Adequate vs. Inadequate Test Suite Reduction Approaches

Carmen Coviello[a,*], Simone Romano[b], Giuseppe Scanniello[a], Alessandro Marchetto[c], Anna Corazza[d], Giuliano Antoniol[e]

[a] *University of Basilicata, Potenza, Italy*
[b] *University of Bari, Bari, Italy*
[c] *Independent Researcher*
[d] *University of Naples "Federico II", Naples, Italy*
[e] *Polytechnique Montreal, Montreal, Canada*

Abstract

Context: Regression testing is an important activity that allows ensuring the correct behavior of a system after changes. As the system grows, the time and resources to perform regression testing increase. Test Suite Reduction (TSR) approaches aim to speed up regression testing by removing obsolete or redundant test cases. These approaches can be classified as adequate or inadequate. Adequate TSR approaches reduce test suites and completely preserve test requirements (*e.g.,* covered statements) of the original test suites. Inadequate TSR approaches do not preserve test requirements. The percentage of satisfied test requirements indicates the inadequacy level.

Objective: We compare some state-of-the-art adequate and inadequate TSR approaches with respect to the size of reduced test suites and their fault-detection capability. We aim to increase our body of knowledge on TSR approaches by comparing: *(i)* well-known traditional adequate TSR approaches; *(ii)* their inadequate variants; and *(iii)* several variants of a novel Clustering-Based (CB) approach for (adequate and inadequate) TSR.

Method: We conducted an experiment to compare adequate and inadequate

*Corresponding author
Email addresses: carmen.coviello@unibas.it (Carmen Coviello), simone.romano@uniba.it (Simone Romano), giuseppe.scanniello@unibas.it (Giuseppe Scanniello), alex.marchetto@gmail.com (Alessandro Marchetto), anna.corazza@unina.it (Anna Corazza), antoniol@ieee.org (Giuliano Antoniol)

TSR approaches. This comparison is founded on a public dataset containing information on real faults.

Results: The most important findings from our experiment can be summarized as follows: *(i)* there is not an inadequate TSR approach that outperforms the others; *(ii)* some inadequate variants of the CB approach, and few traditional inadequate approaches, outperform the adequate ones in terms of reduction in test suite size with a negligible effect on fault-detection capability; and *(iii)* the CB approach is less sensitive than the other inadequate approaches, that is, variations in the inadequacy level have small effect on reduction in test suite size and on loss in fault-detection capability.

Conclusions: These findings imply that inadequate TSR approaches and especially the CB approach might be appealing because they lead to a greater reduction in test suite size (with respect to the adequate ones) at the expense of a small loss in fault-detection capability.

Keywords: Adequate Test Suite Reduction, Clustering, Inadequate Test Suite Reduction, Regression Testing, Test Suite Reduction

---

1. Introduction

Regression testing is conducted after changes are made to a given System Under Test (SUT) to ensure that these changes do not alter its observed behavior with respect to the expected one [1]. The simplest regression testing strategy, named *Retest-all*, consists in re-executing, on the changed version of the SUT, the entire Test Suite (TS) [2]. However, as a system evolves, its TS tends to grow in size. Therefore, Retest-all might not be a viable option because it might require too much time and/or too many resources (*e.g.,* hardware) [3]. To deal with this problem, a number of approaches have been proposed; they can be divided into three main classes: *(i)* Test Suite Reduction (TSR); *(ii)* regression test case selection; and *(iii)* test case prioritization [1]. Both TSR and regression test case selection approaches reduce the size of a TS, but in a different way and with a different aim. In particular, TSR approaches allow reducing the TS

size by removing redundant or obsolete test cases [2], instead a regression test case selection approach re-runs a subset of the TS that executes changed parts (or parts affected by changes) of the SUT [4, 5]. Test case selection approaches also differ from TSR approaches because they need information on the past changes developers implemented in the source code. TSR is sometimes called TS minimization when the elimination of test cases is not permanent; however, these two concepts are often used in an interchangeable way [1]. Finally, test case prioritization approaches sort test cases according to some criteria (*e.g.,* those test cases covering more source code statements are placed before) [6, 7]. The assumption of availability of some a-priori knowledge about the SUT and its TS (*e.g.,* code covered by the TS) is common to these three classes of approaches.

TSR approaches can be classified as adequate or inadequate/non-adequate [8]. Adequate TSR approaches reduce the TS so that the new TS still satisfies the test requirements of the original TS. For instance, let statement coverage be the kind of test requirement, an adequate TSR approach produces a reduced TS covering the same statements of the original TS. A TSR approach is inadequate when the reduced TS does not preserve the test requirements of the original TS. Shi *et al.* proposed such a kind of approach for the first time in [8], where the inadequacy level indicates the percentage of the test requirements that must be satisfied by the reduced TS. It is easy to grasp that adequate TSR approaches are characterized by an inadequacy level equal to 100% since they reduce TSs so that the new TSs must satisfy all the test requirements. Inadequate approaches are appealing if they lead to a greater reduction in TS size at the expense of a negligible (or even better null) loss in fault-detection capability. The potential of such a kind of approaches has been scarcely studied so far [8, 9, 10]. In fact, a greater reduction in the size of TSs could sensibly affect the fault-detection capability of the reduced TSs. Then, empirical evidence is needed to increase our body of knowledge on the use of inadequate approaches to deal with TSR.

In this paper, we present the results of an empirical study (*i.e.,* an experiment) aimed at comparing: *(i)* well-known traditional adequate TSR approaches; *(ii)* their inadequate variants; and *(iii)* a Clustering-Based (CB) TSR

approach in its adequate and inadequate variants. In particular, we studied the following traditional adequate TSR approaches: Harrold-Gupta-Soffa [11], Greedy [12], Delayed Greedy [13], 2-Optimal [12], GE [14], GRE [14], and ILP [15]. For these approaches, except for the last one, we also studied their inadequate variants. To transform an adequate approach into its inadequate variant, we applied the general approach proposed by Shi *et al.* [8]. The CB approach shares the same underlying idea as the approach presented by Coviello *et al.* [9]. That is, test cases that are redundant with respect to a given kind of test requirement (*i.e.,* statement coverage in this paper) are grouped into the same cluster. Similar test cases are considered redundant if they cover nearly the same statements. To estimate test cases redundancy, we compute the similarity among test cases through a number of different measures. The use of one of these measures (seven, in total) in the CB approach identifies an instance of that approach (or simply CB instance from here onwards). In other words, the CB approach is general and we consider seven different instances of that approach, each of which is characterized by a different measure to estimate test case redundancy. A given CB instance identifies a reduced TS so that it contains a test case for each identified cluster, namely the most representative test case of that cluster, which is the test case covering the largest number of statements in the cluster. Such a choice for the most representative test case relies on the postulation that the statements covered by this test case include all (or nearly) the statements that the other test cases in the cluster cover. To some extent, we should be guaranteed that the fault-detection capability of the most representative test case of a cluster is the same as all the test cases in that cluster.

The empirical validation presented in this paper was performed on 19 experimental objects from a public dataset, *i.e.,* SIR (Software-artifact Infrastructure Repository) [16]. We compared and contrasted the different approaches in terms of reduction of the original TS size versus their loss in fault-detection capability.

Summarizing, this paper with respect to that by Coviello *et al.* [9] makes the following new contributions:

- The CB approach studied in this paper produces reduced TSs based on a given inadequacy level, rather than on a tuning value of the clustering algorithm as done in [9]. Although this difference could appear not overly surprising, it has a number of implications. For example, it allows: *(i)* comparing our solution with the above-mentioned inadequate approaches; *(ii)* the practitioner to take a more informed decision when reducing a TS because she is aware of the amount of test requirements the new TS does not satisfy; and *(iii)* the CB approach to behave in an adequate manner by choosing an inadequacy level equal to 100%.

- A more extensive comparison among inadequate TSR approaches in terms of reduction in TS size and loss in fault-detection capability. That is, we took into account the inadequate variants of Harrold-Gupta-Soffa, Greedy, Delayed Greedy, 2-Optimal, GE, and GRE.

- A comparison between the best inadequate TSR approach/es with the following adequate approaches: Harrold-Gupta-Soffa, Greedy, Delayed Greedy, 2-Optimal, GE, GRE, ILP and the adequate variants of the CB approach. This comparison is important to understand if and when inadequate TSR approaches might be competitive with respect to their adequate counterparts.

- An analysis to investigate the effects of the inadequacy levels on TS reductions (*i.e.,* a sensitive analysis). The goal of this analysis is to identify the TSR approaches less sensitive to the inadequacy level. The lower the sensitiveness, the better the approach is, because variations in the inadequacy level slight affect the reduction in both TS size and fault-detection capability of reduced TSs.

Paper Structure. In Section 2, we highlight related work and background information. We describe the CB approach in Section 3. The design of our investigation and the results are provided in Section 4 and Section 5, respectively. Final remarks conclude the paper in Section 6.

2. Related Work and Background

In the following of this section, we first provide the formal definition of the TSR problem by Rothermel *et al.* [2] and then highlight research related to this problem for both its variants: adequate and inadequate. We then discuss approaches that apply clustering to deal with regression testing and conclude by providing background information to understand our proposal.

2.1. Test suite reduction

According to Rothermel *et al.* [2] the TSR problem can be stated as follows: ***Given:*** A TS named T, a set of test requirements $r_1, ..., r_n$, which must be satisfied to provide the desired test coverage of the program, and subsets of T, $T_1, ..., T_n$, one associated with each of the r_is such that any one of the test cases t_j belonging to T_i can be used to test requirement r_i.
Problem: Find a representative set, T', of test cases from T that satisfies all of the r_is.
Many TSR approaches have been proposed in the literature (*e.g.,* [11, 12, 13, 17, 18, 19, 20, 21]). Traditional approaches are based on the aforementioned definition and aim to reduce the size of TSs satisfying all the test requirements. Such kind of approaches is called *adequate* [8]. Researchers have also proposed TSR approaches that relax the constraint of satisfying all the test requirements and have named them *inadequate* [8].

2.1.1. Adequate TSR Approaches

Many of the TSR approaches proposed in the literature [1] use heuristics to identify and discard redundant test cases. If a test case satisfies a subset of the test requirements of another test case, it is considered as redundant. Traditional approaches tend to focus on one kind of test requirement (*e.g.,* code coverage). For example, Harrold *et al.* [11] proposed a heuristic-based approach to identify a representative set of the original TS that satisfies all the test requirements. This approach was named Harrold-Gupta-Soffa (HGS). HGS first analyzes sets of test cases of cardinality one and then chooses the set that satisfies more

test requirements. Then, sets of test cases of cardinality two are analyzed and the set that satisfies more requirements is chosen. This process is iteratively performed until all the test requirements are satisfied. Li *et al.* [12] proposed Greedy (GRD), an approach based on a greedy algorithm to prioritize test cases. It was also applied to reduce TSs [22]. GRD selects each time a test case, among those available, that satisfies the higher number of test requirements that are yet unsatisfied. The selection process concludes when the selected test cases satisfy all the test requirements. The pseudo-code of the Greedy algorithm is shown in Algorithm 1. Li *et al.* [12] also introduced the 2-Optimal (2OPT) approach. It is based on the 2-Optimal algorithm, which represents an instance of the K-Optimal algorithm with $K = 2$. 2OPT selects the first two test cases that together satisfy the largest number of test requirements. Tallam and Gupta [13] proposed a variant of GRD, named Delayed Greedy (DGR). It is based on the following two steps: *(i)* if a set of test requirements satisfied by a test case t_i is a super-set of a set of requirements satisfied by another test case t_j, then t_j is removed from the TS; and *(ii)* if a set of test cases satisfying a requirement r_i is a subset of a set of test cases satisfying r_j, then r_i is removed from the unsatisfied requirements. Chen and Lau [14] introduced two heuristic-based approaches, GE and GRE, which both represent a variant of GRD. The authors defined the opposite of the redundant test case as the essential test case: if a test requirement r_i can be satisfied by only one test case t_i, then t_i is an essential test case. GE and GRE can be summarized as follows [1]: GE first selects all essential test cases in the TS, then it applies the greedy algorithm for the unsatisfied test requirements; while GRE first removes all redundant test cases in the TS, then applies GE on the reduced TS. Black *et al.* [23] by using a binary ILP representation of the TS minimization problem, developed a model able to compute optimal minimized TS. The formulation of the model follows:

$$\textbf{\textit{Minimize:}} \quad \sum_{j=1}^{|TS|} x_j$$

$$\textbf{\textit{Subject to:}} \quad \sum_{j=1}^{|TS|} a_{ij} x_j \geq 1, i = 1, ..., |S|$$

$$x_j \text{ binary for } j = 1, ..., |TS|$$

Algorithm 1 Adequate Greedy algorithm

1: **procedure** AdequateGreedyTSR
2: *Input*:
3: $T \leftarrow$ The set of test cases of the SUT
4: $R \leftarrow$ The set of test requirements in the SUT
5: $S \leftarrow\ S = \{(t, r) \mid t$ satisfies r, $t \in T$, $r \in R\}$
6: *Output*:
7: $T' \leftarrow$ The representative set T
8: *begin*:
9: $T' = \emptyset$
10: *loop*:
11: **while** $R \neq \emptyset$ **do**
12: $t =$ the test case that satisfies the maximum number of requirements.
13: $T' = T' \cup \{t\}$.
14: $T = T - \{t\}$.
15: $R = R - \{r \mid (t, r) \in S\}$.
16: **end**
17: **end procedure**

where S represents the set of test requirements to cover. Each element a_{ij} of the constraint coefficient matrix is equal to 1 if the test case j covers the test requirement i, 0 otherwise. This model allows reducing the TS if two (or more) test cases cover the same subset of test requirements. Thus, all but one will be eliminated because they are redundant. The authors also proposed a bi-criteria binary ILP model in order to take into account the ability of each test case to reveal faults.

2.1.2. Inadequate TSR Approaches

Shi *et al.* [8] presented a study on traditional adequate TSR approaches (*e.g.,* HGS or GRD) and their inadequate variants to verify the benefits that can be achieved when relaxing the constraint of covering test requirements. The

Algorithm 2 Inadequate Greedy algorithm

1: **procedure** INADEQUATEGREEDYTSR
2: *Input*:
3: $T \leftarrow$ The set of test cases of the SUT
4: $R \leftarrow$ The set of test requirements in the SUT
5: $S \leftarrow\ S = \{(t, r) \mid t \text{ satisfies } r,\ t \in T,\ r \in R\}$
6: $l \leftarrow$ The inadequacy level
7: *Output*:
8: $T' \leftarrow$ The representative set T
9: *begin*:
10: $T' = \emptyset$
11: $Threshold = \mid R \mid - \frac{\mid R \mid \times l}{100}$
12: *loop*:
13: **while** $\mid R \mid > Threshold$ **do**
14: $t =$ the test case that satisfies the maximum number of requirements.
15: $T' = T' \cup \{t\}$.
16: $T = T - \{t\}$.
17: $R = R - \{r \mid (t, r) \in S\}$.
18: **end**
19: **end procedure**

studied inadequate variants, instead of producing reduced TSs that satisfy 100% of the test requirements, produce reductions satisfying a fixed percentage of the test requirements. More formally, an inadequate TSR approach can be defined as an algorithm that requires three inputs: *(i)* a function that returns the set of satisfied requirements for a given TS; *(ii)* the original TS denoted by T; and *(iii)* the inadequacy level, namely the percentage $0 \leq l \leq 100$ of requirements that must be satisfied [8]. The algorithm returns a reduced TS, $T' \subseteq T$, that satisfies (at least) $l\%$ of the requirements satisfied by T. In case $l = 100\%$, the approach is adequate. By way of example, the pseudo-code of the inadequate variant of GRD is shown in Algorithm 2. It is easy to observe that the only

instruction changed with respect to Algorithm 1 is the while loop condition (*i.e.,* line 11 in Algorithm 1 and line 13 in Algorithm 2). In Algorithm 2, given as input the inadequacy level (in percentage), the while loop goes on until the number of unsatisfied test requirements is greater than the desired threshold. Similarly, the other traditional adequate approaches are turned into inadequate ones. Shi *et al.* [8] observed that for the adequate approaches, the size of the reduced TSs is (median of) 62.9% of the size of original TSs. Moreover, by dropping down to 95% of the test requirements (*i.e.,* losing 5% of the covered statements), the reduction in the size of the reduced TSs can increase of 17.14%. Several are the differences between the work by Shi *et al.* [8] and that presented in this paper. We summarize the most important ones as follows: *(i)* our approach uses a clustering algorithm and several measures to compare test cases in terms of covered statements and *(ii)* the experimental assessment is founded on SIR.

Coviello *et al.* [9] proposed a clustering-based approach for inadequate TSR, whose underlying process had been instantiated six times. To identify clusters and then reduce the original TS, these instances take as input a parameter of the used clustering algorithm (*i.e.,* the cut level of the dendrogram) and not the inadequacy level. This implies that, given two SUTs and the same cut level, the approach can identify two reduced TSs with a different inadequacy level. That is, there is not a direct correlation between the cut level and the inadequacy level. Coviello *et al.* compared well-known traditional adequate TSR approaches with the CB approach. In their experiment, they found that the CB approach reduces more the size of the TSs than the traditional adequate approaches. The main differences with the paper by Coviello *et al.* can be summarized as follows: *(i)* the identification of the reduced TSs is guided by the inadequacy level, rather than a tuning parameter of the clustering algorithm; *(ii)* an investigation of a new instance based on new measure to compare test cases; *(iii)* a comparison between adequate and inadequate variants of well-known traditional TSR approaches; and *(iv)* an analysis of the sensitiveness of inadequate approaches.

More recently, Coviello *et al.* [10] investigated the use of three kinds of test

requirements (*i.e.,* statement, method, and class coverages) when reducing TSs by means of adequate and inadequate TSR approaches. The results suggest that inadequate approaches behave similarly when considering statement and method coverages as the test requirements (also the adequate approaches perform similarly when considering these two kinds of test requirements). Indeed, the use of statement coverage as kind of test requirement allows obtaining a slightly better trade-off between reduction in TS size and loss in fault-detection capability. The use of class coverage as kind of test requirement allows TSR approaches to reduce more the size of the TSs. However, there could be a price to pay in terms of loss in fault-detection capability of the reduced TSs. On the basis of these outcomes, we consider in this paper the statement coverage (rather than method or class coverage) as the kind of test requirement.

From here onwards, to distinguish between an adequate TSR approach and its inadequate variant, we associate the subscript A and I, respectively, to the label identifying that approach. For example, HGS_A refers to the traditional (adequate) Harrold-Gupta-Soffa approach, while HGS_I indicates its inadequate variant.

2.2. Clustering and Regression Testing

In the regression testing field, clustering has been used to group similar test cases leveraging different kinds of test requirements (*e.g.,* code coverage and execution cost) [24, 25, 26, 27, 28, 29]. Similar test cases within a cluster are considered redundant. TSs are obtained by selecting one (or more) test case per cluster. For example, Parsa *et al.* [24] and Khalilian and Parsa [25] proposed an approach to group test cases based on the similarity of their execution profiles. They applied a heuristic for sampling test cases from clusters by ensuring the same coverage of the original TS. That is, this approach can be classified as adequate. To cluster test cases the approach uses the CLOPE clustering algorithm [30].

Carlson *et al.* [28] proposed a prioritization approach based on hierarchical agglomerative clustering with the average-link criterion. The similarity between

test cases is based on four kinds of test requirements considered separately: code coverage, code complexity, fault history, and combination of code complexity and fault history. They used the Euclidean distance to compute the similarity among test cases for each kind of test requirements. The use of code complexity seemed to produce better prioritization of test cases. Differently, Arafeen and Do [29] used three types of information to prioritize test cases: functional requirements, code complexity, and importance of functional requirement. Prasad *et al.* [27] designed an approach based on a method coverage matrix; it groups test cases with a hierarchical clustering algorithm using Hamming distance. Each cluster is then analyzed to identify subsets of redundant test cases. Redundant test cases are identified by analyzing the similarity among: *(i)* functional flows, *(ii)* line coverage, and *(iii)* branch coverage. The approach is adequate since it does not relax test requirements constraint.

The main differences between our research and the above-mentioned approaches are related to the empirical assessment and the approach. As for the assessment, our primary goal was to increase the body of knowledge on the effectiveness of inadequate approaches to deal with TSR. As for the clustering of test cases, the main difference can be summarized as follows: the CB approach can use different measures (*e.g.,* Euclidean distance or Cosine dissimilarity) to estimate the similarity among the test cases and it can be configured to fulfill the desired inadequacy level, where inadequacy level equal to 100% allow it to behave like an adequate approach.

2.3. Clustering and Dissimilarity/Distance Measures

Clustering algorithms group entities into clusters (*i.e.,* groups) so that entities within a cluster are similar. Entities between different clusters are as much as possible dissimilar [31]. Therefore, in addition to a clustering algorithm, a distance/dissimilarity measure has to be chosen. This measure influences clustering results because it estimates the extent to which two entities are similar/dissimilar [31]. Different kinds of clustering algorithms have been proposed in the literature. Hierarchical Agglomerative Clustering (HAC) is a kind of

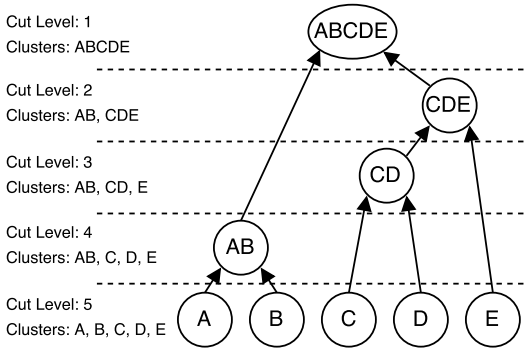

Figure 1: A sample dendrogram.

clustering algorithm that treats each entity as a singleton cluster at the outset and then successively merges (or agglomerates) pairs of clusters until all clusters are merged into a single cluster. At each step the most similar clusters are merged. There are different criteria to compute the similarity of two clusters. The average-link criterion evaluates the similarity of two clusters based on all the similarities between the entities in these clusters. That is, the pair of clusters with the highest inter-cluster cohesion (*e.g.,* it could be computed as the average similarity of all the pairs of entities in the two clusters) is merged at each iteration. This avoids the pitfalls of the single-link (the similarity of two clusters is the similarity of their most similar entities) and complete-link (the similarity of two clusters is the similarity of their most dissimilar entities) criteria that equate cluster similarity with the similarity of a single pair of entities [31].

The arrangement of the clusters produced by a HAC algorithm can be visualized as a tree structure named dendrogram (see Figure 1). A HAC algorithm does not require a pre-specified number of clusters [31]. However, in some applications, we want a partition of disjoint clusters. In these cases, the dendrogram needs to be cut at some level. This is called cut level of the dendrogram and clearly influences clustering results. In Figure 1, five different cut levels are shown as well as the obtained clusters. For example, choosing three as cut level, the algorithm identifies two clusters containing two entities each and one singleton cluster: (A,B), (C,D), and (E). The higher the cut levels, the greater

the number of clusters is. HAC is preferred in several applications because its output is deterministic.

Several distance and similarity measures are amenable for the use in clustering. In the following, we report and briefly describe the definition of the measures used to create the different instances of the CB approach presented in this research work.

Euclidean Distance. Let $\vec{x} = (x_1, x_2, ..., x_n)$ and $\vec{y} = (y_1, y_2, ..., y_n)$ be two vectors, the Euclidean distance between them is defined as:

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{n} (y_\mathrm{i} - x_\mathrm{i})^2} \tag{1}$$

Cosine (Dis)similarity. Let $\vec{x}$ and $\vec{y}$ be two vectors, the Cosine (dis)similarity between them is defined as:

$$d(\vec{x}, \vec{y}) = 1 - \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \|\vec{y}\|} \tag{2}$$

Jaccard-Based Dissimilarity. Let $\mathbf{A}$ and $\mathbf{B}$ be two sets, the Jaccard-based dissimilarity between them is equal to one minus the Jaccard coefficient:

$$d(\mathbf{A}, \mathbf{B}) = 1 - \frac{|\mathbf{A} \cap \mathbf{B}|}{|\mathbf{A} \cup \mathbf{B}|} \tag{3}$$

Hamming Distance. Let $\vec{x}$ and $\vec{y}$ be two vectors of the same length, the Hamming distance between them is the number of elements in which they differ. For example, given $\vec{x} = (\mathbf{1}, 1, 0, \mathbf{1})$ and $\vec{y} = (\mathbf{0}, 1, 0, \mathbf{0})$ their Hamming distance is two.

Levenshtein Edit Distance. Let a and b be two sequences of characters (*i.e.,* two strings), the Levenshtein edit distance (or simply Levenshtein distance) between them is the minimum number of operations required to transform a into b. The operations that can be performed on a sequence are: *(i)* add a new character; *(ii)* delete a character; and *(iii)* substitute a character with another.

K-Based Dissimilarity. The K-based dissimilarity, namely Cohen's Kappa index, measures the degree of agreement between two raters, who classify items over two or more categories. Let p_o be the observed proportion of agreement and

p_e the agreement expected just by chance, the K-based dissimilarity is defined as follows:

$$k = 1 - \frac{p_o - p_e}{1 - p_e} \tag{4}$$

String Kernels-Based Dissimilarity. The kernel is a function that computes the inner product between two vectors in their space [32]. In the machine learning field, a string kernel is a particular type of kernel, which operates on strings to measure the similarity between them. Giving two strings a and b, the standard formula to compute the string kernel function is:

$$k(a, b) = \sum_{s \in A^+} num_s(a)\ num_s(b)\ \lambda_s \tag{5}$$

where A^+ represents the set of non-empty substrings; λ_s is a chosen weight or decay factor; and $num_s(x)$ is the number of the substring s occurrences in x (where x is either a or b). In our work, we consider the *bounded-range kernel* instantiation of the kernel function, where $\lambda_s = 0$ for all $|s| > n$ and n is fixed. This means that, only matching substrings of length less or equal n are considered. The string kernels-based dissimilarity between two strings a and b is computed as follows:

$$d(a, b) = 1 - k^*(a, b) \tag{6}$$

where $k^*(a, b)$ is the normalized string kernel function that returns values in between 0 and 1. In our work, we consider the normalized version of the bounded-range kernel.

3. Clustering-based Approach

In this section, we describe the process underlying the CB approach. We conclude with a running example to show how the CB approach works.

3.1. Process

The UML activity diagram with object flow in Figure 2 shows the process underlying the CB approach. Rounded rectangles represent the phases of the

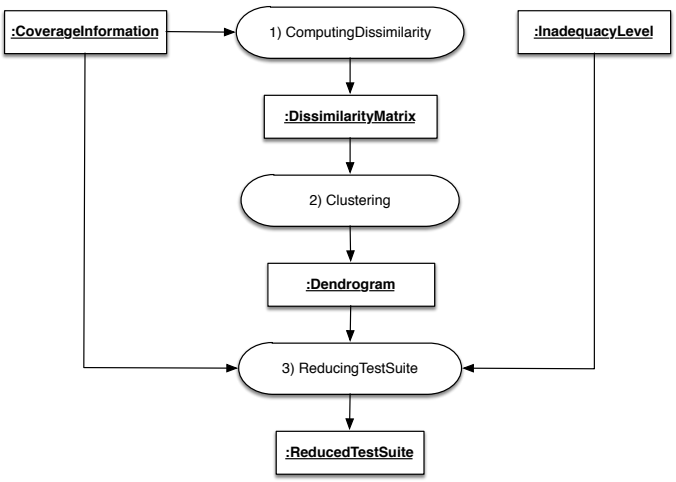

Figure 2: Process underlying the CB approach for TSR.

process, while rectangles are the objects produced/consumed in these phases. In a nutshell, the CB approach first computes the dissimilarity between pairs of test cases based on a given distance/dissimilarity measure. Test cases that are more similar one another (*i.e.,* they can be considered redundant since they satisfy nearly the same test requirements) are grouped into the same cluster through a HAC algorithm. The reduced TS will contain a test case for each cluster that represent the most representative test case of the cluster. The test case satisfying the highest number of test requirements in a cluster is the most representative of that cluster. A description of the phases of the process follows:

1. **ComputingDissimilarity.** This phase computes the distance/dissimilarity (simply dissimilarity from here onwards) between all the pairs of test cases and produces a `DissimilarityMatrix`. This matrix is m x m, where m is the number of test cases in the TS. Each entry of the matrix contains the value of the dissimilarity between the corresponding pair of test cases. To use the dissimilarity measures mentioned in Section 2.3 and to compute the dissimilarity among test cases, it is needed to encode the test requirements that each test case satisfies. As for Euclidean distance, cosine (dis)similarity, and Hamming distance, we computed a binary vector

$\vec{x_i}$ for each test case t_i. The length of the binary vector is equal to the number of test requirements (the number of statements covered by the entire TS). Each entry of $\vec{x_i}$ is associated to a test requirement r_j—an entry equal to 1 means that t_i satisfies that test requirement r_j; an entry equal to 0 means that t_i does not satisfy r_j.

The encoding for the K-based dissimilarity passes from binary vectors as well. However, a further step is needed to obtain p_e and p_o. In particular, for each pair of test cases t_1 and t_2, we build the joined binary vector $\vec{x_{1,2}}$ (*i.e.,* the concatenation of $\vec{x_1}$ and $\vec{x_2}$). Then we computed p_{one} and p_{zero} (*i.e.,* the total number of ones and zeros in $\vec{x_{1,2}}$ divided by the length of $\vec{x_{1,2}}$, respectively) to obtain p_e according to the following formula:

$$p_e = p_{one}^2 + p_{zero}^2 \tag{7}$$

On the other hand, p_o is computed as the number of test requirements that both t_1 and t_2 satisfy plus the number of the requirements that both t_1 and t_2 do not satisfy by the length of $\vec{x_{1,2}}$.

As for the Jaccard-based dissimilarity, we represent the test requirements that a test case t_i satisfies as a set $\vec{A_i}$. That is, if t_i satisfies the requirement r_j, then $r_j \in \vec{A_i}$.

Finally, for the Levenshtein distance and String Kernels-based dissimilarity, the test requirements satisfied by a test case t_i are encoded in a string s_i. That is, each statement r_j can be seen as unique character.[1]

2. **Clustering.** Test cases are grouped using a clustering algorithm. We opted for the HAC algorithm with the average-link criterion (the motivations behind this decision are those sketched in Section 2.3). The adopted

[1] In the implementation of our tool prototype, we had to deal with encoding issues because the number of characters is limited and this number depends on the char-set used for the encoding. To deal with this limitation, we encode the covered statements of a test case each time it is compared with another test case.

clustering algorithm builds a `Dendrogram` using the `DissimilarityMatrix` produced in the previous phase.

3. **ReducingTestSuite.** Given the `Dendrogram`, a cut level produces a set of clusters. For each cluster, we select the most representative test case. We retain the test case that satisfies the largest number of test requirements as the most representative of the cluster. The rationale behind the choice of the most representative test case relies on the postulation that the requirements that a representative test case satisfies are roughly a superset of the requirements that the other test cases in the cluster satisfy. That is, we postulate that the fault-detection capability of the most representative test case is the same as the fault-detection capability of the test cases in that cluster. If more than one test case satisfies the same number of test requirements, we randomly select one among them. The set of most representative test cases composes the reduced TS. Once a cut level is selected, we obtain a reduced TS that is characterized by an inadequacy level. If that TS does not satisfy all the test requirements of the original TS the inadequacy level is less than 100%. To achieve a reduction with the desired inadequacy level (*i.e.,* one of the input of the process), we consider all the possible cut levels and choose the one that allows obtaining the desired inadequacy level. This is done via the algorithm shown in Algorithm 3, where l is the desired inadequacy level, D is the dendrogram, $lowerLevel$ is the minimum (in the first iteration) level of the dendrogram and $higherLevel$ is the maximum (in the first iteration) level of the dendrogram. The computational complexity of this algorithm is logarithmic with respect to the height of the dendrogram. For l equals to 100% the CB approach behaves like an adequate TSR approach. On the other hand, if l is equals to 0% it means that the approach selects the test case of the TS of the SUT that covers the largest number of statements.

The CB approach was implemented in a prototype of a supporting tool named CUTER. It is an Eclipse plug-in that allows reducing (JUnit) TS of

Algorithm 3 TS Reduction algorithm

```
1: procedure CBReduction
2: Input:
3:     l ← The fixed inadequacy level
4:     D ← The dendrogram
5:     lowerLevel ← The lower level of the dendrogram to be analyzed
6:     higherLevel ← The higher level of the dendrogram to be analyzed
7: Output:
8:     T_h ← Reduced TS
9: begin:
10:    h ← ⌈(higherLevel − lowerLevel)/2⌉
11:    l_Th ← The inadequacy level of T_h by cutting D at level h
12: start:
13:    if (l_Th == l OR h == lowerLevel OR h == higherLevel) then return
       T_h
14:    else if l_Th > l then
15:        return CBReduction(l, D, lowerLevel, h − 1)
16:    else
17:        return CBReduction(l, D, h + 1, higherLevel)
18: end procedure
```

software systems written in Java. Further information on CUTER can be found in the paper by Coviello *et al.* [33]. CUTER is also available for download.[2]

3.2. Running example of the approach application

Let $T = \{t_1, t_2, t_3\}$ be the TS of a SUT and let r_1, r_2, and r_3 be the test requirements, we show in the matrix of Table 1 the test requirements satisfied by each test case in T (*e.g.*, the test cases t_1 and t_2 satisfy the requirements r_1 and r_2, but they do not satisfy r_3). To encode the test requirements that each

[2] www2.unibas.it/sromano/CUTER.html

test case satisfies, we applied the following strategies:

- As for the Euclidean distance, cosine (dis)similarity, Hamming distance, and K-based dissimilarity, we build the binary vectors for t_1, t_2, and t_3, namely: $\vec{x_1} = \vec{x_2} = (1, 1, 0)$ and $\vec{x_3} = (0, 0, 1)$.

- The K-based dissimilarity requires a further encoding, namely a joined binary vector for each pair of test cases (*e.g.,* $\vec{x_{1,2}} = (1, 1, 0, 1, 1, 0)$ is the joined binary vector of $\vec{x_1}$ and $\vec{x_2}$). Then we compute $p_{one} = 4/6$ and $p_{zero} = 2/6$ (*i.e.,* the number of ones and zeros by the length of $\vec{x_{1,2}}$, respectively) to obtain $p_e = 0.5$. (*i.e.,* the expected probability of the agreement just by chance). On the other hand, p_o (*i.e.,* the number of agreement by the test requirements) is equal to 1.

- As for the Jaccard-based dissimilarity, we build the sets of test requirements that t_1, t_2, and t_3 satisfy: $\vec{A_1} = \vec{A_2} = \{r_1, r_2\}$, $\vec{A_3} = \{r_3\}$.

- As for the String Kernels-Based dissimilarity and Levenshtein edit distance, each test requirement is encoded as a unique character (*i.e.,* $r_1 = A$, $r_2 = B$, and $r_3 = C$). Then, for each test case, the satisfied requirements are represented as a string obtained by concatenating these characters: $s_1 = s_2 = AB$ and $s_3 = C$.

Since the running-example is illustrative, we will focus on the instance based on the Levenshtein edit distance. The other instances of the CB approach work in a similar fashion. In Table 2, we report the `Dissimilarity Matrix` obtained from Table 1 by computing the Levenshtein edit distance (*e.g.,* the distance between t_1 and t_3 is 3). The dendrogram D in Figure 3 is built from that matrix. To obtain a reduction of the TS with 66% as the inadequacy level, we run Algorithm 3 with the following input data: $l = 66\%$; D; $lowerLevel = 1$; $higherLevel = 3$. Since Algorithm 3 cuts the D dendrogram at level 1, the reduced TS contains a single test case randomly chosen between t_1 and t_2, *i.e.,* the test cases satisfying the highest number of test requirements. The inadequacy level of the reduced TS could assume a value different from that

Table 1: Sample matrix showing the test requirements (*i.e.*, r_1, r_2, and r_3) each test case (*i.e.*, t_1, t_2, and t_3) satisfies.

	r_1	r_2	r_3
t_1	•	•	
t_2	•	•	
t_3			•

Table 2: Dissimilarity matrix built by using the Levenshtein edit distance based on the information on test requirements shown in Table 1.

	t_1	t_2	t_3
t_1	0	0	3
t_2	0	0	3
t_3	3	3	0

specified as the input. Our approach identifies a TS reduction whose inadequacy level is as the closest as possible to the desired one.

4. Empirical Study

We followed the guidelines by Wohlin *et al.* [34] to plan and conduct our experiment. The planning of this experiment is shown in the following subsections. The used template is based on that proposed by Wohlin *et al.* [34].

4.1. Definition and Context

The goal of our experiment, using the GQM (Goal Question Metrics) template by Basili *et al.* [35], was defined as follows:

> **Analyze** adequate and inadequate TSR approaches
> **for the purpose of** comparing them
> **with respect to** reduction in TS size and loss in fault-detection capability of the reduced TS
> **from the point of view** of the researcher and the practitioner

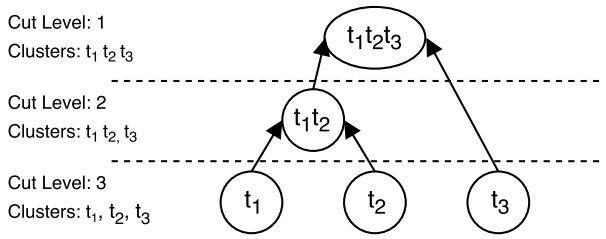

Figure 3: Dendrogram built by using a HAC algorithm with average-link criterion based on the dissimilarity matrix reported in Table 2.

in the context of open-source object-oriented software systems implemented in Java.

Given the experiment goal, we formulated and investigated the following Research Questions (RQs):

RQ1 - Is there an inadequate TSR approach that outperforms the other (inadequate) TSR approaches?

GOAL - We aim to study inadequate approaches in terms of their capability to reduce TSs and the capability of the reduced TSs to reveal faults. The best approach is the one that reduces more the TSs without a significant effect on the fault-detection capability of the reduced TSs. To perform a fair comparison among inadequate TSR approaches, the same inadequacy level was used and the obtained TSs were compared with respect to the size of the reduced TSs and their fault-detection capability. We fixed the inadequacy level at 95%. We chose this value because it allows reducing the size of the TSs with the smallest loss in their fault-detection capability. Further details can be found in Appendix A, where the results obtained by applying inadequacy levels in between 60% and 90% are shown and discussed.

RQ2 - Does the best inadequate TSR approach(es) outperform adequate approaches, in terms of reductions in TS size at the cost of a negligible effect on the reduction of fault-detection capability?

GOAL - We defined this RQ to investigate if the best inadequate TSR approach(es) might represent a viable alternative to the adequate ones. The best inadequate approach is identified in the answer to RQ1. Inadequate approach/es could be considered a viable alternative to adequate ones when they lead to a greater reduction in TS size at the cost of a negligible effect on fault-detection capability.

RQ3 - Which is the inadequate TSR approach less sensitive to inadequacy level variations?

GOAL - We aimed to help a more informed decision in the choice of an inadequacy level while using an inadequate TSR approach. The lower the sensitiveness of an inadequate approach to the inadequacy level, the better the approach is. It means that small variations in the inadequacy level slightly affect reduced TSs in terms of both their size and their fault-detection capability. That is, the practitioner can be aware that small variations in the inadequacy level cause at least a small loss in fault-detection capability and gain in the size of the reduced TSs.

The experimental objects considered in this study are 19 versions of four Java software systems: eight versions for *Ant*, three versions for *JTopas*, five versions for *JMeter*, and three versions for *XMLSecurity*. These experimental objects were the same as Zhang *et al.* [36] used in their study. In Table 3, we show basic information of the experimental objects: name, version, size (in terms of LOC), number of test cases (#Test Cases), number of statements the TS covers (#Covered Statements), number of faults the TS reveals (#Faults), and fault density ($\frac{\#Faults}{LOC}$).

All the artifacts were downloaded from SIR,[3] a public repository whose goal is to support rigorous experiments on regression testing [16]. The downloaded artifacts are documented with all the details (*e.g.,* faults) needed to compute the loss in fault-detection capability and thus to compare different TSR approaches.

[3] sir.unl.edu

Table 3: Information on the experimental objects.

System	Version	LOC	#Test Cases	#Covered Statements	#Faults	Fault Density
Ant	v1 (1.3)	23,796	133	3,701	3	1.26e-4
	v2 (1.4)	37,478	212	6,824	1	2.66e-5
	v3 (1.4.1)	37,554	217	6,950	2	5.32e-5
	v4 (1.5)	64,445	533	16,401	8	1.24e-4
	v5 (1.5.2)	66,085	539	16,746	7	1.05e-4
	v6 (1.5.3-1)	66,144	570	16,932	1	1.51e-5
	v7 (1.6beta)	88,414	842	24,386	10	1.13e-4
	v8 (1.6beta2)	88,449	845	24,383	2	2.26e-5
JTopas	v1 (0.4)	4,276	26	2,018	5	0.0011
	v2 (0.5.1)	4,520	28	2,122	4	8.8e-4
	v3 (0.6)	10,117	56	5,454	5	4.9e-4
JMeter	v1 (1.8)	33,620	51	2,388	4	1.18e-4
	v2 (1.8.1)	33,290	63	4,054	3	9e-5
	v3 (1.9.RC1)	37,474	78	4,933	8	2.13e-4
	v4 (1.9.RC2)	38,613	78	5,023	2	5.18e-4
	v5 (1.9)	40,989	91	4,942	2	4.87e-5
XMLSecurity	v1 (1.0.4)	21,601	94	4,730	5	2.13e-4
	v2 (1.0.5D2)	27,990	94	4,887	6	2.14e-4
	v3 (1.0.71)	19,731	84	4,668	3	1.52e-4
Mean		39,188.73	243.8	8,502.21	4.2	1.07e-4

4.2. Planning

We studied seven instances of the CB approach. Each instance differs on the basis of the chosen dissimilarity measure, that is: Euclidean distance (EUCL); cosine (dis)similarity (COS); Jaccard-based dissimilarity (JACC); Hamming distance (HAMM); Levenshtein distance (LEV); K-based dissimilarity (K); and string kernels-based dissimilarity (SK). To distinguish among the CB instances, we use the notation CB_X, where X denotes the chosen dissimilarity measure. For example, CB_{Cos} indicates the CB instance based on the cosine (dis)similarity.

We took into account in our experiment both the adequate and inadequate variants of the CB instances. By fixing the inadequacy level equal to 100%, the CB approach (and its instances) is adequate; for lower values of the inadequacy level, the CB approach is inadequate. To further distinguish between the adequate and inadequate variants of each CB instance, we act as for the traditional

TSR approaches, *i.e.,* we use the subscript A and I, respectively. For example, $\text{CB}_{\text{Cos A}}$ indicates the adequate variant of the CB instance based on the cosine (dis)similarity, while $\text{CB}_{\text{Cos I}}$ indicates the inadequate variant.

The CB instances were compared against the following well-known traditional TSR approaches in their adequate variant (see Section 2.1.1): HGS_A, GRD_A, DGR_A, 2OPT_A, GE_A, GRE_A, and ILP_A. We also studied the inadequate variants of the above-mentioned approaches: HGS_I, GRD_I, DGR_I, 2OPT_I, GE_I, and GRE_I. To obtain these inadequate variants, we applied the strategy by Shi *et al.* [8] highlighted in Section 2.1.2.

We chose these traditional approaches (both adequate and inadequate) because they are quite common in experiments on TSR (*e.g.,* [8, 22]). Besides, we considered the CB approach, which is based on that presented by Coviello *et al.* [9]—it is one of the most recent research contributions and then it might be considered one of the state-of-the-art approaches in the context of TSR.

4.3. Independent and Dependent Variables

In this experiment, we have one independent variable, which indicates the studied TSR approach. We named this (nominal) variable as **Method**. It assumes as values: $\text{CB}_{\text{Cos A}}$, $\text{CB}_{\text{Cos I}}$, GRD_A, GRD_I, and so on.

In prior empirical investigations, TSR approaches have been evaluated on the basis of the size of the reduced TSs and their capability to identify faults. Two metrics have been largely used as dependent variables (*e.g.,* [1, 2, 19, 37]) to estimate these constructs: Reduction in TS Size (RS) and Reduction in Fault-detection capability (RF). RS is computed as follows:

$$RS = \frac{|T| - |T'|}{|T|} \times 100 \tag{8}$$

where $|T|$ and $|T'|$ are the size of the original and reduced TSs, respectively. RS, that assume values between 0% and 100%, reflects the number of test cases within the original TS that are not present in the reduced one. From a practical perspective, a value equal to 0% means that the original TS has not been reduced at all. Thus, a high value for RS is desirable.

As for RF, it is computed via the following formula:

$$RF = \frac{F - F'}{F} \times 100 \tag{9}$$

where F is the number of faults detected by the original TS and F' is the number of faults detected by the reduced TS. RF assumes values between 0% and 100%. In particular, RF indicates the percentage of faults that a reduced TS does not detect with respect to the faults detected by the original TS. From a practical perspective, a RF value equal to 0% means that reduced TS has preserved the same capability of detecting faults as the original TS. Thus, a value equal to 0% is desirable.

To identify the inadequate TSR approach less sensitive to the inadequacy level, we used the Sensitiveness Measure (SM). To compute this measure, given an experimental object and TSR approach, we used the values of RS and RF at inadequacy levels ranging in between 5% and 100% with an increment of 5%. The use of a smaller increment in the inadequacy level would produce more accurate SM values, but it should not affect the comparison among the inadequate TSR approaches. We do not consider 0% because it would imply a reduction without test cases or at the most one test case for the CB approach. By plotting RS and RF values for each considered inadequacy level, we obtain a line (see for example Figure 4). It is worth noting that as RS increases, RF grows or remains the same (in the best case scenario). The SM value is computed as the area under this broken line. We compute this area by applying the trapezoidal rule. Therefore, SM can assume values in between 0 and 10,000. The smaller the SM value, the better it is. This means that if the RS value increases we have a small increment of the RF value while varying the inadequacy level. In other words, the lower the SM value, the lower the sensitiveness of the approach is. From a practical perspective, a less sensitive approach is better because the practitioner can lower the inadequacy level being confident that the reduction in fault-detection capability and in TS size slowly gets worse and improves, respectively.

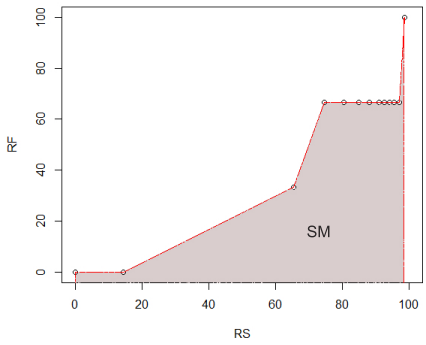

Figure 4: An example of SM value, *i.e.*,, the SM value computed for $2OPT_I$ on AntV1.

4.4. Hypotheses Formulation

To study RQ1, we formulated the following (parametrized) null hypothesis:

NH1$_Z$ - There is no statistically significant difference in the values of the dependent variable Z (*i.e.,* either RS or RF) computed by applying inadequate TSR approaches.

On the other hand, we used the following (parametrized) null hypothesis to study RQ2:

NH2$_Z$ - There is no statistically significant difference in the values of the dependent variable Z (*i.e.,* either RS or RF) computed by applying the best inadequate TSR approach/es (if any) and those computed by applying adequate approaches.

Finally, we used the following null hypothesis for RQ3:

NH3$_{SM}$ - There is no statistically significant difference in the values of SM when applying the inadequate TSR approaches.

4.5. Data Analysis

For each approach, we computed some descriptive statistics, which summarize the distributions of the RS, RF, and SM values. We also graphically

represented these distributions by means of boxplots. To test the null hypotheses, we planned to use Linear Mixed Model (LMM) analysis methods. If the assumptions behind these methods (*i.e.,* normality of residuals and mean of residuals approximately equals to 0) were not verified (neither by applying data transformations), we planned to use the Friedman test [38]. For each statistical test, we decided to accept (as it is customary) a probability of 5% of committing Type-I-error (*i.e.,* $\alpha = 0.05$).

4.6. Instrumentation

For most of the adequate approaches, we exploited RAISE. It is a Java tool that implements HGS_A, GRD_A, 2OPT_A, DGR_A. RAISE is available on the web[4] and has been used in previous empirical studies (*e.g.,* [22, 39]). To implement ILP_A, we used IBM's CPLEX Optimizer solver (version 12.7.1). As for the CB instances (in their adequate and inadequate variants), we used CUTER [33] (see Section 3.1). For the missing TSR approaches (*e.g.,* GE_A or GE_I), we implemented a Java prototype of supporting tool. Code coverage information was collected by means of JaCoCo.[5]

4.7. Threats to Validity

To understand strengths and limitations of our experiment, we discuss here threats that could affect the validity of the results.

- *Construct validity* threats concern the relationship between theory and observation. In our experiment, a possible threat to this this kind of validity concerns the use of single measure to assess constructs. In the literature, RS and RF are widely adopted to quantify the reductions in TS size and fault-detection capability, respectively. That is, they represent the standard for the assessment of TSR approaches (*e.g.,* [2, 19, 37]). On the other hand, SM has been newly defined in our research. We defined it

[4] code.google.com/archive/p/raise
[5] www.jacoco.org

because in the literature there are not measures conceived for this purpose. The use of SM, however, represents a possible threat to the validity of the results observed for all the studied approaches.

- *Internal validity* threats concern factors internal to the investigation. How the approaches were implemented might threaten the validity of the results. For example, the presence of bugs might affect the results in an unexpected way. To mitigate this threat, we tested any prototype we used.

- *Conclusion validity* threats concern the relationship between the dependent and independent variables. Statistical tests used to analyze the collected data might threaten conclusion validity. To mitigate the effect of this kind of threat, we planned to apply robust and sensitive statistical tests that are well-known and widely adopted in several research fields. The fact that inadequate approaches identify reductions that approximatively satisfy the fixed inadequacy level represents another threat to conclusion validity. The reliability of the used measures might also affect conclusion validity. To deal with this kind of threat, we opted for measures that did not require any subjective evaluation to be computed. As for SM, the use of the increment of 5% for inadequacy level represents another possible threat to conclusion validity. However, the use of such an increment would equally affect the computation of SM for each inadequate TSR approach. How the used dataset (*i.e.,* SIR) has been created represents another threat. It is worth mentioning that SIR was created by external researchers [16] and used in several studies on regression testing (*e.g.,* [40, 41, 42]).

- *External validity* threats concern the possibility of generalizing results. Although in our investigation we considered software systems previously used in other studies and these applications cover different application domains, we cannot guarantee that our findings can be generalized to the

universe of Java systems. Future work is needed to verify to what extent our findings hold for other experimental objects (*e.g.,* commercial ones).

- *Reliability validity* threats concern the capability of external researchers to replicate our study. We mitigate this kind of threat by making available on web the replication package that comprises, for example, the raw data.[6]

5. Results and Discussion

In this section, we present and discuss the results of our experiment according to the defined RQs; then we highlighting possible implications for these results and future directions for our research.

5.1. RQ1— Is there an inadequate TSR approach outperforming the others?

Descriptive Statistics and Exploratory Analysis. Figure 5 shows the boxplots of the RS values for the inadequate approaches at 95% of inadequacy level.

By looking at the boxplots in Figure 5, we can observe that the approaches are mostly comparable one another. Indeed, two of them (*i.e.,* HGS_I and DGR_I) seem to be worse than the others—the boxplots for HGS_I and DGR_I are lower. We can also observe that there are boxplots less skewed than others. This is the case of the CB instances with the only exception of $CB_{SK\ I}$.

Table 4 reports the descriptive statistics—median, mean, Standard Deviation (SD), and Confidence Interval (CI)—for inadequate approaches. As for RS, the descriptive statistics confirm that there is not a huge difference among inadequate approaches, with only two exceptions: HGS_I and DGR_I (their median and mean values for RS are the lowest).

Table 4 also reports the RF values and we can notice no major difference among the studied TSR approaches. That is, the loss in fault-detection capability is very similar, the average values for RF ranges between 0% and 0.88%,

[6] www2.unibas.it/gscanniello/IST/RawData.zip

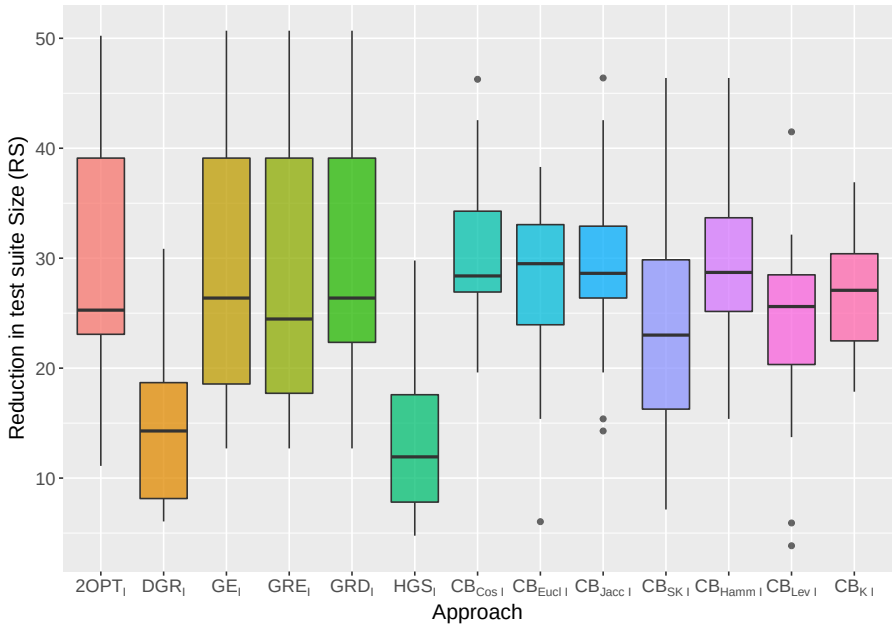

Figure 5: Boxplots of the RS values of each inadequate TSR approach with (inadequacy level = 95%).

with a median always equal to 0%. We can also observe that two approaches (*i.e.,* $2OPT_I$ and GE_I) have no loss in fault-detection capability—their mean and SD values for RF are equal to 0%.

We further investigated the differences among the reduced TSs the inadequate TSR approaches identified. To this end, we applied the strategy by Marchetto *et al.* [21] to quantify the diversity of the reduced TSs produced by two TSR approaches. The authors defined a construct—Diversity—, which represents the difference among a pair of TSs (S_1 and S_2.) with respect to the test cases they contain. To quantify such a construct, they defined the Div measure, which is computed as the number of test cases shared between S_1 and S_2 out of the number of test cases of the original TS. The higher $Div(S_1, S_2)$, the lower the diversity between S_1 and S_2 is. That is, a high value of $Div(S_1, S_2)$ means that S_1 and S_2 have a high number of test cases in common and share nearly the same test cases of the original TS. Div assumes values in between 0 and 1,

Table 4: Some descriptive statistics for RS and RF for each inadequate TSR approach (inadequacy level = 95%).

	RS (%)				RF (%)			
Approach	median	mean	SD	CI	median	mean	SD	CI
$2OPT_I$	25.27	29.66	13.25	[23.27;36.04]	0	0	0	[0;0]
DGR_I	14.29	15.48	8.18	[11.54;19.42]	0	0.75	3.28	[-0.83;2.33]
GE_I	26.37	29.17	13.82	[22.51;35.83]	0	0	0	[0;0]
GRE_I	24.47	28.98	13.94	[22.26;35.7]	0	0.87	2.67	[-0.42;2.15]
GRD_I	26.37	29.57	13.52	[23.06;36.09]	0	0.88	2.71	[-0.43;2.19]
HGS_I	11.93	14.26	7.78	[10.51;18.01]	0	0.75	3.28	[-0.83;2.33]
$CB_{\text{Cos I}}$	28.39	30.6	7	[27.22;33.97]	0	0.66	2.87	[-0.72;2.04]
$CB_{\text{Eucl I}}$	29.5	27.99	8.12	[24.08;31.91]	0	0.75	3.28	[-0.83;2.33]
$CB_{\text{Jacc I}}$	28.62	29.26	8.08	[25.36;33.16]	0	0.66	2.87	[-0.72;2.04]
$CB_{\text{SK I}}$	23.01	22.7	10.44	[17.66;27.73]	0	0.75	3.28	[-0.83;2.33]
$CB_{\text{Hamm I}}$	28.71	29.34	7.48	[25.74;32.94]	0	0.75	3.28	[-0.83;2.33
$CB_{\text{Lev I}}$	25.6	23.73	9.05	[19.37;28.09]	0	0.75	3.28	[-0.83;2.33]
$CB_{\text{K I}}$	27.08	27.12	5.93	[24.27;29.98]	0	0.75	3.28	[-0.83;2.33]
Mean	24.5	25.7	9.73	-	0	0.64	2.6	-

where 0 indicates that two reduce TSs are completely different.

In Table 5, we report some descriptive statistics (*i.e.,* median, mean, and SD) for the Div measure computed for each pair of inadequate TSR approach. The mean values of $Div(2OPT_I, CB_{\text{Hamm I}})$, $Div(DGR_I, HGS_I)$, $Div(CB_{\text{Eucl I}}, CB_{\text{Hamm I}})$, $Div(CB_{\text{Eucl I}}, CB_{\text{K I}})$, and $Div(CB_{\text{Hamm I}}, CB_{\text{K I}})$ are close to 1. This indicates that there is a high number of test cases in common between the reduced TSs that these pairs of approaches identified. The SD values are low, thus suggesting that the reduced t are generally very similar one another. In general, the results of this further analysis confirm that the approaches that produced worse results in terms of RS (*i.e.,* DGR_I and HGS_I) identified similar reduced TSs. This outcome does not hold for the better approaches, namely those that reduced more the TSs. For example, the mean values of the $Div(2OPT_I, CB_{\text{Cos I}})$, $Div(GE_I, CB_{\text{Cos I}})$, $Div(GRE_I, CB_{\text{Cos I}})$, $Div(GRD_I, CB_{\text{Cos I}})$ are low despite these approaches allowed obtaining an appreciable reduction of the original TSs with a negligible effect on fault-detection capability. Finally, we can observe that the reduced TSs of the traditional approaches are different one another, while the reduced TSs of the CB instances

are not so different one another. Indeed, $\text{CB}_{\text{Cos I}}$ behaves slightly different (in terms of the diversity of the reduced TSs) from the other CB instances as the descriptive statistics suggest; the mean values of Div range in between 0.46 and 0.48, while the SD values are higher than the others. We can then postulate that the effect of the similarity measures to compare test cases slight affect the reduced TSs, while the clustering algorithm could make a difference. We devise this latter point as a possible future direction for our research.

Hypotheses Testing. To test the effect of Method on RS and RF (*i.e.,* $NH1_{RS}$ and $NH1_{RF}$), we ran the Friedman test since the assumptions to apply LMM analysis methods were not verified. In particular, the residuals were not normally distributed even after applying data transformations. The Friedman test allowed us to reject $NH1_{RS}$ (the returned p-value was equal to 4.92e-10). In other words, there is a statistically significant difference in RS. However, we could not reject $NH1_{RF}$ (p-value equal to 0.9265). This means that, from a statistical point of view, the studied inadequate approaches are not significantly different with respect to RF.

Since $NH1_{RS}$ was rejected, we performed a post-hoc analysis, namely pairwise comparisons among the inadequate TSR approaches. To this end, we applied a two-sided Wilcoxon rank-sum test (also known as Mann-Whitney U test) [34]. Through this test, we verified the following null hypothesis: *there is no statistically significant difference in the RS values computed by applying* $Method1$ *and* $Method2$*, where* $Method1$ *and* $Method2$ *are two inadequate TSR approaches.* In case of a statistically significant difference, we quantified the magnitude of such a difference using the Cliff's δ effect size.[7] This kind of effect size is used in case data are not normally distributed or the normality assumption is discarded [44].

The results of the post-hoc analysis for RQ1 are summarized in Table 6; p-values are reported in bold only when significant. We can observe that all the

[7]The Cliff's δ effect size is considered: Negligible (N) if $|\delta| < 0.147$; Small (S) if $0.147 \leq |\delta| < 0.33$; Medium (M) if $0.33 \leq |\delta| < 0.474$; or Large (L) otherwise [43].

Table 5: Some descriptive statistics for the Div measure computed for each pair of inadequate TSR approach (inadequacy level = 95%). This matrix is symmetric and, therefore, we reported only the Div values above the main diagonal.

	Approach	$2OPT_I$	DGR_I	GE_I	GRE_I	GRD_I	HGS_I	$CB_{COS\,I}$	$CB_{EUCL\,I}$	$CB_{JACC\,I}$	$CB_{SK\,I}$	$CB_{HAMM\,I}$	$CB_{LEV\,I}$	$CB_{K\,I}$
median			0.45	0.56	0.56	0.53	0.45	0.39	0.45	0.45	0.46	0.71	0.43	0.45
mean	$2OPT_I$		0.5	0.59	0.59	0.52	0.51	0.36	0.49	0.48	0.51	0.71	0.49	0.49
SD			0.09	0.09	0.09	0.16	0.1	0.14	0.09	0.08	0.1	0.07	0.1	0.09
median				0.56	0.45	0.45	0.82	0.45	0.61	0.56	0.58	0.6	0.64	0.64
mean	DGR_I			0.49	0.49	0.49	0.79	0.46	0.59	0.57	0.62	0.57	0.63	0.59
SD				0.09	0.09	0.09	0.11	0.16	0.11	0.09	0.12	0.09	0.12	0.09
median					0.56	0.56	0.45	0.38	0.45	0.44	0.46	0.44	0.43	0.45
mean	GE_I				0.59	0.59	0.5	0.36	0.48	0.47	0.5	0.48	0.49	0.49
SD					0.09	0.09	0.09	0.14	0.09	0.08	0.1	0.09	0.1	0.09
median						0.56	0.45	0.38	0.47	0.44	0.46	0.44	0.45	0.46
mean	GRE_I					0.59	0.5	0.36	0.48	0.47	0.51	0.48	0.49	0.49
SD						0.09	0.09	0.14	0.08	0.08	0.1	0.09	0.1	0.09
median							0.45	0.38	0.45	0.44	0.46	0.44	0.43	0.45
mean	GRD_I						0.5	0.36	0.48	0.47	0.5	0.48	0.49	0.49
SD							0.09	0.14	0.09	0.08	0.1	0.09	0.1	0.09
median								0.46	0.61	0.56	0.59	0.6	0.64	0.64
mean	HGS_I							0.46	0.6	0.58	0.63	0.59	0.64	0.6
SD								0.16	0.11	0.09	0.13	0.09	0.12	0.09
median									0.58	0.6	0.55	0.56	0.57	0.56
mean	$CB_{COS\,I}$								0.46	0.48	0.46	0.46	0.47	0.46
SD									0.21	0.23	0.2	0.2	0.2	0.2
median										0.64	0.59	0.69	0.68	0.71
mean	$CB_{EUCL\,I}$									0.65	0.61	0.7	0.69	0.71
SD										0.06	0.07	0.07	0.08	0.06
median											0.59	0.64	0.65	0.64
mean	$CB_{JACC\,I}$										0.62	0.64	0.65	0.65
SD											0.1	0.07	0.08	0.07
median												0.59	0.61	0.6
mean	$CB_{SK\,I}$											0.6	0.64	0.62
SD												0.08	0.08	0.08
median													0.67	0.71
mean	$CB_{HAMM\,I}$												0.68	0.71
SD													0.07	0.07
median														0.69
mean	$CB_{LEV\,I}$													0.69
SD														0.07
median														
mean	$CB_{K\,I}$													
SD														

Table 6: Results from the pairwise comparisons between the inadequate TSR approach for RS. We report in bold p-values less than 0.003 (*i.e.*, α value normalized by applying the Bonferroni correction). The reported matrix is symmetrical except for the sign of the Cliff's δ effect size.

	$2OPT_I$	DGR_I	GE_I	GRE_I	GRD_I	HGS_I	$CB_{Cos\,I}$	$CB_{Eucl\,I}$	$CB_{Jacc\,I}$	$CB_{SK\,I}$	$CB_{Hamm\,I}$	$CB_{Lev\,I}$	$CB_{K\,I}$
$2OPT_I$	-	**0.0013** L (0.6122)	0.9185	0.9883	0.8038	**0.0004** L (0.6787)	0.2734	0.5206	0.4389	0.1609	0.5206	0.397	0.8839
DGR_I	**0.0013** L (-0.6122)	-	**0.0024** L (-0.5789)	**0.0028** L (-0.5706)	**0.0016** L (-0.6012)	0.5492	**2.02E-02** L (-0.8116)	**0.0002** L (-0.7063)	**9.13E-02** L (-0.7452)	0.0265	**6.33E-02** L (-0.7618)	0.0086	**0.0002** L (-0.7119)
GE_I	0.9185	**0.0024** L (0.5789)	-	0.9417	0.9767	**0.0005** L (0.662)	0.2488	0.5206	0.3576	0.2312	0.4478	0.5018	0.7815
GRE_I	0.9883	**0.0028** L (0.5706)	0.9417	-	0.9185	**0.0006** L (0.651)	0.2255	0.5018	0.3278	0.2547	0.4304	0.5592	0.7591
GRD_I	0.8038	**0.0016** L (0.6012)	0.9767	0.9185	-	**0.0003** L (0.6842)	0.2735	0.5791	0.3889	0.1699	0.5206	0.4136	0.9186
HGS_I	**0.0004** L (-0.6787)	0.5492	**0.0005** L (-0.662)	**0.0006** L (-0.651)	**0.0003** L (-0.6842)	-	**5.23E-03** L (-0.867)	**8.08E-02** L (-0.7506)	**2.61E-02** L (-0.8005)	0.0093	**2.29E-02** L (-0.8061)	**0.0029** L (-0.5679)	**3.83E-02** L (-0.7839)
$CB_{Cos\,I}$	0.2734	**2.02E-02** L (0.8116)	0.2488	0.2255	0.2735	**5.23E-03** L (0.867)	-	0.5992	0.9767	0.0245	0.726	0.0147	0.161
$CB_{Eucl\,I}$	0.5206	**0.0002** L (0.7063)	0.5206	0.5018	0.5791	**8.08E-02** L (0.7506)	0.5992	-	0.9651	0.0539	0.8954	0.0902	0.4653
$CB_{Jacc\,I}$	0.4389	**9.13E-02** L (0.7452)	0.3576	0.3278	0.3889	**2.61E-02** L (0.8005)	0.9767	0.9651	-	0.042	1	0.0439	0.3068
$CB_{SK\,I}$	0.1609	0.0265	0.2312	0.2547	0.1699	0.0093	0.0245	0.0539	0.042	-	0.0318	0.748	0.1289
$CB_{Hamm\,I}$	0.5206	**6.33E-02** L (0.7618)	0.4478	0.4304	0.5206	**2.29E-02** L (0.8061)	0.726	0.8954	1	0.0318	-	0.0616	0.3653
$CB_{Lev\,I}$	0.397	0.0086	0.5018	0.5592	0.4136	**0.0029** L (0.5679)	0.0147	0.0902	0.0439	0.748	0.0616	-	0.2931
$CB_{K\,I}$	0.8839	**0.0002** L (0.7119)	0.7815	0.7591	0.9186	**3.83E-02** L (0.7839)	0.161	0.4653	0.3068	0.1289	0.3653	0.2931	-

approaches (except $CB_{SK\,I}$ and $CB_{Lev\,I}$) reduce more than DGR_I and HGS_I. The effect size is large (label L).

Summary. Given a fixed inadequacy level (*i.e.*, 95%), we are not able to identify a single inadequate TSR approach performs the best in terms of high reduction in TS size and low loss in fault-detection capability. However, there are several approaches that achieved comparable results in terms of RS and RF. DGR_I and HGS_I seem to perform worse than the other inadequate approaches studied in our experiment because they reduce significantly less the size of reduced TSs with a very similar loss in fault-detection capability. Results also suggest that the CB instances are less affected by the experimental objects (*i.e.*, the boxes in Figure 5 are less skewed) and the median values for RS are generally higher than those obtained by applying the other inadequate TSR approaches

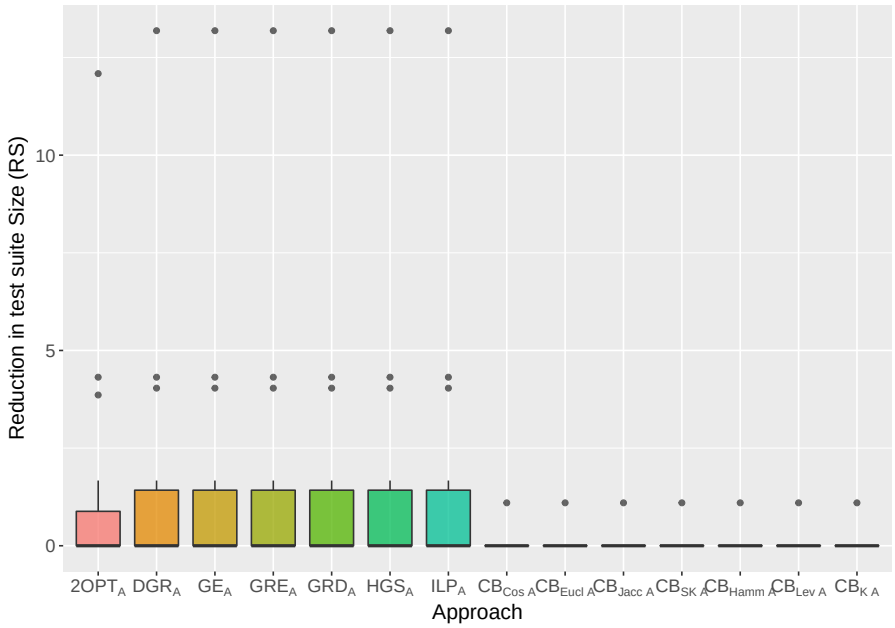

Figure 6: Boxplots of the RS values of each adequate TSR approach.

(it is not completely true for $\text{CB}_{\text{SK I}}$ and $\text{CB}_{\text{Lev I}}$). To conclude, even if 2OPT_{I} and GE_{I} tend to perform slightly better than other approaches in terms of RF, we did not observe a single clear winner among the studied inadequate TSR approaches, rather there is a set of approaches that behave similarly and that can be successfully applied: 2OPT_{I}, GE_{I}, GRE_{I}, GRD_{I}, $\text{CB}_{\text{Cos I}}$, $\text{CB}_{\text{Eucl I}}$, $\text{CB}_{\text{Jacc I}}$, $\text{CB}_{\text{Hamm I}}$, and $\text{CB}_{\text{K I}}$.

5.2. RQ2—Do the best inadequate TSR approaches outperform adequate ones?

Descriptive Statistics and Exploratory Analysis. In Figure 6, we show the RS boxplots for the studied adequate TSR approaches. The boxplots for the well-known traditional adequate approaches seem to suggest that they reduce more than the instances of the CB approach. Indeed, the CB instances did not reduce the TSs for all the experimental objects with the only exception of JMeter v5 (the outliers in Figure 6) on which a reduction of 1.1% was obtained. As for the traditional approaches, the outliers are JMeter v5, Ant v4 and Ant

v6. On JMeter v5, the traditional approaches allowed reducing the TS size of 13.19% (except for $2OPT_A$ that reduced the original TS of 12.09%). These approaches reduced the original TS of Ant v4 of 4.32%, while they reduced the original TS of Ant v6 of 4.04% (with the only exception of $2OPT_A$ that allowed reducing the TS of Ant v6 of 3.86%). We can speculate that the outliers are due to the redundancy, in terms of code coverage, of some test cases in the TSs.

These findings from the boxplots were corroborated with stronger evidence by the descriptive statistics for RS shown in Table 7. This table also reports the descriptive statistics for RF. Median, mean, SD, and CI values for RF are all equal to 0% so indicating that the reduction in TS sizes (if any) does not affect fault-detection capability.

The best inadequate approaches were not able to achieve the same results on RF (see Table 4). Only $2OPT_I$ and GE_I behaved like traditional adequate approaches on RF (the values for the median, mean, and SD are equal to 0). Nevertheless, the mean values on RF, for all the inadequate approaches, are always less than one. On the other hand, all the inadequate approaches outperform adequate ones in terms of RS.

Hypotheses Testing. We applied the Friedman test to verify the effect of Method (*i.e.,* adequate TSR approaches and the best inadequate ones) in RS and RF, *i.e.,* to test $NH2_{RS}$ and $NH2_{RF}$, respectively. The result of the Friedman test showed that there was a statistically significant difference on RS (p-value was 2.20e-16). That is, we could reject $NH2_{RS}$. On the other hand, we could not reject $NH2_{RF}$ (p-value was 0.4424). Note that we did not apply LMM analysis methods because the assumptions were not verified (*i.e.,* residuals were not normally distributed).

Since we rejected $NH2_{RS}$, we further investigated on RS. In particular, we performed pairwise comparisons (post-hoc analysis) among TSR approaches. Similarly to RQ1, we applied a two-sided Wilcoxon rank-sum test. The obtained results are summarized in Table 8 together with the Cliff's δ effect size values. We can note that the inadequate approaches reduce significantly more than adequate approaches and the effect size is always large (L).

Table 7: Some descriptive statistics for RS and RF for each adequate TSR approach.

	RS (%)				RF (%)			
Approach	median	mean	SD	CI	median	mean	SD	CI
$2OPT_A$	0	1.25	2.93	[-0.17;2.66]	0	0	0	[0;0]
DGR_A	0	1.42	3.15	[-0.1;2.94]	0	0	0	[0;0]
GE_A	0	1.4	3.15	[-0.12;2.91]	0	0	0	[0;0]
GRE_A	0	1.4	3.15	[-0.12;2.91]	0	0	0	[0;0]
GRD_A	0	1.4	3.15	[-0.12;2.91]	0	0	0	[0;0]
HGS_A	0	1.42	3.15	[-0.1;2.94]	0	0	0	[0;0]
ILP_A	0	1.42	3.15	[-0.1;2.94]	0	0	0	[0;0]
$CB_{Cos\ A}$	0	0.06	0.25	[-0.06;0.18]	0	0	0	[0;0]
$CB_{Eucl\ A}$	0	0.06	0.25	[-0.06;0.18]	0	0	0	[0;0]
$CB_{Jacc\ A}$	0	0.06	0.25	[-0.06;0.18]	0	0	0	[0;0]
$CB_{SK\ A}$	0	0.06	0.25	[-0.06;0.18]	0	0	0	[0;0]
$CB_{Hamm\ A}$	0	0.06	0.25	[-0.06;0.18]	0	0	0	[0;0]
$CB_{Lev\ A}$	0	0.06	0.25	[-0.06;0.18]	0	0	0	[0;0]
$CB_{K\ A}$	0	0.06	0.25	[-0.06;0.18]	0	0	0	[0;0]
Mean	0	0.72	1.68	-	0	0	0	-

Summary. The obtained results allow us to positively answer RQ2. In particular, we can conclude that the best inadequate TSR approaches outperform adequate ones in terms of reduction in TS size with a negligible effect on fault-detection capability. Adequate approaches seem to be viable competitors of inadequate ones only in case of the test cases in a given TS are redundant in terms of code coverage. We also noted that when the CB approach is used in its adequate variant, it reduces less than the other adequate approaches. It is worth mentioning that when the CB approach is used as inadequate its instances perform better than traditional adequate ones (*i.e.,* they reduce the TSs more with a small effect on the fault-detection capability).

5.3. RQ3—Which is the less sensitive inadequate TSR approach?

Descriptive Statistics and Exploratory Analysis. Figure 7 reports the boxplots for SM. CB instances give rise to less skewed distributions with lower median values. This is an evidence that CB instances are less sensitive to the inadequacy level. Among the CB instances, $CB_{Cos\ I}$ and $CB_{Jacc\ I}$ exhibit the smallest interquartile range, while their median values are slightly worse than

Table 8: Results from the pairwise comparisons between the best TSR inadequate approaches and each adequate approach on RS. All the p-values are less than 0.002 (*i.e.*, α value normalized by applying the Bonferroni correction).

	$2OPT_A$	DGR_A	GE_A	GRE_A	GRD_A	HGS_A	ILP_A	$CB_{Cos\ A}$	$CB_{Eucl\ A}$	$CB_{Jacc\ A}$	$CB_{SK\ A}$	$CB_{Hamm\ A}$	$CB_{Lev\ A}$	$CB_{K\ A}$
$2OPT_I$	9.57E-08 L (0.9944)	1.09E-07 L (0.9944)	1.09E-07 L (0.9944)	1.09E-07 L (0.9944)	1.09E-07 L (0.9944)	1.09E-07 L (0.9944)	1.09E-07 L (0.9944)	2.70E-08 L (1)	2.70E-08 L (1)	2.70E-08 L (1)	2.70E-08 L (1)	2.70E-08 L (1)	2.70E-08 L (1)	2.70E-08 L (1)
GE_I	8.17E-08 L (1)	1.10E-07 L (0.9944)	1.10E-07 L (0.9944)	1.10E-07 L (0.9944)	1.10E-07 L (0.9944)	1.10E-07 L (0.9944)	1.10E-07 L (0.9944)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)
GRE_I	8.13E-08 L (1)	1.09E-07 L (0.9944)	1.09E-07 L (0.9944)	1.09E-07 L (0.9944)	1.09E-07 L (0.9944)	1.09E-07 L (0.9944)	1.09E-07 L (0.9945)	2.71E-08 L (1)	2.71E-08 L (1)	2.71E-08 L (1)	2.71E-08 L (1)	2.71E-08 L (1)	2.71E-08 L (1)	2.71E-08 L (1)
GRD_I	8.17E-08 L (1)	1.10E-07 L (0.9944)	1.10E-07 L (0.9944)	1.10E-07 L (0.9944)	1.10E-07 L (0.9944)	1.10E-07 L (0.9944)	1.10E-07 L (0.9944)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)
$CB_{Cos\ I}$	8.17E-08 L (1)	9.32E-08 L (1)	9.32E-08 L (1)	9.32E-08 L (1)	9.32E-08 L (1)	9.32E-08 L (1)	9.32E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)
$CB_{Eucl\ I}$	9.64E-08 L (0.9944)	1.10E-07 L (0.9944)	1.10E-07 L (0.9944)	1.10E-07 L (0.9944)	1.10E-07 L (0.9944)	1.10E-07 L (0.9944)	1.10E-07 L (0.9944)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)
$CB_{Jacc\ I}$	8.17E-08 L (1)	9.32E-08 L (1)	9.32E-08 L (1)	9.32E-08 L (1)	9.32E-08 L (1)	9.32E-08 L (1)	9.32E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)
$CB_{Hamm\ I}$	8.17E-08 L (1)	9.32E-08 L (1)	9.32E-08 L (1)	9.32E-08 L (1)	9.32E-08 L (1)	9.32E-08 L (1)	9.32E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)
$CB_{K\ I}$	8.15E-08 L (1)	9.29E-08 L (1)	9.29E-08 L (1)	9.29E-08 L (1)	9.29E-08 L (1)	9.29E-08 L (1)	9.29E-08 L (1)	2.71E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)	2.72E-08 L (1)

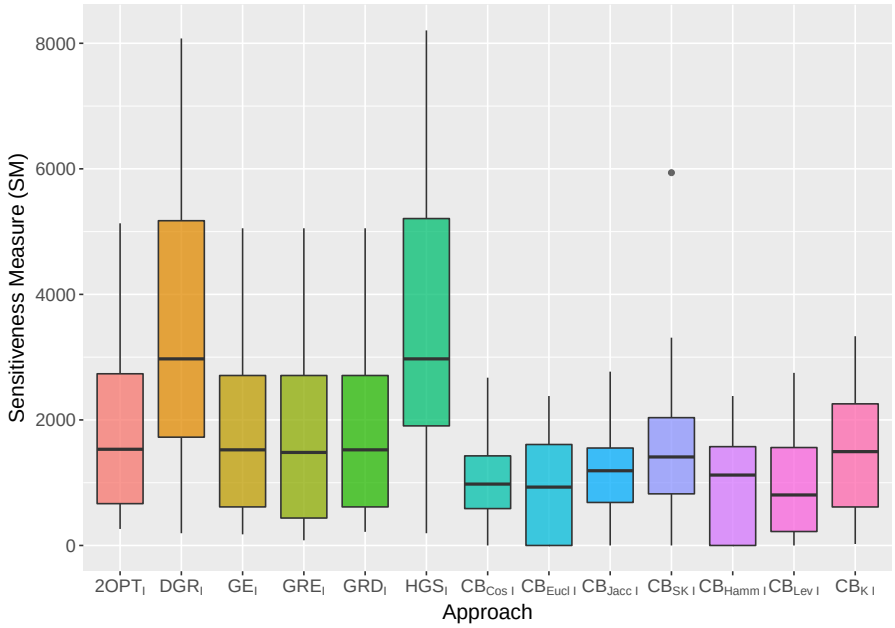

Figure 7: Boxplots of the SM values of each inadequate TSR approach.

Table 9: Some descriptive statistics for SM for each inadequate TSR approach.

	SM			
Approach	median	mean	SD	CI
$2OPT_I$	1,532.54	1,881.68	1,396.78	[1208.45;2554.9]
DGR_I	2,972.76	3,402.89	2,276.01	[2305.89;4499.89]
GE_I	1,523.67	1,892.66	1,452.36	[1192.64;2592.67]
GRE_I	1,482.25	1,832.32	1,501.46	[1108.64;2556]
GRD_I	1,523.67	1,940.81	1,551.37	[1193.07;2688.55]
HGS_I	2,972.76	3,473.5	2,269.73	[2379.53;4567.48]
$CB_{Cos\,I}$	977.96	1,068.76	798.35	[683.96;1453.55]
$CB_{Eucl\,I}$	929.93	967.31	861.68	[551.99;1382.63]
$CB_{Jacc\,I}$	1,160.72	1,188.65	849.22	[784;1602.7]
$CB_{SK\,I}$	1,410.26	1,600.57	1,400.23	[925.68;2275.45]
$CB_{Hamm\,I}$	1,121.73	962.16	851.52	[547.73;1367.19]
$CB_{Lev\,I}$	804.63	1,000.26	895.8	[568.5;1432.02]
$CB_{K\,I}$	1,494.83	1,460.93	1,015.08	[971.67;1950.18]
Mean	1,531.36	1,744.04	1,316.89	-

the median values of $CB_{Eucl\,I}$ and $CB_{Lev\,I}$. To complete data exploration, we report some descriptive statistics in Table 9 (mean, median, SD, and CI) for SM. The reported statistics confirm the visual inspection.

Hypotheses Testing. We applied the Friedman test to study $NH3_{SM}$. We exploited this non-parametric test because the assumptions to apply linear mixed model analysis methods were not verified (*i.e.,* residuals were not normally distributed also by applying data transformations). The results of the Friedman test allowed us to reject $NH3_{SM}$ since the p-value was equal to 5.04e-11. That is, the test revealed the presence of a statistically significant difference for SM among the studied inadequate TSR approaches. This outcome justifies a post-hoc analysis, *i.e.,* pairwise comparisons among the distributions for SM attained by applying these approaches to the experimental objects. The obtained results are reported in Table 10. In particular, the results of the two-sided Wilcoxon rank-sum test suggest that there is not a statistically significant difference among the CB instances. The results of the post-hoc analysis also indicate a statistically significant difference between the CB instances and DGR_I and HGS_I. The effect size is large (*i.e.,* L) in all the cases and sign is negative so suggesting that such a difference is in favor of the CB approach. It is also possible to note that

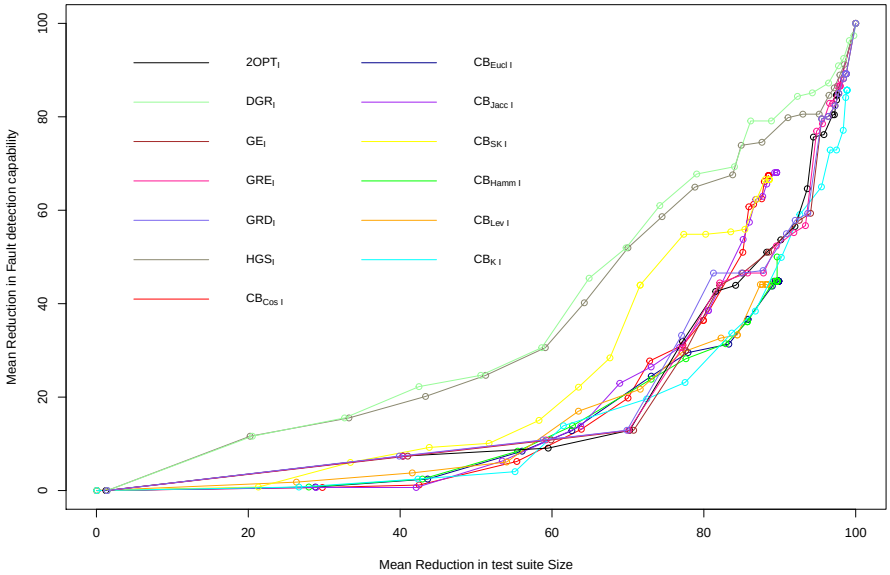

Figure 8: Linear plots for each inadequate TSR approach built with the mean values for RS and RF for inadequacy levels ranging in between 5% and 100%. Note that the instances of the CB approach select, by definition, at least one test case and this is why they cannot reach RS equal to 100%.

HGS_I is also significantly worse than the inadequate variants of the traditional adequate TSR approaches. The effect size in these cases is medium (*i.e.*, M).

We further investigate the sensitiveness of the studied inadequate TSR approaches by using the line-plot in Figure 8. Each line represents the mean values for RS and RF of each approach on all the experimental objects for inadequacy levels ranging in between 5% and 100%. The area underlying each line—computed by applying the trapezoidal rule—provides an indication on the average trend of a given approach with respect to its sensitiveness. The lower the area, the better the approach is. This means that the approach is not very affected by the inadequacy level. From a practical perspective, the tester can be aware that an approach less sensitive to inadequacy levels slightly affects the size of the reduced TSs with a small negative effect (if any) on fault-detection

Table 10: Results from the pairwise comparisons between inadequate TSR approaches on SM. We report in bold p-values less than 0.0385 (*i.e.*, α value normalized by applying the Bonferroni correction).

	$2OPT_I$	DGR_I	GE_I	GRE_I	GRD_I	HGS_I	$CB_{Cos\,I}$	$CB_{Eucl\,I}$	$CB_{Jacc\,I}$	$CB_{SK\,I}$	$CB_{Hamm\,I}$	$CB_{Lev\,I}$	$CB_{K\,I}$
$2OPT_I$	-	0.041	0.8724	0.6935	0.8267	0.0286	0.0798	**0.0327** M (0.4072)	0.1364	0.4654	**0.0352** M (0.4017)	**0.0381** M (0.3961)	0.4655
DGR_I	0.041	-	**0.0382** M (0.3961)	**0.0331** M (0.4072)	0.041	0.7927	**0.0006** L (0.651)	**0.0003** L (0.6933)	**0.001** L (0.6288)	**0.0086** L (0.5014)	**0.0002** L (0.7064)	**0.0004** L (0.6731)	**0.0086** L (0.5014)
GE_I	0.8724	**0.0382** M (0.3961)	-	0.8609	1	**0.0286** M (-0.4183)	0.0903	**0.0352** M (0.4017)	0.1525	0.4478	**0.0378** M (0.3961)	0.0503	0.4478
GRE_I	0.6935	0.0331 M (0.4072)	0.8609	-	0.8724	**0.0246** M (-0.4294)	0.161	0.05	0.2312	0.5398	0.05	0.0797	0.6197
GRD_I	0.8267	0.041	1	0.8724	-	**0.0331** M (-0.4072)	0.0903	**0.0352** M (0.4017)	0.1525	0.4478	**0.0378** M (0.3961)	0.0503	0.4655
HGS_I	0.0286	0.7927	**0.0286** M (0.4183)	**0.0246** M (0.4294)	**0.0331** M (0.4072)	-	**0.0004** L (0.6787)	**0.0002** L (0.7175)	**0.0008** L (0.6399)	**0.0066** L (0.518)	**0.0002** L (0.7119)	**0.0003** L (0.6898)	**0.0055** L (0.5291)
$CB_{Cos\,I}$	0.0798	**0.0006** L (-0.651)	0.0903	0.161	0.0903	**0.0004** L (-0.6787)	-	0.7024	0.6087	0.1607	0.7801	0.8148	0.2548
$CB_{Eucl\,I}$	**0.0327** M (-0.4072)	**0.0003** L (-0.6933)	**0.0352** M (-0.4017)	0.05	**0.0352** M (-0.4017)	**0.0002** L (-0.7175)	0.7024	-	0.4898	0.1347	0.9882	0.8829	0.0955
$CB_{Jacc\,I}$	0.1364	**0.001** L (-0.6288)	0.1525	0.2312	0.1525	**0.0008** L (-0.6399)	0.6087	0.4898	-	0.3497	0.4625	0.5581	0.4654
$CB_{SK\,I}$	0.4654	**0.0086** L (-0.5014)	0.4478	0.5398	0.4478	**0.0066** L (-0.518)	0.1607	0.1347	0.3497	-	0.1272	0.1832	0.9534
$CB_{Hamm\,I}$	**0.0352** M (-0.4017)	**0.0002** L (-0.7064)	**0.0378** M (-0.3961)	0.05	**0.0378** M (-0.3961)	**0.0002** L (-0.7119)	0.7801	0.9882	0.4625	0.1272	-	0.8829	0.1077
$CB_{Lev\,I}$	**0.0381** M (-0.3961)	**0.0004** L (-0.6731)	0.0503	0.0797	0.0503	**0.0003** L (-0.6898)	0.8148	0.8829	0.5581	0.1832	0.8829	-	0.1216
$CB_{K\,I}$	0.4655	**0.0086** L (-0.5014)	0.4478	0.6197	0.4655	**0.0055** L (-0.5291)	0.2548	0.0955	0.4654	0.9534	0.1077	0.1216	-

capability. In Table 11, we summarize the results of the area under the line for a given approach. Each entry of this table indicates the area under the line for an approach fixed an inadequacy level range. Analyzing the entries for each approach, we have indications on the inadequacy level that does not produce increment in the area under the line. This means that by choosing lower inadequacy levels, a greater reduction of the TSs could be obtained that do not negatively affect fault-detection capability. The plots in Figure 8 and the results shown in Table 11 suggest that $\mathrm{CB}_{\mathrm{HAMM\ I}}$ and $\mathrm{CB}_{\mathrm{LEV\ I}}$ are the approaches that allow obtaining a reduction of the TSs of about 80% when choosing an inadequacy level ranging from 5% to 40%. In this case, these approaches had fault-detection capability less than 40%. That is, they reduce more than other approaches and compared with them they lose a lower number of faults. As for $\mathrm{CB}_{\mathrm{EUCL\ I}}$, similar considerations can be done. With regards to this further analysis, we can observe that the worse approaches are: $\mathrm{DGR}_{\mathrm{I}}$, $\mathrm{HGS}_{\mathrm{I}}$, and $\mathrm{GRD}_{\mathrm{I}}$.

Summary. The results suggest that the CB approach is less sensitive to inadequacy level variations with respect to the other approaches studied in our experiment. Indeed, the best CB instances in terms of sensitiveness seem to be $\mathrm{CB}_{\mathrm{EUCL\ I}}$, $\mathrm{CB}_{\mathrm{HAMM\ I}}$, and $\mathrm{CB}_{\mathrm{LEV\ I}}$. Among the inadequate variants of the traditional adequate TSR approaches, we observed that $\mathrm{HGS}_{\mathrm{I}}$ and $\mathrm{DGR}_{\mathrm{I}}$ perform worse than the others.

5.4. Implications and Future Extensions

There are several foreseeable future directions for our research. Traditionally, adequate approaches have been preferred due to the intrinsic ability to preserve the test requirements of the original TSs[8] (*i.e.,* statement coverage in our experiment). However, the ever-increasing demand for fast development and deployment cycles challenges the adequate TSR superiority from the prac-

[8]Notice that, satisfying all the test requirements does not mean that the fault-detection capability of the original TS is preserved.

Table 11: Area values under the line obtained by computed for each TSR approach the mean values for RS and RF according to different inadequacy level ranges.

	Approach												
inadequacy level range	$2OPT_I$	DGR_I	GE_I	GRE_I	GRD_I	HGS_I	$CB_{Cos\ I}$	$CB_{Eucl\ I}$	$CB_{Jacc\ I}$	$CB_{SK\ I}$	$CB_{Hamm\ I}$	$CB_{Lev\ I}$	$CB_{K\ I}$
100-95	144.1	111.19	145.92	143.58	141.81	109.43	9.77	10.82	9.49	7.99	10.49	23.71	10
100-90	301.52	276.09	317.72	315.02	314.18	286.24	21.5	34.58	18.17	49.01	34.55	66.31	35.2
100-85	419.44	461.32	446.78	444.39	445.28	466.3	69.2	102.36	57.74	128.18	102.82	127.74	76.78
100-80	575.29	652.12	593.9	598.26	611.05	643.81	151.39	171.94	161.66	204.32	183.34	236.94	133.83
100-75	738.1	873.87	758.99	782.61	778.79	860.64	252.54	366.87	255.99	287	378.57	394.41	317.78
100-70	851.97	1,111.25	895.11	949.97	951.4	1,042.49	320.31	497.67	357.44	383.66	497.7	535.41	425.38
100-65	1,046.47	1,348.27	1,062.79	1,046.86	1,083.71	1,307.14	449.4	661.12	358.01	488.2	653.67	695.26	600.42
100-60	1,143.62	1,599.25	1,062.79	1,130.7	1,240.61	1,555.59	540.34	746.22	596.3	632.13	750.49	766	712.12
100-55	1,246.1	1,911.97	1,281.7	1,254.7	1,305.7	1,822.92	540.34	747.24	596.3	632.13	750.49	766	863.16
100-50	1,344.52	2,252.84	1,369.11	1,340.38	1,402.66	2,152.61	767.34	747.24	807.22	916.12	872.92	883.39	995.91
100-45	1,400.68	2,411.32	1,471.54	1,438.5	1,530.78	2,232.1	811.8	875.66	853.13	1,074.1	889.57	895.82	1,172.62
100-40	1,505.99	2,627.37	1,539.7	1,498.69	1,598.94	2,434.79	850.17	892.3	903.11	1,256.24	910.7	919.9	1,251.768
100-35	1,596.03	2,907.55	1,612.25	1,573.79	1,671.49	2,699.34	916.1	913.44	960.71	1,359.79	910.7	919.9	1,312.93
100-30	1,614.5	3,073.79	1,653.63	1,608.7	1,712.86	2,856.63	937.84	913.44	994.92	1,451.23	910.7	919.9	1,379.08
100-25	1,638.32	3,258.67	1,707.43	1,660.83	1,766.66	3,029.47	972.07	913.44	1,064.4	1,523.83	910.7	919.9	1,405.32
100-20	1,640.42	3,373.64	1,730.28	1,683.14	1,789.51	3,136.06	972.65	913.44	1,082.23	1,559.19	910.7	919.9	1,418.94
100-15	1,640.42	3,437.33	1,739.77	1,692.35	1,799.01	3,195.37	975.73	913.44	1,082.23	1,559.19	910.7	919.9	1,420.54
100-10	1,640.42	3,507.76	1,739.77	1,692.35	1,799.01	3,260.69	975.73	913.44	1,082.23	1,559.19	910.7	919.9	1,420.54
100-5	1,640.42	3,565.37	1,739.77	1,701.87	1,799.01	3,314.28	975.73	913.44	1,082.23	1,559.19	910.7	919.9	1,420.54

tical standpoint. In the following, we summarize our point-of-view and discuss implications for our results.

- Inadequate TSR approaches can be considered competitors of adequate ones although they relax constraints on test requirements (*i.e.,* statement coverage). This is relevant for the researcher interested in further studying this point, but it is even more relevant for the practitioner. In fact, the practitioner can adopt inadequate TSR approaches being conscious that they will substantially gain in reduction in TS size with a small effect on fault-detection capability.

- The results of our experiment help a more informed decision on the approaches to be chosen to perform TSR during regression testing. The practitioner might be interested in our results because she can choose the approach being conscious of the effects that the CB approach and those traditional have on size, code coverage, and fault-detection capability of the reduced TSs. In this regard, we also performed an analysis of the time these approaches need to be executed (see Appendix B). For example, if the practitioner is interested in reducing the

time to execute TSR, as well as regression testing, at the cost of a small loss in fault-detection capability, CB_{Cos} might be the choice. On the other hand, if the practitioner is interested in maximizing fault-detection capability and reducing the time to execute a TSR approach, then ILP_A and $CB_{K\ A}$ might be the right option.

- Among the inadequate TSR approaches (fixing an inadequacy level at 95%) the CB instances (except $CB_{SK\ I}$ and $CB_{Lev\ I}$) and $2OPT_I$ and GE_I reach a good compromise between reduction in fault-detection capability and reduction in the size of the original TS. These approaches reduce about 30% (on average) the original TS with a negligible effect on the fault-detection capability. This outcome is relevant for the practitioner because she can reduce the time to perform regression testing with a small effect on fault-detection capability. This is also interesting for the researcher in studying how to reduce more TS size without affecting fault-detection capability. For example, the researcher could be interested in studying different clustering algorithms to group test cases and different test requirements and possibly their combination. Our results seem to justify further research on this point.

- We observed that the experimental objects affect less the reduction in fault-detection capability in the case of CB instances (with the only exceptions of $CB_{SK\ I}$ and $CB_{Lev\ I}$). This outcome is clearly relevant for the practitioner because it seems that CB instances are less sensitive to the systems and their TSs. The researcher could be interested in studying why some dissimilarity measures perform better than the other (*e.g.,* Hamming distance vs. Levenshtein distance). A first step in this direction has been performed analyzing the diversity of the reduced TSs produced by the inadequate approaches. However, further research is needed on this matter.

- CB instances are less sensitive to the inadequacy level. That is, small variations in the inadequacy levels slightly affect the reduction in both TS size and fault-detection capability. This result is clearly relevant for the practitioner because she can be aware that small inadequacy level variations could reduce

the time to perform regression testing with a slight effect on the fault-detection capability of the reduced TSs. This outcome is also relevant for the researcher interested in further studying approaches less sensitive to inadequacy level. In fact, our study justifies further research on the relation between sensitiveness and the goodness of inadequate TSR approaches.

- The use of the studied approaches does not require a complete and radical change process within a software company. This is relevant for the practitioner. In fact, the diffusion of a new technology/method is made easier when empirical evaluations are performed and their results show that such a technology/method solves actual issues [45]. This is why the results of our study could promote the transferring of the developed technology to the software industry. This is of particular interest for the practitioner, while the researcher could be interested in identifying opportunities (*e.g.,* industrial case studies and experiments) to speed up this process.

6. Final Remarks

We investigate several instances of the CB approach and also compare them with well-known traditional adequate TSR approaches and their inadequate variants. The CB approach groups test cases that are similar. Test cases are similar if they cover nearly the same statements, namely, they satisfy nearly the same test requirements. To estimate such a similarity, we considered several measures. A hierarchical agglomerative clustering is applied to group test cases. A reduced TS will contain a test case for each of the identified clusters. For each cluster, the approach chooses the test case that covers the largest number of statements. We founded our investigation on a public dataset. The most important take-away results are: *(i)* there is not an inadequate TSR approach that performs the best when fixing an inadequacy level even if the CB instances seem more promising; *(ii)* the CB instances and a few traditional inadequate approaches outperform adequate ones in terms of reductions in TS size with a negligible effect on their capability in the fault-detection; and *(iii)* the CB

instances are less sensitive than other inadequate TSR approaches to inadequacy level.

Appendix A. Further Analysis on Reduction in Test Suite Size and Reduction in Fault-detection Capability at Different Inadequacy Levels

In Figures A.9 and A.10, we show the boxplots of the RS and RF values of each inadequate TSR approach when varying inadequacy levels from 60% to 95%. As for RF, we can observe that using 95% as inadequacy level all the approaches allow detecting nearly the same faults as the original TSs—all the boxes are lines crossing the zero. This inadequacy level allows obtaining a good average reduction of the original TSs equal to 25.7% (see Table 4). When lowering the inadequacy level, the reduction in TS size improves at the cost of a loss in fault-detection capability that is even more clear for the traditional TSR approaches (see Figure A.9 and Figure A.10, respectively). For example, if we consider an inadequacy level equal to 90%, we can observe that, while the TSR approaches reduce the size of TSs more as compared with an inadequacy level equal to 95%, their reduction in fault-detection capability gets worse. Summing up, 95% inadequacy level allows obtaining a good reduction of the TSs with a fault-detection capability very close to that of the original TS.

Appendix B. Further Analysis on Execution Time

In Table B.12, we report some descriptive statistics (*i.e.,* median, mean, SD, and CI) about the time needed to apply the adequate TSR approaches to the experimental objects. We focused on the adequate variants because given an approach its inadequate variant is less computation expensive than its adequate counterpart. Considering only adequate approaches allows us to include also ILP_A in the analysis presented in this section. To compute the time needed to reduce a TS does not include the time to gather coverage information because it is the same for each TSR approach studied. The experiment has been conducted

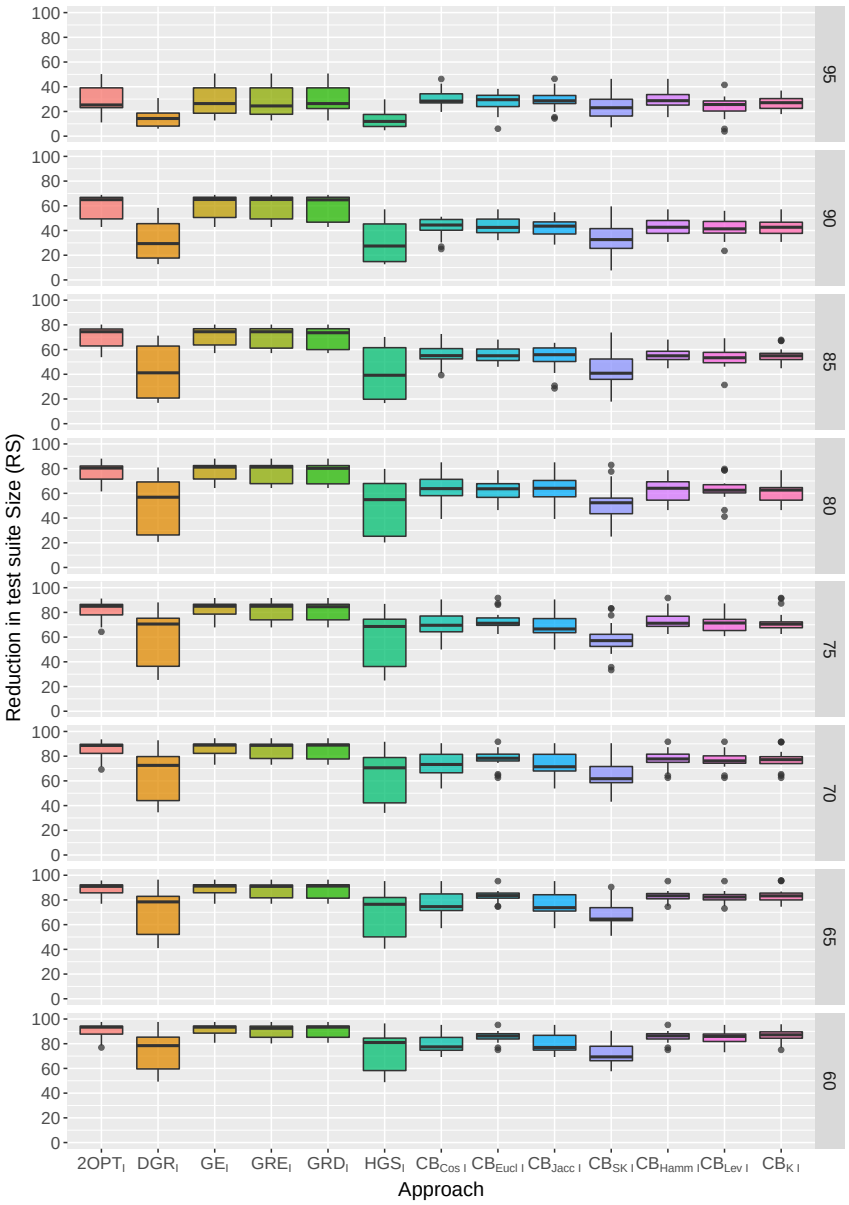

Figure A.9: Boxplots of the RS values of each inadequate TSR approach when varying inadequacy levels from 60% to 95%.

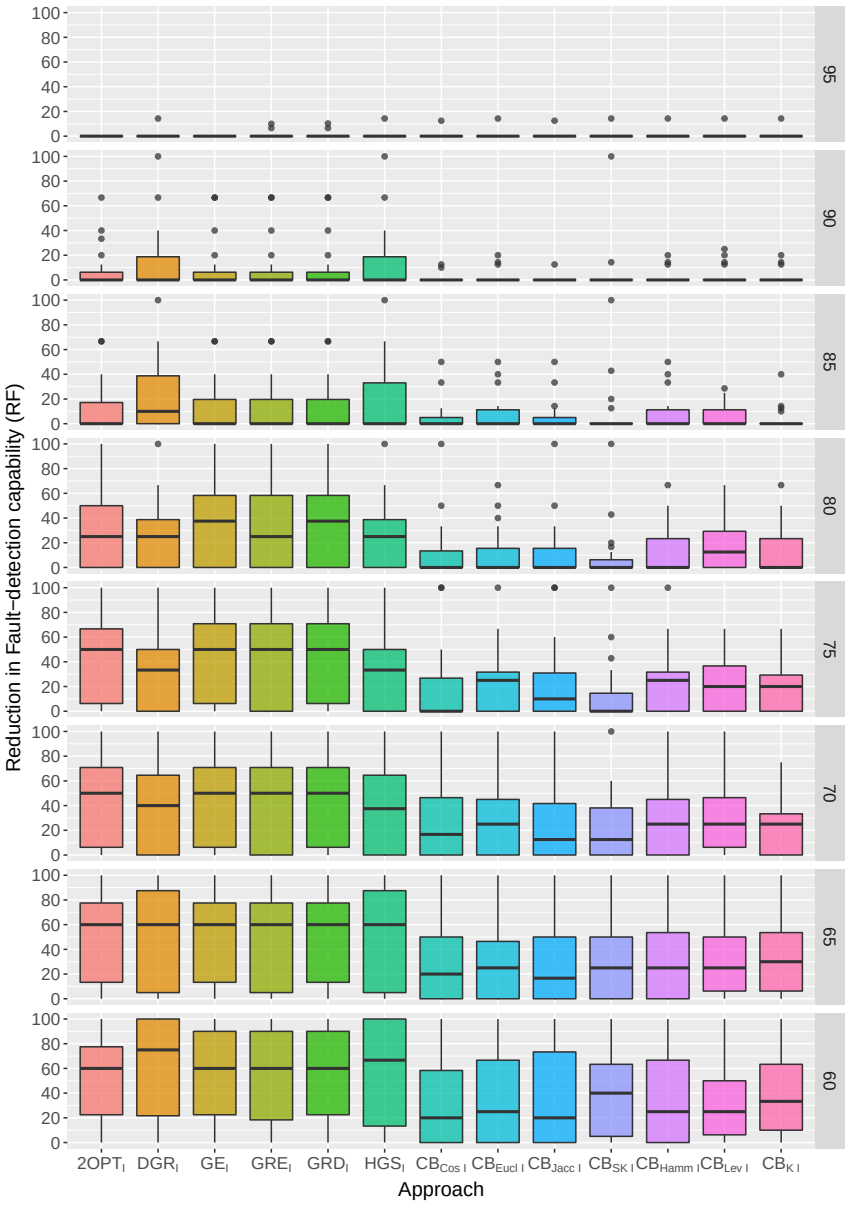

Figure A.10: Boxplots of the RF values of each inadequate TSR approach when varying inadequacy levels from 60% to 95%.

Table B.12: Some descriptive statistics for the execution time (expressed in milliseconds) of each adequate TSR approach.

	Execution Time			
Approach	median	mean	SD	CI
$2OPT_A$	826	19,469.74	39,897.74	[239.64;38,699.83]
DGR_A	546	4,067.42	6,941.87	[721.55;7,413.29]
GE_A	525	5,798.95	10,805.93	[590.65;11,007.24]
GRE_A	564	5,886.84	11,010.65	[579.88;11,193.81]
GRD_A	504	5,781.42	10,762.39	[594.11;10,968.73]
HGS_A	349	2,815.11	4,704.79	[547.47;5,082.74]
ILP_A	72	176.74	196.58	[81.99;271.49]
$CB_{Cos\ A}$	102	362.58	450.36	[145.51;579.64]
$CB_{Eucl\ A}$	126	335.88	384.69	[150.44;521.27]
$CB_{Jacc\ A}$	36,197	70,090.68	86,374.88	[28,459.33;111,722.04]
$CB_{SK\ A}$	68,999	157,796.79	194,949.13	[63,834.32;251,759.26]
$CB_{Hamm\ A}$	86	343.3	436.18	[133.06;553.53]
$CB_{Lev\ A}$	13,918	24,843.53	29,529.42	[10,610.8;39,076.25]
$CB_{K\ A}$	109	271.74	305.59	[124.45;419.03]
Mean	8,780.21	21,288.62	28,339.3	-

with a PC equipped with 2.40 GHz Intel Core i7 as CPU, 12 GB of RAM, and Windows 10 (64-bit) as the operating system.

Descriptive statistics suggest that one of the faster approaches is ILP_A. However, we can observe a small difference among this approach and four out seven instances of the CB approach: $CB_{Cos\ A}$, $CB_{Eucl\ A}$, $CB_{Hamm\ A}$, and $CB_{K\ A}$. Among these instances, the best instance seems to be $CB_{K\ A}$ because the median and mean values and CI are lower. The remaining CB instances (*i.e.*, $CB_{Jacc\ A}$, $CB_{SK\ A}$, and $CB_{Lev\ A}$) are also slower than the studied traditional TSR approaches. It is also worth mentioning that the traditional approaches are all comparable to one another with the only exception of $2OPT_A$ that exhibits higher values for median and mean. Also, the CI is worse for $2OPT_A$.

References

[1] S. Yoo, M. Harman, Regression testing minimization, selection and prioritization: A survey, Softw. Test. Verif. Reliab. 22 (2) (2012) 67–120.

[2] G. Rothermel, M. J. Harrold, J. von Ronne, C. Hong, Empirical studies of test-suite reduction, Softw. Test. Verif. Reliab. 12 (4) (2002) 219–249.

[3] S. Biswas, R. Mall, M. Satpathy, S. Sukumaran, Regression test selection techniques: A survey., Informatica 35 (3) (2011) 289–321.

[4] G. Rothermel, M. J. Harrold, Analyzing regression test selection techniques, IEEE Trans. Softw. Eng. 22 (8) (1996) 529–551.

[5] S. Romano, G. Scanniello, G. Antoniol, A. Marchetto, Spiritus: a simple information retrieval regression test selection approach, Information & Software Technology 99 (2018) 62–80.

[6] S. Elbaum, A. G. Malishevsky, G. Rothermel, Test case prioritization: A family of empirical studies, IEEE Trans. Softw. Eng. 28 (2) (2002) 159–182.

[7] A. Marchetto, M. M. Islam, W. Asghar, A. Susi, G. Scanniello, A multiobjective technique to prioritize test cases, IEEE Transactions on Software Engineering 42 (10) (2016) 918–940.

[8] A. Shi, A. Gyori, M. Gligoric, A. Zaytsev, D. Marinov, Balancing tradeoffs in test-suite reduction, in: Proceedings of International Symposium on Foundations of Software Engineering, ACM, 2014, pp. 246–256.

[9] C. Coviello, S. Romano, G. Scanniello, A. Marchetto, G. Antoniol, A. Corazza, Clustering Support for Inadequate Test Suite Reduction, in: IEEE International Conference on Software Analysis, Evolution and Reengineering, IEEE Computer Society, 2018, pp. 95–105.

[10] C. Coviello, S. Romano, G. Scanniello, An Empirical Study of Inadequate and Adequate Test Suite Reduction Approaches, in: Proceedings of Inter-

national Conference on Empirical Software Engineering and Measurement, ACM, 2018, pp. 12:1–12:10.

[11] M. J. Harrold, R. Gupta, M. L. Soffa, A methodology for controlling the size of a test suite, ACM Trans. Softw. Eng. Methodol. 2 (3) (1993) 270–285.

[12] Z. Li, M. Harman, R. M. Hierons, Search algorithms for regression test case prioritization, IEEE Trans. Softw. Eng. 33 (4) (2007) 225–237.

[13] S. Tallam, N. Gupta, A concept analysis inspired greedy algorithm for test suite minimization, in: Proceedings of Program Analysis for Software Tools and Engineering, ACM, 2005, pp. 35–42.

[14] T. Chen, M. Lau, Heuristics towards the optimization of the size of a test suite, Vol. 14, WIT Press, 1970.

[15] H. P. Williams, Model building in mathematical programming, John Wiley & Sons, 2013.

[16] H. Do, S. G. Elbaum, G. Rothermel, Supporting controlled experimentation with testing techniques: An infrastructure and its potential impact., Empirical Softw. Engg. 10 (4) (2005) 405–435.

[17] T. Y. Chen, M. F. Lau, Dividing strategies for the optimization of a test suite, Information Processing Letters 60 (3) (1996) 135 – 141.

[18] J. A. Jones, M. J. Harrold, Test-suite reduction and prioritization for modified condition/decision coverage, IEEE Transactions on Software Engineering 29 (3) (2003) 195–209.

[19] D. Jeffrey, N. Gupta, Test suite reduction with selective redundancy, in: Proceedings of International Conference on Software Maintenance, IEEE Computer Society, 2005, pp. 549–558.

[20] S. K. Mohapatra, S. Prasad, Minimizing test cases to reduce the cost of regression testing, in: Proceedings of International Conference on Com-

puting for Sustainable Global Development (INDIACom), IEEE Computer Society, 2014, pp. 505–509.

[21] A. Marchetto, G. Scanniello, A. Susi, Combining code and requirements coverage with execution cost for test suite reduction, IEEE Transactions on Software Engineering 45 (4) (2019) 363–390. doi:10.1109/TSE.2017.2777831.

[22] A. M. Smith, G. M. Kapfhammer, An empirical study of incorporating cost into test suite reduction and prioritization, in: Proceedings of Symposium on Applied Computing, ACM, 2009, pp. 461–467.

[23] J. Black, E. Melachrinoudis, D. Kaeli, Bi-criteria models for all-uses test suite reduction, in: Proceedings of the 26th International Conference on Software Engineering, ICSE '04, IEEE Computer Society, Washington, DC, USA, 2004, pp. 106–115.

[24] S. Parsa, A. Khalilian, Y. Fazlalizadeh, A new algorithm to test suite reduction based on cluster analysis, in: Proceedings of International Conference on Computer Science and Information Technology, IEEE Computer Society, 2009, pp. 189–193.

[25] A. Khalilian, S. Parsa, Bi-criteria test suite reduction by cluster analysis of execution profiles, in: Proceedings of IFIP TC 2 Central and East European Conference on Advances in Software Engineering Techniques, Springer-Verlag, 2012, pp. 243–256.

[26] S. Yoo, M. Harman, P. Tonella, A. Susi, Clustering test cases to achieve effective and scalable prioritisation incorporating expert knowledge, in: Proceedings of International Symposium on Software Testing and Analysis, ACM, 2009, pp. 201–212.

[27] S. Prasad, M. Jain, S. Singh, C.Patvardhan, A multi coverage criteria test suite minimization technique, International Journal of Applied Information Systems 1 (8) (2012) 5–11.

[28] R. Carlson, H. Do, A. Denton, A clustering approach to improving test case prioritization: An industrial case study, in: Proceedings of International Conference on Software Maintenance, IEEE Computer Society, 2011, pp. 382–391.

[29] M. J. Arafeen, H. Do, Test case prioritization using requirements-based clustering, in: Proceedings of International Conference on Software Testing, Verification and Validation, IEEE Computer Society, 2013, pp. 312–321.

[30] Y. Yang, X. Guan, J. You, Clope: A fast and effective clustering algorithm for transactional data, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, 2002, pp. 682–687.

[31] C. D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, Cambridge, 2008.

[32] A. Karatzoglou, I. Feinerer, Kernel-based machine learning for fast text mining in r, Computational Statistics & Data Analysis 54 (2) (2010) 290–297.

[33] C. Coviello, S. Romano, G. Scanniello, CUTER: clustering-based test suite reduction, in: Proceedings of the International Conference on Software Engineering, 2018, pp. 306–307.

[34] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, A. Wesslén, Experimentation in Software Engineering, Springer, 2012.

[35] V. Basili, G. Caldiera, D. H. Rombach, The Goal Question Metric Paradigm, Encyclopedia of Software Engineering, John Wiley and Sons, 1994.

[36] L. Zhang, D. Marinov, L. Zhang, S. Khurshid, An empirical study of junit test-suite reduction, in: Proceedings of International Symposium on Software Reliability Engineering, IEEE Computer Society, 2011, pp. 170–179.

[37] W. E. Wong, J. R. Horgan, S. London, A. P. Mathur, Effect of test set minimization on fault detection effectiveness, in: Proceedings of International Conference on Software Engineering, ACM, 1995, pp. 41–50.

[38] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, Journal of the American Statistical Association 32 (200) (1937) 675–701.

[39] A. M. Smith, J. Geiger, G. M. Kapfhammer, M. L. Soffa, Test suite reduction and prioritization with call trees, in: Proceedings of Symposium on Applied Computing, ACM, 2007, pp. 539–540.

[40] A. Shi, T. Yung, A. Gyori, D. Marinov, Comparing and combining test-suite reduction and regression test selection, in: Proceedings of Joint Meeting on Foundations of Software Engineering, ACM, 2015, pp. 237–247.

[41] A. Panichella, R. Oliveto, M. Di Penta, A. De Lucia, Improving multiobjective test case selection by injecting diversity in genetic algorithms, IEEE Trans. Software Eng. 41 (4) (2015) 358–383.

[42] B. Jiang, Z. Zhang, T. H. Tse, T. Y. Chen, How well do test case prioritization techniques support statistical fault localization, in: Proceedings of International Computer Software and Applications Conference, IEEE Press, 2009, pp. 99–106.

[43] J. Romano, J. D. Kromrey, J. Coraggio, J. Skowronek, Appropriate statistics for ordinal level data: Should we really be using t-test and Cohen'sd for evaluating group differences on the NSSE and other surveys?, in: annual meeting of the Florida Association of Institutional Research, 2006, pp. 1–3.

[44] N. Cliff, Ordinal methods for behavioral data analysis, Psychology Press, 1996.

[45] S. L. Pfleeger, W. Menezes, Marketing technology to software practitioners, IEEE Softw. 17 (1) (2000) 27–33.