19	0	0.0	0.0	0.0
5	6.13030	0.96489	1000	12	0	0.0	0.0	0.0
6	6.36193	1.19235	1000	17	0	0.0	0.0	0.0
7	5.82595	0.50334	1000	12	0	0.0	0.0	0.0
8	5.57203	0.14319	1000	19	0	0.0	0.0	0.0
9	5.89170	0.09726	1000	12	0	0.0	0.0	0.0
10	5.55120	0.14461	1000	19	0	0.0	0.0	0.0
11	5.51202	0.16540	1000	11	0	0.0	0.0	0.0
12	5.49750	0.12281	1000	20	0	0.0	0.0	0.0
13	5.41242	0.14069	1000	11	0	0.0	0.0	0.0
14	10.10865	0.18232	1000	20	0	0.0	0.0	0.0
15	5.34265	0.03817	1000	10	0	0.0	0.0	0.0
16	5.37437	0.11606	1000	22	0	0.0	0.0	0.0
17	5.37355	0.10629	1000	2	0	0.0	0.0	100.0
18	5.41648	0.14396	1000	22	0	0.0	0.0	0.0
19	5.31572	0.02521	1000	9	0	0.0	0.0	0.0
20	5.33228	0.16377	1000	22	0	0.0	0.0	0.0
21	5.21910	0.08223	1000	8	0	0.0	0.0	0.0
22	5.33557	0.11080	1000	22	0	0.0	0.0	0.0
23	5.34982	0.09532	1000	9	0	0.0	0.0	0.0
24	5.29295	0.09194	1000	21	0	0.0	0.0	0.0
25	5.37657	0.15068	1000	9	0	0.0	0.0	0.0
26	5.24657	0.09581	1000	21	0	0.0	0.0	0.0

Table G.85: PBT_R_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	79.67400	0.91034	1000	31000	0	0.0	0.0	0.0
2	53.88925	1.33582	1000	12000	0	0.0	0.0	0.0
3	13.44875	0.18168	1000	10	0	0.0	0.0	0.0
4	7.08427	0.22454	1000	19	0	0.0	0.0	0.0
5	5.57705	0.09709	1000	12	0	0.0	0.0	0.0
6	5.85190	0.18692	1000	17	0	0.0	0.0	0.0
7	5.65445	0.07520	1000	12	0	0.0	0.0	0.0
8	5.62473	0.11852	1000	19	0	0.0	0.0	0.0
9	5.89070	0.09709	1000	12	0	0.0	0.0	0.0
10	5.40608	0.08229	1000	19	0	0.0	0.0	0.0
11	5.42895	0.14174	1000	11	0	0.0	0.0	0.0
12	5.48763	0.08261	1000	20	0	0.0	0.0	0.0
13	5.45920	0.11238	1000	11	0	0.0	0.0	0.0
14	10.09450	0.01718	1000	20	0	0.0	0.0	0.0
15	5.39363	0.06650	1000	10	0	0.0	0.0	0.0
16	5.44907	0.10102	1000	22	0	0.0	0.0	0.0
17	5.38940	0.12391	1000	2	0	0.0	0.0	100.0
18	5.35275	0.04803	1000	22	0	0.0	0.0	0.0
19	5.35942	0.08378	1000	9	0	0.0	0.0	0.0
20	5.44368	0.16374	1000	22	0	0.0	0.0	0.0
21	5.35153	0.05819	1000	8	0	0.0	0.0	0.0
22	5.33880	0.07731	1000	22	0	0.0	0.0	0.0
23	5.36065	0.10181	1000	9	0	0.0	0.0	0.0
24	5.32510	0.08564	1000	21	0	0.0	0.0	0.0
25	5.32592	0.07448	1000	9	0	0.0	0.0	0.0
26	5.36110	0.06203	1000	21	0	0.0	0.0	0.0

Table G.86: PBT_CG_S1 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22169.7500	96.45303	1000	1000000	0	73.27	0.0	0.0
2	25115.7500	274.33043	1000	1000000	0	85.77	0.0	0.0
3	185.8350	6.77030	1000	10	0	33.32	0.0	0.0
4	175.2175	5.37626	1000	19	0	33.58	0.0	0.0
5	173.7475	4.52306	1000	12	0	37.33	0.0	0.0
6	164.5525	1.50919	1000	17	0	37.07	0.0	0.0
7	158.7475	1.46653	1000	12	0	40.82	0.0	0.0
8	158.7575	0.63503	1000	19	0	40.56	0.0	0.0
9	158.0050	2.10775	1000	12	0	44.32	0.0	0.0
10	162.6100	1.78086	1000	19	0	44.05	0.0	0.0
11	161.7650	1.23706	1000	11	0	47.81	0.0	0.0
12	161.3825	2.29271	1000	20	0	47.55	0.0	0.0
13	156.0675	0.72171	1000	11	0	51.30	0.0	0.0
14	164.0950	1.06425	1000	20	0	51.04	0.0	0.0
15	172.8575	1.24843	1000	10	0	54.80	0.0	0.0
16	164.5950	1.08928	1000	22	0	54.54	0.0	0.0
17	192.5800	16.34087	1000	6	1	75.49	0.0	100.0
18	164.9725	21.93638	1000	22	0	58.03	0.0	0.0
19	141.7700	4.09908	1000	9	0	61.78	0.0	0.0
20	138.3900	2.22062	1000	22	0	61.52	0.0	0.0
21	136.1800	2.32930	1000	8	0	65.28	0.0	0.0
22	137.9700	2.03224	1000	22	0	65.01	0.0	0.0
23	136.4025	1.14058	1000	9	0	68.77	0.0	0.0
24	137.8750	7.23001	1000	21	0	68.50	0.0	0.0
25	136.7700	5.36878	1000	9	0	72.26	0.0	0.0
26	150.2425	4.34993	1000	21	0	72.26	0.0	0.0

Table G.87: PBT_CG_S2 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22256.2500	134.56295	1000	1000000	0	73.27	0.0	0.0
2	25202.7500	86.46206	1000	1000000	0	85.77	0.0	0.0
3	186.1075	3.46786	1000	10	0	33.32	0.0	0.0
4	175.5300	4.44533	1000	19	0	33.58	0.0	0.0
5	170.0950	2.15886	1000	12	0	37.33	0.0	0.0
6	164.8350	1.04251	1000	17	0	37.07	0.0	0.0
7	159.6725	1.31921	1000	12	0	40.82	0.0	0.0
8	159.6675	2.13920	1000	19	0	40.56	0.0	0.0
9	158.4750	1.94400	1000	12	0	44.32	0.0	0.0
10	161.7400	1.81981	1000	19	0	44.05	0.0	0.0
11	163.4875	0.86332	1000	11	0	47.81	0.0	0.0
12	162.4450	0.83470	1000	20	0	47.55	0.0	0.0
13	156.8725	1.29287	1000	11	0	51.30	0.0	0.0
14	165.9925	2.52398	1000	20	0	51.04	0.0	0.0
15	172.8075	1.17508	1000	10	0	54.80	0.0	0.0
16	167.5925	0.53835	1000	22	0	54.54	0.0	0.0
17	188.4850	20.53375	1000	3	0	75.49	0.0	100.0
18	178.8775	25.79689	1000	22	0	58.03	0.0	0.0
19	149.2400	10.54463	1000	9	0	61.78	0.0	0.0
20	141.2700	4.05523	1000	22	0	61.52	0.0	0.0
21	139.3250	2.34091	1000	8	0	65.28	0.0	0.0
22	141.2075	3.90739	1000	22	0	65.01	0.0	0.0
23	136.8775	2.65627	1000	9	0	68.77	0.0	0.0
24	135.1425	1.53789	1000	21	0	68.50	0.0	0.0
25	135.9950	1.36120	1000	9	0	72.26	0.0	0.0
26	152.7475	4.78551	1000	21	0	72.26	0.0	0.0

Table G.88: PBT_CG_S3 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22186.5000	67.83251	1000	1000000	0	73.27	0.0	0.0
2	25387.2500	92.98488	1000	1000000	0	85.77	0.0	0.0
3	199.7575	4.26184	1000	10	0	33.32	0.0	0.0
4	185.3425	4.66788	1000	19	0	33.58	0.0	0.0
5	181.2800	3.83125	1000	12	0	37.33	0.0	0.0
6	170.7200	0.50105	1000	17	0	37.07	0.0	0.0
7	170.5050	0.78085	1000	12	0	40.82	0.0	0.0
8	166.5700	3.02409	1000	19	0	40.56	0.0	0.0
9	164.6875	1.36564	1000	12	0	44.32	0.0	0.0
10	169.4875	1.42205	1000	19	0	44.05	0.0	0.0
11	169.1875	1.21215	1000	11	0	47.81	0.0	0.0
12	167.7750	1.86800	1000	20	0	47.55	0.0	0.0
13	166.1175	1.67418	1000	11	0	51.30	0.0	0.0
14	178.6250	3.32351	1000	20	0	51.04	0.0	0.0
15	180.0550	3.24857	1000	10	0	54.80	0.0	0.0
16	171.2825	1.69088	1000	22	0	54.54	0.0	0.0
17	198.4850	14.53054	1000	2	0	75.49	0.0	100.0
18	169.0775	20.78263	1000	22	0	58.03	0.0	0.0
19	146.2350	3.99116	1000	9	0	61.78	0.0	0.0
20	148.9100	4.12365	1000	22	0	61.52	0.0	0.0
21	150.7925	5.61358	1000	8	0	65.28	0.0	0.0
22	148.0250	2.80155	1000	22	0	65.01	0.0	0.0
23	144.7275	2.54093	1000	9	0	68.77	0.0	0.0
24	142.9550	2.28628	1000	21	0	68.50	0.0	0.0
25	141.0400	0.82101	1000	9	0	72.26	0.0	0.0
26	156.1850	2.48864	1000	21	0	72.26	0.0	0.0

Table G.89: PBT_CG_S4 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22193.5000	58.77287	1000	1000000	0	73.27	0.0	0.0
2	25173.7500	193.34474	1000	1000000	0	85.77	0.0	0.0
3	182.3325	3.90683	1000	10	0	33.32	0.0	0.0
4	170.7875	3.53008	1000	19	0	33.58	0.0	0.0
5	172.4125	2.66397	1000	12	0	37.33	0.0	0.0
6	162.2300	3.74819	1000	17	0	37.07	0.0	0.0
7	157.4300	2.95649	1000	12	0	40.82	0.0	0.0
8	156.1725	3.50734	1000	19	0	40.56	0.0	0.0
9	156.0875	2.96262	1000	12	0	44.32	0.0	0.0
10	159.4325	2.97140	1000	19	0	44.05	0.0	0.0
11	161.3800	2.75664	1000	11	0	47.81	0.0	0.0
12	159.3800	3.77631	1000	20	0	47.55	0.0	0.0
13	156.0825	2.80868	1000	11	0	51.30	0.0	0.0
14	165.0275	2.63722	1000	20	0	51.04	0.0	0.0
15	187.2550	23.51432	1000	10	0	54.80	0.0	0.0
16	161.6925	7.24769	1000	22	0	54.54	0.0	0.0
17	170.4375	22.06524	1000	5	2	75.49	0.0	100.0
18	178.4725	33.09210	1000	22	0	58.03	0.0	0.0
19	146.4525	11.22435	1000	9	0	61.78	0.0	0.0
20	139.4975	3.37590	1000	22	0	61.52	0.0	0.0
21	138.7700	3.65188	1000	8	0	65.28	0.0	0.0
22	139.3275	4.51488	1000	22	0	65.01	0.0	0.0
23	135.1775	3.54390	1000	9	0	68.77	0.0	0.0
24	134.3625	2.52839	1000	21	0	68.50	0.0	0.0
25	135.4500	2.16132	1000	9	0	72.26	0.0	0.0
26	149.2025	2.99455	1000	21	0	72.26	0.0	0.0

Table G.90: PBT_CG_S5 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22097.5000	17.16828	1000	1000000	0	73.27	0.0	0.0
2	25182.2500	127.11093	1000	1000000	0	85.77	0.0	0.0
3	194.7425	4.07197	1000	10	0	33.32	0.0	0.0
4	185.5475	2.10672	1000	19	0	33.58	0.0	0.0
5	182.7175	2.94024	1000	12	0	37.33	0.0	0.0
6	171.5600	1.21114	1000	17	0	37.07	0.0	0.0
7	171.2950	1.01177	1000	12	0	40.82	0.0	0.0
8	164.9600	1.14029	1000	19	0	40.56	0.0	0.0
9	162.0825	0.83125	1000	12	0	44.32	0.0	0.0
10	168.2175	0.77918	1000	19	0	44.05	0.0	0.0
11	168.5925	1.73524	1000	11	0	47.81	0.0	0.0
12	164.7150	2.06551	1000	20	0	47.55	0.0	0.0
13	163.5050	0.87888	1000	11	0	51.30	0.0	0.0
14	176.2425	0.96256	1000	20	0	51.04	0.0	0.0
15	179.6125	3.53331	1000	10	0	54.80	0.0	0.0
16	183.0600	24.25260	1000	22	0	54.54	0.0	0.0
17	200.9900	20.98800	1000	2	0	75.49	0.0	100.0
18	159.3750	6.60324	1000	22	0	58.03	0.0	0.0
19	147.6150	3.01099	1000	9	0	61.78	0.0	0.0
20	146.3500	1.54543	1000	22	0	61.52	0.0	0.0
21	143.5025	1.47613	1000	8	0	65.28	0.0	0.0
22	143.3825	0.92689	1000	22	0	65.01	0.0	0.0
23	140.4650	1.33586	1000	9	0	68.77	0.0	0.0
24	139.4150	1.12442	1000	21	0	68.50	0.0	0.0
25	138.5250	1.22796	1000	9	0	72.26	0.0	0.0
26	152.2450	0.61260	1000	21	0	72.26	0.0	0.0

Table G.91: PBT_CG_S6 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22265.0000	113.07077	1000	1000000	0	73.27	0.0	0.0
2	25048.2500	125.10471	1000	1000000	0	85.77	0.0	0.0
3	186.7025	1.56768	1000	10	0	33.32	0.0	0.0
4	172.7475	1.78584	1000	19	0	33.58	0.0	0.0
5	172.1300	1.49900	1000	12	0	37.33	0.0	0.0
6	164.0475	1.13164	1000	17	0	37.07	0.0	0.0
7	159.3975	2.55112	1000	12	0	40.82	0.0	0.0
8	157.6900	1.83476	1000	19	0	40.56	0.0	0.0
9	158.5875	4.50259	1000	12	0	44.32	0.0	0.0
10	159.9825	1.32517	1000	19	0	44.05	0.0	0.0
11	165.0075	2.57405	1000	11	0	47.81	0.0	0.0
12	162.5275	2.31492	1000	20	0	47.55	0.0	0.0
13	159.6675	2.48917	1000	11	0	51.30	0.0	0.0
14	167.3025	0.63775	1000	20	0	51.04	0.0	0.0
15	172.5900	1.88940	1000	10	0	54.80	0.0	0.0
16	182.2550	15.13007	1000	22	0	54.54	0.0	0.0
17	182.2350	23.11880	1000	3	0	75.49	0.0	100.0
18	146.3125	9.01897	1000	22	0	58.03	0.0	0.0
19	139.2700	4.09205	1000	9	0	61.78	0.0	0.0
20	139.2775	4.46023	1000	22	0	61.52	0.0	0.0
21	135.8525	3.36570	1000	8	0	65.28	0.0	0.0
22	137.8325	1.97486	1000	22	0	65.01	0.0	0.0
23	136.1375	0.78052	1000	9	0	68.77	0.0	0.0
24	133.9775	1.94505	1000	21	0	68.50	0.0	0.0
25	135.9500	1.43407	1000	9	0	72.26	0.0	0.0
26	148.9000	2.77200	1000	21	0	72.26	0.0	0.0

Table G.92: PBT_CG_S7 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22312.5000	38.24591	1000	1000000	0	73.27	0.0	0.0
2	25395.5000	89.72876	1000	1000000	0	85.77	0.0	0.0
3	198.3750	2.15124	1000	10	0	33.32	0.0	0.0
4	188.5925	1.65480	1000	19	0	33.58	0.0	0.0
5	185.7900	2.36329	1000	12	0	37.33	0.0	0.0
6	177.9825	6.57217	1000	17	0	37.07	0.0	0.0
7	168.8400	3.12774	1000	12	0	40.82	0.0	0.0
8	165.2475	1.65486	1000	19	0	40.56	0.0	0.0
9	165.0975	0.62335	1000	12	0	44.32	0.0	0.0
10	169.6575	1.09545	1000	19	0	44.05	0.0	0.0
11	168.5650	1.08029	1000	11	0	47.81	0.0	0.0
12	166.3575	2.30366	1000	20	0	47.55	0.0	0.0
13	165.3075	2.79121	1000	11	0	51.30	0.0	0.0
14	181.6000	1.59341	1000	20	0	51.04	0.0	0.0
15	180.9125	1.46636	1000	10	0	54.80	0.0	0.0
16	183.0850	17.75688	1000	22	0	54.54	0.0	0.0
17	188.4675	20.33549	1000	2	0	75.49	0.0	100.0
18	164.7375	18.92686	1000	22	0	58.03	0.0	0.0
19	148.3150	8.43934	1000	9	0	61.78	0.0	0.0
20	145.2225	2.38922	1000	22	0	61.52	0.0	0.0
21	143.8375	2.49659	1000	8	0	65.28	0.0	0.0
22	144.0950	0.68009	1000	22	0	65.01	0.0	0.0
23	141.5000	0.95297	1000	9	0	68.77	0.0	0.0
24	139.7300	1.54070	1000	21	0	68.50	0.0	0.0
25	139.4725	0.97361	1000	9	0	72.26	0.0	0.0
26	154.4075	1.50910	1000	21	0	72.26	0.0	0.0

Table G.93: PBT_CG_S8 - Properties' results.

	Mean Time (ms)	Std Time (ms)	Sample Size	Mean Test Cases	Std Test Cases	Mean Coverage (%)	Std Coverage (%)	Bug Found (%)
1	22363.5000	50.35127	1000	1000000	0	73.27	0.0	0.0
2	25341.0000	82.66499	1000	1000000	0	85.77	0.0	0.0
3	194.3775	1.99526	1000	10	0	33.32	0.0	0.0
4	188.1775	6.64192	1000	19	0	33.58	0.0	0.0
5	183.5450	2.67842	1000	12	0	37.33	0.0	0.0
6	172.8400	1.80251	1000	17	0	37.07	0.0	0.0
7	167.9300	2.30610	1000	12	0	40.82	0.0	0.0
8	164.4200	1.29761	1000	19	0	40.56	0.0	0.0
9	163.1525	1.24283	1000	12	0	44.32	0.0	0.0
10	168.2325	1.38189	1000	19	0	44.05	0.0	0.0
11	169.4300	1.60281	1000	11	0	47.81	0.0	0.0
12	168.2500	1.63380	1000	20	0	47.55	0.0	0.0
13	167.5800	1.43344	1000	11	0	51.30	0.0	0.0
14	185.6450	1.47081	1000	20	0	51.04	0.0	0.0
15	183.5375	4.39894	1000	10	0	54.80	0.0	0.0
16	180.9100	14.64157	1000	22	0	54.54	0.0	0.0
17	195.1375	19.72838	1000	2	0	75.49	0.0	100.0
18	156.1400	5.11179	1000	22	0	58.03	0.0	0.0
19	144.6825	2.06708	1000	9	0	61.78	0.0	0.0
20	145.8000	3.15652	1000	22	0	61.52	0.0	0.0
21	143.4475	1.78851	1000	8	0	65.28	0.0	0.0
22	144.5150	1.20836	1000	22	0	65.01	0.0	0.0
23	141.7225	1.20914	1000	9	0	68.77	0.0	0.0
24	140.9025	1.84068	1000	21	0	68.50	0.0	0.0
25	140.6050	1.75587	1000	9	0	72.26	0.0	0.0
26	157.8500	4.58390	1000	21	0	72.26	0.0	0.0

References

[1] Roberto Blanco, Dale Miller, and Alberto Momigliano. Property-based testing via proof reconstruction. In *Proceedings of the 21st International Symposium on Principles and Practice of Declarative Programming*, PPDP '19, New York, NY, USA, 2019. Association for Computing Machinery.

[2] Niels Boyen, Carine Lucas, and Patrick Steyaert. Generalized mixin-based inheritance to support multiple inheritance. Technical report, Citeseer, 1994.

[3] M. Böhme, V. Pham, and A. Roychoudhury. Coverage-based greybox fuzzing as markov chain. *IEEE Transactions on Software Engineering*, 45(5):489–506, 2019.

[4] Cristian Cadar, Daniel Dunbar, Dawson R Engler, et al. Klee: unassisted and automatic generation of high-coverage tests for complex systems programs. In *OSDI*, volume 8, pages 209–224, 2008.

[5] Sooyoung Cha and Hakjoo Oh. Concolic testing with adaptively changing search heuristics. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2019, page 235–245, New York, NY, USA, 2019. Association for Computing Machinery.

[6] P. Chen and H. Chen. Angora: Efficient fuzzing by principled search. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 711–725, 2018.

[7] Tsong Yueh Chen, Hing Leung, and IK Mak. Adaptive random testing. In *Annual Asian Computing Science Conference*, pages 320–329. Springer, 2004.

[8] Mingi Cho, Seoyoung Kim, and Taekyoung Kwon. Intriguer: Field-level constraint solving for hybrid fuzzing. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, page 515–530, New York, NY, USA, 2019. Association for Computing Machinery.

[9] Koen Claessen and John Hughes. Quickcheck: A lightweight tool for random testing of haskell programs. *SIGPLAN Not.*, 35(9):268–279, September 2000.

[10] Richard A. DeMillo. *Software Testing*, page 1645–1649. John Wiley and Sons Ltd., GBR, 2003.

[11] B. Dolan-Gavitt, P. Hulin, E. Kirda, T. Leek, A. Mambretti, W. Robertson, F. Ulrich, and R. Whelan. Lava: Large-scale automated vulnerability addition. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 110–121, 2016.

[12] Arpita Dutta, Shubham Kumar, and Sangharatna Godboley. Enhancing test cases generated by concolic testing. In *Proceedings of the 12th Innovations on Software Engineering Conference (Formerly Known as India Software Engineering Conference)*, ISEC'19, New York, NY, USA, 2019. Association for Computing Machinery.

[13] G. Fink and K. Levitt. Property-based testing of privileged programs. In *Tenth Annual Computer Security Applications Conference*, pages 154–163, 1994.

[14] George Fink and Matt Bishop. Property-based testing: A new approach to testing for assurance. *SIGSOFT Softw. Eng. Notes*, 22(4):74–80, July 1997.

[15] Gordon Fraser and Andrea Arcuri. Evosuite: Automatic test suite generation for object-oriented software. In *Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering*, ESEC/FSE '11, page 416–419, New York, NY, USA, 2011. Association for Computing Machinery.

[16] S. Gan, C. Zhang, X. Qin, X. Tu, K. Li, Z. Pei, and Z. Chen. Collafl: Path sensitive fuzzing. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 679–696, 2018.

[17] M. Harman, Y. Jia, and Y. Zhang. Achievements, open problems and challenges for search based software testing. In *2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST)*, pages 1–12, 2015.

[18] Winston Haynes. *Tukey's Test*, pages 2303–2304. Springer New York, New York, NY, 2013.

[19] Fred Hebert. Proper testing. https://propertesting.com/book_shrinking.html, 2018. Accessed: 2021-01-26.

[20] Roger Higdon. *Hypothesis Testing*, pages 931–933. Springer New York, New York, NY, 2013.

[21] Tahbildar Hitesh and Kalita Bichitra. Automated software test data generation: Direction of research. *International Journal of Computer Science and Engineering Survey*, 2, 02 2011.

[22] ISTQB. Foundation level syllabus. https://www.istqb.org/downloads/send/2-foundation-level-documents/281-istqb-ctfl-syllabus-2018-v3-1.html. Accessed: 2021-02-09.

[23] ISTQB. Istqb glossary. https://glossary.istqb.org. Accessed: 2021-01-27.

[24] Staffs Keele et al. Guidelines for performing systematic literature reviews in software engineering. Technical report, Technical report, Ver. 2.3 EBSE Technical Report. EBSE, 2007.

[25] Leonidas Lampropoulos, Michael Hicks, and Benjamin C. Pierce. Coverage guided, property based testing. *Proc. ACM Program. Lang.*, 3(OOPSLA), October 2019.

[26] Caroline Lemieux, Rohan Padhye, Koushik Sen, and Dawn Song. Perffuzz: Automatically generating pathological inputs. In *Proceedings of the 27th ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA 2018, page 254–265, New York, NY, USA, 2018. Association for Computing Machinery.

[27] H. Liang, L. Jiang, L. Ai, and J. Wei. Sequence directed hybrid fuzzing. In *2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 127–137, 2020.

[28] Andreas Löscher and Konstantinos Sagonas. Targeted property-based testing. In *Proceedings of the 26th ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA 2017, page 46–56, New York, NY, USA, 2017. Association for Computing Machinery.

[29] Chenyang Lyu, Shouling Ji, Chao Zhang, Yuwei Li, Wei-Han Lee, Yu Song, and Raheem Beyah. {MOPT}: Optimized mutation scheduling for fuzzers. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 1949–1966, 2019.

[30] A. Löscher and K. Sagonas. Automating targeted property-based testing. In *2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST)*, pages 70–80, 2018.

[31] Source Making. Design patterns. https://sourcemaking.com/design_patterns. Accessed: 2021-06-03.

[32] Valentin J. M. Manès, Soomin Kim, and Sang Kil Cha. Ankou: Guiding grey-box fuzzing towards combinatorial difference. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, ICSE '20, page 1024–1036, New York, NY, USA, 2020. Association for Computing Machinery.

[33] Phil McMinn. Search-based software test data generation: a survey. *Software testing, Verification and reliability*, 14(2):105–156, 2004.

[34] Xianya Mi, Baosheng Wang, Yong Tang, Pengfei Wang, and Bo Yu. Shfuzz: Selective hybrid fuzzing with branch scheduling based on binary instrumentation. *Applied Sciences*, 10(16):5449, 2020.

[35] Saahil Ognawala, Thomas Hutzelmann, Eirini Psallida, and Alexander Pretschner. Improving function coverage with munch: A hybrid fuzzing and directed symbolic execution approach. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, SAC '18, page 1475–1482, New York, NY, USA, 2018. Association for Computing Machinery.

[36] Rohan Padhye, Caroline Lemieux, and Koushik Sen. Jqf: Coverage-guided property-based testing in java. In *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA 2019, page 398–401, New York, NY, USA, 2019. Association for Computing Machinery.

[37] Rohan Padhye, Caroline Lemieux, Koushik Sen, Mike Papadakis, and Yves Le Traon. Semantic fuzzing with zest. In *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA 2019, page 329–340, New York, NY, USA, 2019. Association for Computing Machinery.

[38] Brian S Pak. Hybrid fuzz testing: Discovering software bugs via fuzzing and symbolic execution. *School of Computer Science Carnegie Mellon University*, 2012.

[39] Manolis Papadakis and Konstantinos Sagonas. A proper integration of types and function specifications with property-based testing. In *Proceedings of the 10th ACM SIGPLAN Workshop on Erlang*, Erlang '11, page 39–50, New York, NY, USA, 2011. Association for Computing Machinery.

[40] Theofilos Petsios, Jason Zhao, Angelos D. Keromytis, and Suman Jana. Slowfuzz: Automated domain-independent detection of algorithmic complexity vulnerabilities. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, page 2155–2168, New York, NY, USA, 2017. Association for Computing Machinery.

[41] Sanjay Rawat, Vivek Jain, Ashish Kumar, Lucian Cojocar, Cristiano Giuffrida, and Herbert Bos. Vuzzer: Application-aware evolutionary fuzzing. In *NDSS*, volume 17, pages 1–14, 2017.

[42] Sameer Reddy, Caroline Lemieux, Rohan Padhye, and Koushik Sen. Quickly generating diverse valid test inputs with reinforcement learning. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, ICSE '20, page 1410–1421, New York, NY, USA, 2020. Association for Computing Machinery.

[43] José Miguel Rojas, Gordon Fraser, and Andrea Arcuri. Seeding strategies in search-based unit test generation. *Software Testing, Verification and Reliability*, 26(5):366–401, 2016.

[44] Rashmi Rekha Sahoo and Mitrabinda Ray. Pso based test case generation for critical path using improved combined fitness function. *Journal of King Saud University - Computer and Information Sciences*, 32(4):479 – 490, 2020. Emerging Software Systems.

[45] Rashmi Rekha Sahoo, Mitrabinda Ray, and Gayatri Nayak. Test case generation based on search-based testing. In Debahuti Mishra, Rajkumar Buyya, Prasant Mohapatra, and Srikanta Patnaik, editors, *Intelligent and Cloud Computing*, pages 309–317, Singapore, 2021. Springer Singapore.

[46] Koushik Sen. Concolic testing. In *Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering*, pages 571–572, 2007.

[47] Koushik Sen, Darko Marinov, and Gul Agha. Cute: A concolic unit testing engine for c. *ACM SIGSOFT Software Engineering Notes*, 30(5):263–272, 2005.

[48] Hugo Sereno Ferreira, André Restivo, and Rui Gonçalves. Fluent check: A fluent property-based testing framework with existential quantification, 2021.

[49] Suhwan Song, Chengyu Song, Yeongjin Jang, and Byoungyoung Lee. *CrFuzz: Fuzzing Multi-Purpose Programs through Input Validation*, page 690–700. Association for Computing Machinery, New York, NY, USA, 2020.

[50] Larissa Stanberry. *Analysis of Variance*, pages 24–25. Springer New York, New York, NY, 2013.

[51] Nick Stephens, John Grosen, Christopher Salls, Andrew Dutcher, Ruoyu Wang, Jacopo Corbetta, Yan Shoshitaishvili, Christopher Kruegel, and Giovanni Vigna. Driller: Augmenting fuzzing through selective symbolic execution. In *NDSS*, volume 16, pages 1–16, 2016.

[52] Y. Wang, Z. Wu, Q. Wei, and Q. Wang. Epfuzzer: Improving hybrid fuzzing with hardest-to-reach branch prioritization. *KSII Transactions on Internet and Information Systems*, 14(9):3885–3906, 2020. cited By 0.

[53] Shengbo Yan, Chenlu Wu, Hang Li, Wei Shao, and Chunfu Jia. Pathafl: Path-coverage assisted fuzzing. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, ASIA CCS '20, page 598–609, New York, NY, USA, 2020. Association for Computing Machinery.

[54] W. You, X. Liu, S. Ma, D. Perry, X. Zhang, and B. Liang. Slf: Fuzzing without valid seed inputs. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pages 712–723, 2019.

[55] Insu Yun, Sangho Lee, Meng Xu, Yeongjin Jang, and Taesoo Kim. QSYM : A practical concolic execution engine tailored for hybrid fuzzing. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 745–761, Baltimore, MD, August 2018. USENIX Association.

[56] L. Zhao, P. Cao, Y. Duan, H. Yin, and J. Xuan. Probabilistic path prioritization for hybrid fuzzing. *IEEE Transactions on Dependable and Secure Computing*, pages 1–1, 2020.