HKU CS Tech Report TR-2015-01

Postprint of article in *International Journal of Web Services Research, 12*(3): 1–24 (2015)

Connecting the Average and the Non-Average: A Study of the Rates of Fault Detection in Testing WS-BPEL Services*

Changjiang Jia, City University of Hong Kong, Tat Chee Avenue, Hong Kong
& National University of Defense Technology, Changsha, China

Lijun Mei, IBM Research—China, Beijing, China

W.K. Chan, City University of Hong Kong, Tat Chee Avenue, Hong Kong

Yuen Tak Yu, City University of Hong Kong, Tat Chee Avenue, Hong Kong

T.H. Tse, The University of Hong Kong, Pokfulam, Hong Kong

ABSTRACT

Many existing studies measure the effectiveness of test case prioritization techniques using the average performance on a set of test suites. However, in each regression test session, a real-world developer may only afford to apply one prioritization technique to one test suite to test a service once, even if this application results in an adverse scenario such that the actual performance in this test session is far below the average result achievable by the same technique over the same test suite for the same application. It indicates that assessing the average performance of such a technique cannot provide adequate confidence for developers to apply the technique. The authors ask a couple of questions: To what extent does the effectiveness of prioritization techniques in average scenarios correlate with that in adverse scenarios? Moreover, to what extent may a design factor of this class of techniques affect the effectiveness of prioritization in different types of scenarios?

To the best of their knowledge, the authors report in this paper the first controlled experiment to study these two new research questions through more than 300 million APFD and HMFD data points produced from 19 techniques, eight WS-BPEL benchmarks and 1000 test suites prioritized by each technique 1000 times. A main result reveals a strong and linear correlation between the effectiveness in the average scenarios and that in the adverse scenarios. Another interesting result is that many pairs of levels of the same design factors significantly change their relative *strengths of being more effective within the same pairs in handling a wide spectrum of prioritized test suites produced by the same techniques over the same test suite in testing the same benchmarks, and the results obtained from the average scenarios are more similar to those of the more effective end than otherwise. This work provides the first piece of strong evidence for the research community to re-assess how they develop and validate their techniques in the average scenarios and beyond.*

Keywords: *Adverse Scenario, Average Scenario, Controlled Experiment Correlation, Empirical Study, Regression Testing, Test Case Prioritization, WS-BPEL, XML-based Artifact*

INTRODUCTION

In an ecosystem of services, a WS-BPEL program (WS-BPEL Version 2.0, 2007) offers a service by invoking external services (Mei et al. 2014b) to implement its workflow steps. If the business requirements of the ecosystem for the program are not met, its service consumers may discontinue consuming the service it provides, and switch to competing services dynamically. A conventional wisdom is that if a customer discards a product or service, it is intuitively difficult to attract the same customer to reuse the same product or service. Hence, to stay competitive, developers need to rapidly maintain and deploy the service to meet these latest business requirements. Moreover, each modified service should be rapidly and thoroughly tested to reduce the potential impact of any latent fault on its consumers. In short, from the testing viewpoint, maintaining such a service demands highly efficient test sessions, and every test session should be as efficient as possible.

In a regression test session (Leung and White 1989; Onoma et al. 1998), developers execute a modified service over a suite of regression test cases to assess to what extent this modified service passes the regression test. Suppose that these test cases have been prioritized (Rothermel et al. 2001; Rothermel et al. 2002), meaning that some test cases are scheduled to execute earlier than others, in order to expose all the faults in the service detectable by these test cases as fast as possible. Developers would expect that executing these test cases according to their priority can quickly expose faults *in their own situations*.

Existing studies on test case prioritization (Do et al. 2004; Elbaum et al. 2002; Mei et al. 2014b), or TCP for short, evaluate various design factors of TCP techniques or these techniques directly based on their effectiveness *on average* (that is, mean or median effectiveness statistically). Yet, in practice, in a regression test session, a developer only applies at most *one* test case prioritization technique to *one* test suite *once* to test the same version of the same program. The same developers do not have the luxury to apply multiple test suites or the same test suite multiple times to look for the average effectiveness of the technique on the service under test.

Thus, even when the average effectiveness of a TCP technique or a design factor in TCP is excellent, if the technique or the factor performs ineffectively in scenarios that are far below average (hereafter simply referred to as the *adverse scenarios*), the technique or the factor may not be reliably used in practice. This problem is general across a wide range of software domains, and we are particularly interested in it within the regression testing of services.

In the preliminary version of this paper (Jia et al. 2014), we reported the *first* controlled experiment investigating whether the effectiveness results (the rate of fault detection measured by APFD (Elbaum et al. 2002)) of both TCP techniques and their design factors can be extrapolated from the average scenarios to the adverse scenarios. The controlled experiment included 10 TCP techniques plus two control techniques, eight benchmarks, and 100 test suites per benchmark with 100 repeated applications of every TCP technique on every such test suite. In total, we computed 0.96 million raw APFD values. It compared the consistency between the whole effectiveness dataset of each technique against that of random ordering (labeled as C1 in the Preliminaries section of this paper) and the dataset consisting of the lowest 25 percentile of the former dataset. The results showed that less than half of all the techniques and factors exhibited such consistency.

This paper significantly extends the preliminary version (Jia et al. 2014). We investigate two completely new research questions (denoted by RQ1 and RQ2), which are refined from the research questions presented in the preliminary version. RQ1 investigates the correlation of the change of effectiveness (relative to random ordering) between the average scenarios and adverse scenarios. RQ2 investigates whether switching among levels of a design factor of TCP techniques may result in statistically distinguishable changes of effectiveness in handling a wide spectrum of prioritized test suites produced by the same techniques over the same test suite in testing the same benchmarks. The new controlled experiment uses 19 TCP techniques, eight benchmarks, 1000 test suites per benchmark, and applies every TCP technique to every test suite 1000 times, producing 152 million APFD values and 152 million HMFD values (where HMFD is another measure of TCP effectiveness (Zhai et al. 2014)).

For RQ1, the main result shows that the prioritization effectiveness in the average scenarios has a *strong linear correlation* with the effectiveness in the adverse scenarios. Moreover, with respect to random ordering, we find that in handling a test suite where a technique is (in-)effective in the average scenarios, the same technique tends to amplify the corresponding (in-)effectiveness in the adverse scenarios; and such an amplification effect is stronger on the ineffective side than on the effective side. For RQ2, we find that many pairs of levels of the same TCP design factors significantly affect the effectiveness in handling a wide spectrum of prioritized test suites produced by the same techniques over the same test suite in testing the same benchmarks. We also find that the results obtained from the average scenarios are more similar to those of the more effective end of the above spectrum than otherwise.

The main contribution of this paper is twofold. (i) To the best of our knowledge, this paper is the *first* work studying the extent that the effectiveness of test case prioritization techniques in average scenarios correlates with that in adverse scenarios, and the extent that TCP design factors exhibit statistically distinguishable and conflicting results in different regions in the effectiveness spectra above. (ii) This paper also reports the *first* large-scale controlled experiment to study the two research questions.

The rest of the paper is organized as follows. The next section describes the preliminaries. After that, we formulate the research questions, followed by describing the setup of the experiment. Then, we present the data analyses. Finally, we review related work and conclude the paper.

PRELIMINARIES

Test case prioritization (TCP) techniques can be designed to achieve certain goals in a regression test, such as to improve the rate of code coverage or the fault detection rate. The *test case prioritization problem* has been formally defined in (Elbaum et al. 2002), which we adapt as follows:

Given: *T*, a test suite; *PT*, a set of permutations of *T*; and *f*, a function from *PT* to real numbers.

Objective: To find a reordered test suite $T'\in PT$ such that $\forall T''\in PT$, $f(T') \geq f(T'')$.

To measure how effectively TCP techniques achieve the goal of obtaining higher fault detection rates, the weighted *Average of the Percentage of Faults Detected* (APFD) (Elbaum et al. 2002) and the *Harmonic Mean of Rate of Fault Detection* (HMFD) (Zhai et al. 2014) are two metrics we use in our experiments.

Let T be a test suite containing n test cases, F be a set of m faults revealed by T, and TF_i be the index of the first test case in the reordered test suite T' that reveals fault i. The following two equations compute the APFD value and HMFD value of T', respectively.

$$APFD = 1 - \frac{TF_1 + TF_2 + \cdots + TF_m}{nm} + \frac{1}{2n}$$

$$HMFD = \frac{m}{\frac{1}{TF_1} + \frac{1}{TF_2} + \cdots + \frac{1}{TF_m}}$$

A higher APFD value (or a lower HMFD value) indicates a higher fault detection rate. In this paper, the function f maps every permutation T' in PT to the APFD or HMFD value of T'.

We follow (Elbaum et al. 2002; Rothermel et al. 2001) to compare TCP techniques with *random ordering* as a control technique, defined as follows:

C1: Random ordering. This technique randomly orders the test cases in a test suite T.

RESEARCH QUESTIONS

We study two main research questions in this paper.

RQ1: To what extent does the effectiveness of prioritization techniques for service regression testing in the average scenarios correlate with that in the adverse scenarios?

RQ2: To what extent do different levels of the same factors in designing test case prioritization techniques for service regression testing exhibit significantly different effectiveness results with respect to different regions of effectiveness results of random ordering?

RQ1 has a significant impact on regression testing research on programs in general and workflow-based services in particular. As stated in the Introduction section, many pieces of such work have reported their effectiveness results in average scenarios. The result of RQ1 augments a large body of such work to help them to extrapolate their results to handle adverse scenarios.

RQ2 examines how to design TCP techniques for service regression testing with different focuses of optimization in mind. For instance, configuring a factor to one particular level may be more effective in one region but configuring the same factor to another level may be more effective in another region. Discovering this information can help researchers investigate whether designing an adaptive TCP technique is necessary.

SETUP OF EXPERIMENTS

This section presents the setup of the experiments, including the description of the benchmark and test suites, TCP techniques, and experimental procedures.

Benchmark and Test Suites

We included eight representative service-based subjects (Mei et al. 2008) as our benchmarks. They are all developed in WS-BPEL. Table 1 shows the statistics of these benchmarks. These benchmarks have been used in other service regression testing studies (Jia et al. 2014; Mei et al. 2011; Mei et al. 2014b).

We used a set of faults and associated test suites for each benchmark to measure the effectiveness of different TCP techniques. For each subject, we generated the faulty versions by seeding one fault with three typical types of mutations (Andrews et al. 2005): value mutation, decision mutation, and statement mutation. The statistics of the faults in the modified versions have been reported in Mei et al. (2014a). Since BPEL can be treated as Control Flow Graphs (CFGs), the mutations were performed in the same way as seeding faults in CFGs. An XPath fault is a wrong usage of XPath expressions, such as extracting the wrong content or failing to extract any content. Similarly, a WSDL fault is a wrong usage of the WSDL specifications such as binding to a wrong WSDL specification or an inconsistent message definition.

Table 1. Benchmarks and their descriptive statistics

Ref.	Application	Modified Versions	Elements	LOC	XPaths	XRG Branches	WSDL Elements	Used Versions
A	atm	8	94	180	3	12	12	5
B	buybook	7	153	532	3	16	14	5
C	dslservice	8	50	123	3	16	20	5
D	gymlocker	7	23	52	2	8	8	5
E	loanapproval	8	41	102	2	8	12	7
F	marketplace	6	31	68	2	10	10	4
G	purchase	7	41	125	2	8	10	4
H	triphandling	9	94	170	6	36	20	8
	Total	60	527	1352	23	114	106	43

For each subject, we constructed a test pool that included 1000 randomly generated test cases. Then, we randomly selected test cases from the pool one by one and put it into a test suite (which was initially empty) until all the workflow branches, XRG branches, and WSDL elements had been covered at least once. This process was the same as that in the test suite construction in Elbaum et al. (2002) and Mei et al. (2014a) except that we used the adequacy on BPEL, XRG and WSDL instead of that on program statements as the stopping criterion. We repeated this process for each subject 1000 times. In total, we constructed 1000 test suites for each subject.

Table 2 shows the maximum, mean, and minimum sizes of these test suites. We followed existing work (Do et al. 2004; Elbaum et al. 2002; Mei et al. 2011) to exclude a faulty version from data analyses if more than 20 percent of the test cases detected the fault from the version. The numbers of faulty versions actually used are shown in the rightmost column of Table 1. For each generated test suite, we further marked which test case reveals which fault. That is, to determine whether the test case revealed a fault, our tool compared its execution result against the original subject program with its result against a faulty version. If there is any difference, we deemed that the output of the faulty version revealed a fault.

Table 2. Sizes of test suites for each subject

Size \ Ref.	A	B	C	D	E	F	G	H	Average
Max	182	128	157	165	197	139	151	149	158.5
Mean	78	48	64	85	155	43	84	85	80.3
Min	20	12	15	18	43	23	16	26	21.6

Test Case Prioritization Techniques

In total, we included 18 prioritization techniques shown in Table 3. In addition to the 10 techniques M1–M10 presented in (Jia et al. 2014), we further included eight more techniques M11–M18 taken from Mei et al. (2014a), which applied a *Refinement-Oriented Level-Exploration* (ROLE) strategy to refine two techniques (Total-BPEL-Workflow and Addtl-BPEL-Workflow) in Elbaum et al. (2002). We will briefly review the process of applying the ROLE strategy.

The ROLE strategy relies on a multilevel coverage model (Mei et al. 2014a), which is a six-tuple $\langle T, \Pi_\alpha, \Pi_\beta, \Pi_\gamma, \Pi_\delta, \Pi_\theta\rangle$ for a service P, where (a) T is a regression test suite for P and (b) Π_α, Π_β, Π_γ, Π_δ, and Π_θ represent, respectively, sets of workflow branches, sets of XRG branches, sets of WSDL elements, sets of XRG patterns, and sets of tag values and unique tags in XML messages collected from the executions of all the test cases in T against P. For any test case t in T, Π_α, Π_β, Π_γ, Π_δ, and Π_θ represent, respectively, the set of workflow branches, the set of XRG branches, the set of WSDL elements, the set of XRG patterns, and the set of tag values and unique tags in XML messages covered by the execution of t against P. The five levels are referred to as CM*i* levels, where CM stands for Coverage Model and i = 1 to 5.

With the multilevel coverage model, the ROLE strategy refines a technique (M9 and M10 in this paper) at a level CM*i* when encountering tie cases by using the coverage in the next level CM(*i*+1). M9 and M10 conduct the prioritization with coverage data at the CM1 level. Applying the ROLE strategy to refine M9 and M10 on the other four coverage levels generated four new TCP techniques for each of M9 and M10. To simplify our presentation, we use M9-CM*i*-Refine (i = 2 to 5) to stand for the generated techniques of M9 with ROLE at the CM*i* coverage level. We also use a similar shorthand M10-CM*i*-Refine (i = 2 to 5) to stand for the generated techniques of M10 with ROLE at the CM*i* coverage level. Thus, using ROLE, these CM*i* techniques form a subsumption hierarchy (Mei et al. 2014a).

The following presents each of the 18 techniques listed in Table 3. We note that all the techniques used in our controlled experiments have been used in existing work.

M1: **Total BPEL activity coverage prioritization** (Total-BPEL-Activity). It is the same as the classical total-statement technique (Elbaum et al. 2002; Mei et al. 2011) except that M1 sorts the test cases in descending order of the total number of BPEL activities (instead of statements) executed by each test case. If multiple test cases cover the same number of BPEL activities, M1 orders them randomly.

M2: **Additional BPEL activity coverage prioritization** (Addtl-BPEL-Activity). It is the same as the classical addtl-statement technique (Elbaum et al. 2002; Mei et al. 2011), where M2 iteratively selects a test case that yields the maximum cumulative BPEL activity (instead of statement) coverage, and then removes the covered activities from the coverage information of each remaining test case. Additional iterations will be conducted until all the activities have been covered by at least one test case. If multiple test cases cover the same number of activities in the current coverage information of the test cases, M2 selects one of them randomly. Having achieved the complete coverage of all the activities by the prioritized subset of test cases in the given test suite, M2 resets the coverage information of each remaining test case to its initial value and then reapplies the algorithm to the remaining test cases.

Table 3. Prioritization techniques and factors under study:
strategy (*A: Additional; T: total; I: Iterative*), *order direction* (*A: Ascending; D: Descending*)

Index	**Name of Technique**	**Factors in Designing TCP Techniques**				
		Artifact Type	**Strategy**	**Order Direction**	**Type of Coverage Data**	**ROLE Hierarchy**
M1	Total-BPEL-Activity	BPEL	T	D	Executable	-
M2	Addtl-BPEL-Activity	BPEL	A	D	Executable	-
M3	Total-XPath-Selection	XRG	T	D	Executable	-
M4	Addtl-XPath-Selection	XRG	A	D	Executable	-
M5	Ascending-XRG-Node	XRG	I	A	Non-Executable	-
M6	Descending-XRG-Node	XRG	I	D	Non-Executable	-
M7	Ascending-WSDL-Element	WSDL	I	A	Non-Executable	-
M8	Descending-WSDL-Element	WSDL	I	D	Non-Executable	-
M9	Total-BPEL-Workflow	BPEL	I	D	Executable	CM1
M10	Addtl-BPEL-Workflow	BPEL	I	D	Executable	CM1
M11	Total-BPEL-CM2-Refine	BPEL, XRG	T	D	Executable	CM2
M12	Addtl-BPEL-CM2-Refine	BPEL, XRG	A	D	Executable	CM2
M13	Total-BPEL-CM3-Refine	BPEL, XRG, WSDL	T	D	Both	CM3
M14	Addtl-BPEL-CM3-Refine	BPEL, XRG, WSDL	A	D	Both	CM3
M15	Total-BPEL-CM4-Refine	BPEL, XRG, WSDL, XRG Pattern	T	D	Both	CM4
M16	Addtl-BPEL-CM4-Refine	BPEL, XRG, WSDL, XRG Pattern	A	D	Both	CM4
M17	Total-BPEL-CM5-Refine	BPEL, XRG, WSDL, XML, XRG Pattern	T	D	Both	CM5
M18	Addtl-BPEL-CM5-Refine	BPEL, XRG, WSDL, XML, XRG Pattern	A	D	Both	CM5

M3 and **M4**: **Total XPath selection coverage prioritization** (Total-XPath-Selection) and **Additional XPath selection coverage prioritization** (Addtl-XPath-Selection) (Jia et al. 2014). They are the same as M1 and M2, respectively, except that M3 and M4 measure test coverage in terms of XPath selections rather than BPEL activities.

M5 and **M6**: **Ascending XRG node coverage prioritization** (Ascending-XRG-Node) and **Descending XRG node coverage prioritization** (Descending-XRG-Node) (Jia et al. 2014). Each technique first partitions test cases into groups such that all the test cases with the same number of XRG nodes are placed in the same group. Suppose that the partitioning process results in $m+1$ groups G_0, G_1, ..., G_m, where G_i is a group of test cases each of which covers exactly i XRG nodes. M5 and M6 will select one test case randomly from a group starting from G_0 to G_m in ascending order and descending order of the index of the groups, respectively. It then iterates the procedure until all the test cases in all the groups have been selected.

M7 and **M8**: **Ascending WSDL element coverage prioritization** (Ascending-WSDL-Element) and **Descending WSDL element coverage prioritization** (Descending-WSDL-Element) (Mei et al. 2011). M7 and M8 are the same as M5 and M6 except that they measure test coverage in terms of the elements in WSDL documents rather than XRG nodes.

M9 and **M10**: **Total BPEL workflow coverage prioritization** (Total-BPEL-Workflow) and **Additional BPEL workflow coverage prioritization** (Addtl-BPEL-Workflow) (Do et al. 2004; Mei et al. 2011). M9 and M10 are the same as M1 and M2 except that they measure test coverage in terms of BPEL workflow transitions rather than BPEL activities. They are adapted from the classical total-branch and addtl-branch techniques presented in Elbaum et al (2002), respectively.

M11 (M9-CM2-Refine), **M13 (M9-CM3-Refine)**, **M15 (M9-CM4-Refine)**, and **M17 (M9-CM5-Refine)** (Mei et al. 2014a). M9-CM*i*-Refine (*i* = 2 to 5) is the same as M9-CM(*i* –1)-Refine (where M9-CM1-Refine means M9) except when multiple test cases cover the same number of CM(*i*–1), it reorders them in descending order of the number of CM*i* items covered by each test case involved in the tie. If there is still a tie, M9-CM*i*-Refine randomly orders the test cases involved in tie cases.

M12 (M10-CM2-Refine), **M14 (M10-CM3-Refine)**, **M16 (M10-CM4-Refine)**, and **M18 (M10-CM5-Refine)** (Mei et al. 2014a). M10-CM*i*-Refine (*i* = 2 to 5) is the same as M10-CM(*i*–1)-Refine (where M10-CM1-Refine means M10) except three things: (a) In each iteration, M10-CM*i*-Refine removes the covered CM1 to CM*i* items of the selected test cases from the remaining test cases to indicate that the removed items have been covered. (Note that M10-CM*i*-Refine still selects test cases based on the CM1 item coverage as in M10.) (b) If multiple test cases cover the same number of CM(*i*–1) items in the current round of selection, M10-CM*i*-Refine selects the test case that has the maximum number of uncovered CM*i* items. If there is still a tie, it randomly selects one of the test cases involved. (c) When resetting is needed, M10-CM*i*-Refine resets each remaining test case to the original coverage of CM1 to CM*i* items.

We applied the ROLE strategy to M9 and M10 because M9 and M10 have been evaluated in existing work (Mei et al. 2014a). M1 to M8 have not been evaluated with ROLE, and yet our controlled experiment is already very large in scale. Also, to the best of our knowledge, the classical versions of M9 and M10 are still the most effective series of TCP techniques since the inception of TCP research (Zhang et al. 2013).

Experimental Procedure

For each random test suite *T*, we applied each prioritization technique *M* 1000 times to *T* to gain statistical power on each technique. Then, for each prioritized test suite *T'*, we computed its APFD value and HMFD value. In total, we collected 152,000,000 (= 8 subjects × 1000 test suites × 19 techniques × 1000 times) APFD and HMFD items for our data analyses, respectively.

RQ1 studies the correlation between the effectiveness measures of TCP techniques in the average scenarios and the adverse scenarios for each test suite. For each test suite *T*, we define the *average scenarios* of applying technique M*i* (*i* = 1 to 18) to *T* as the set of all 1000 prioritized test suites, each being generated by M*i* on *T*. We define the *adverse scenarios* as the 250 test suites that lead to the lowest 25% prioritization effectiveness (i.e., the lowest 25 percentile of all the 1000 APFD values or the highest 25 percentile of all the 1000 HMFD values). First, we performed a statistical comparison using the Analysis of Variance (ANOVA) at the 5% significance level followed by a multiple mean comparison using Matlab (with HSD (Jia et al. 2014), which is the default option in Matlab for such comparisons) between M*i* and random ordering (C1) on *T* in the average scenarios. The comparison result can be one of the three possible cases: M*i* is significantly more effective than C1, M*i* has no significant difference from C1, and M*i* is significantly less effective than C1. We followed Jia et al. (2014) to use ">", "=", and "<" to represent the above three cases, respectively. We also want to study whether M*i* in the adverse scenarios is statistically more effective than the mean effectiveness of C1. Thus, we first computed the mean effectiveness of C1 in the average scenarios. Then, we performed a statistical comparison between M*i* in the adverse scenarios and the mean effectiveness of C1. Such a comparison also results in three similar cases. By doing so, for each test suite *T*, we can find the relative distribution of comparison results between M*i* and C1 in the average scenarios (which we take as the *x* dimension) and the comparison results in the adverse scenarios (which we take as the *y* dimension).

RQ2 studies the effect of factors on prioritization effectiveness with respect to the whole spectrum of random ordering. In many existing studies, a TCP technique is viewed as a combination of chosen levels of different factors. Table 3 shows the possible values of the five factors identified by previous work for the 18 techniques studied in our controlled experiment.

To study whether the factor significantly affects the prioritization effectiveness, we compared two techniques M*i* and M*j* that contain different levels of the same factor. Table 4 shows each factor, the possible pairwise level comparisons of the factors, and the pair of techniques corresponding to the pairwise level comparisons of the factors. The rightmost column of the table shows the number of technique-pairs in the same row.

Table 4. Pairwise comparisons of levels and techniques for each factor

Factor	**Pairwise Level Comparison**	**Technique-Pair**	**No. of Pairs**
Artifact type	BPEL-XRG	M*i*–M*j*, where M*i* ∈{M1, M2, M9, M10} and M*j* ∈{M3, M4, M5, M6}	16
	BPEL-WSDL	M*i*–M*j*, where M*i* ∈{M1, M2, M9, M10} and M*j* ∈{M7, M8}	8
	XRG-WSDL	M*i*–M*j*, where M*i* ∈{M3, M4, M5, M6} and M*j* ∈{M7, M8}	8
Strategy	Total-Additional	M*i*–M*j*, where M*i* ∈{M1, M3, M9, M11, M13, M15, M17} and M*j* ∈{M2, M4, M10, M12, M14, M16, M18}	49
	Total-Iterative	M*i*–M*j*, where M*i* ∈{M1, M3, M9, M11, M13, M15, M17} and M*j* ∈{M5, M6, M7, M8}	28
	Additional-Iterative	M*i*–M*j*, where M*i* ∈{M2, M4, M10, M12, M14, M16, M18} and M*j* ∈{M5, M6, M7, M8}	28
Order direction	Ascending-Descending	M*i*–M*j*, where M*i* ∈{M5, M7} and M*j* ∈{M1, M2, M3, M4, M6, M8, M9, M10, M11, M12, M13, M14, M15, M16, M17, M18}	32
Coverage data	Executable-Nonexecutable	M*i*–M*j*, where M*i* ∈{M1, M2, M3, M4, M9, M10, M11, M12} and M*j* ∈{M5, M6, M7, M8}	32
ROLE hierarchy	CM1–CM2	M9–M11 and M10–M12	2
	CM1–CM3	M9–M13 and M10–M14	2
	CM1–CM4	M9–M15 and M10–M16	2
	CM1–CM5	M9–M17 and M10–M18	2
	CM2–CM3	M11–M13 and M12–M14	2
	CM2–CM4	M11–M15 and M12–M16	2
	CM2–CM5	M11–M17 and M12–M18	2
	CM3–CM4	M13–M15 and M14–M16	2
	CM3–CM5	M13–M17 and M14–M18	2
	CM4–CM5	M15–M17 and M16–M18	2

In our preliminary version (Jia et al. 2014), we compared techniques and factors only on the lowest 25th effectiveness percentile and the whole effectiveness. In this paper, we compare each factor in a fine-grained level. Specifically, we divide the effectiveness spectrum of M*i* into five regions (i.e., Regions 1–5). The boundaries of the five regions are defined by two parameters: the mean and standard deviation of the effectiveness of random ordering (C1). We refer to these two parameters as μ and δ, respectively. For a prioritized test suite *T'* generated by applying M*i* to a test suite *T*, we refer to its APFD effectiveness value as APFD(*T'*). Then, the boundaries of the five regions are defined as follows with the three-sigma rule (i.e., the classical 68–95–99.7 rule in statistics) on random ordering. To simplify the presentation, we define the whole set of effectiveness data as Region *All*. The five regions of APFD(*T'*) together with Region *All* are defined as follows:

- Region 1: $\mu - 3 \times \delta \leq \text{APFD}(T') < \mu - 1.5 \times \delta$.
- Region 2: $\mu - 1.5 \times \delta \leq \text{APFD}(T') < \mu - 0.5 \times \delta$.
- Region 3: $\mu - 0.5 \times \delta \leq \text{APFD}(T') < \mu + 0.5 \times \delta$.
- Region 4: $\mu + 0.5 \times \delta \leq \text{APFD}(T') < \mu + 1.5 \times \delta$.
- Region 5: $\mu + 1.5 \times \delta \leq \text{APFD}(T') < \mu + 3 \times \delta$.
- Region *All*: all effectiveness data are in this region.

In the above five regions, Region 1 is the least effective region and Region 5 is the most effective region. Then, for each M*i*, we distributed all its effectiveness data into them, and compared them pairwise to see to what extent the same factor affects the prioritization effectiveness.

We also define the corresponding six regions for HMFD result on each prioritized test suite as follows. Note that the boundaries of HMFD regions are defined with the same effectiveness semantics (i.e., from Region 1 to Region 5, the measured effectiveness increases) against that of APFD regions.

- Region 1: $\mu + 1.5 \times \delta \leq \text{HMFD}(T') < \mu + 3 \times \delta$.
- Region 2: $\mu + 0.5 \times \delta \leq \text{HMFD}(T') < \mu + 1.5 \times \delta$.
- Region 3: $\mu - 0.5 \times \delta \leq \text{HMFD}(T') < \mu + 0.5 \times \delta$.
- Region 4: $\mu - 1.5 \times \delta \leq \text{HMFD}(T') < \mu - 0.5 \times \delta$.
- Region 5: $\mu - 3 \times \delta \leq \text{HMFD}(T') < \mu - 1.5 \times \delta$.
- Region *All*: all effectiveness data are in this region.

DATA ANALYSES

Answering RQ1

In RQ1, we investigate to what extent the effectiveness of TCP techniques in the average scenarios correlates with the effectiveness in the adverse scenarios.

Table 5 shows the distribution of comparison results (*a*, *b*) of techniques with random ordering in the average scenarios and adverse scenarios. There are nine possible combinations in total. The value of *a* is the measure on the extent that M*i* is statistically more effective than C1 on *T* in the average scenarios. If M*i* is statistically more effective than, has no difference from, and is less effective than C1 at the 5% significance level, then the value of *a* is positive (>), zero (=), and negative (<), respectively. The value of *b* is for the adverse scenarios and can be interpreted similarly to the value of *a* in the average scenarios. The value in each cell is the number of comparisons for all the techniques located in the column category for the subject in the corresponding row. The last row shows the statistics of each column category for taking all the subjects as a whole.

Table 5. Distribution of comparison results in nine cases for all techniques on all test suites for each subject

	No. of Comparison Results (*a*, *b*) in Average Scenarios and Adverse Scenarios in Each of the Nine Cases																	
	APFD									**HMFD**								
Case	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9
Ref	(>, >)	(>, =)	(>, <)	(=, >)	(=, =)	(=, <)	(<, >)	(<, =)	(<, <)	(>, >)	(>, =)	(>, <)	(=, >)	(=, =)	(=, <)	(<, >)	(<, =)	(<, <)
A	8909	487	5736	0	0	2414	0	0	454	11312	282	2465	0	0	2173	0	0	1768
B	1005	173	6752	0	0	7804	0	0	2266	1311	209	6410	0	0	8013	0	0	2057
C	8393	128	4039	12	0	2863	0	0	2565	13333	32	230	0	0	2057	0	0	2348
D	13004	337	1691	0	0	979	0	0	1989	13435	411	1193	0	0	1036	0	0	1925
E	16308	204	909	0	0	141	0	0	438	17148	7	183	0	0	199	0	0	463
F	12674	239	2308	2	3	2054	0	0	720	14356	46	906	1	0	2105	0	0	586
G	14204	286	2205	0	0	210	0	0	1095	16354	10	641	1	0	260	0	0	734
H	10461	124	2429	7	0	3418	0	0	1561	12076	12	912	0	0	3402	0	0	1598
Total	84958	1978	26069	21	3	19883	0	0	11088	99325	1009	12940	2	0	19245	0	0	11479
Proportion	0.59	0.01	0.18	0.00	0.00	0.14	0.00	0.00	0.08	0.69	0.01	0.09	0.00	0.00	0.13	0.00	0.00	0.08

We observe from Table 5 that for 59% of the cases for APFD and 69% of the cases for HMFD, the studied 18 TCP techniques are consistently more effective than random ordering (see Case 1 in the (>, >) column in the table). Even considering Case 2, Case 4, and Case 5 as well, the proportions are only 60% and 70% in terms of APFD and HMFD, respectively. This indicates that for 60% of the cases for APFD and 70% of the cases for HMFD, the studied techniques performed no worse than random ordering in both the average and adverse scenarios. In other words, for 40% of the cases for APFD and 30% of the cases for HMFD, the studied techniques performed less effectively than random ordering in at least one of the two scenarios. This finding strongly shows that the studied techniques have noticeable probabilities of being less effective than random ordering.

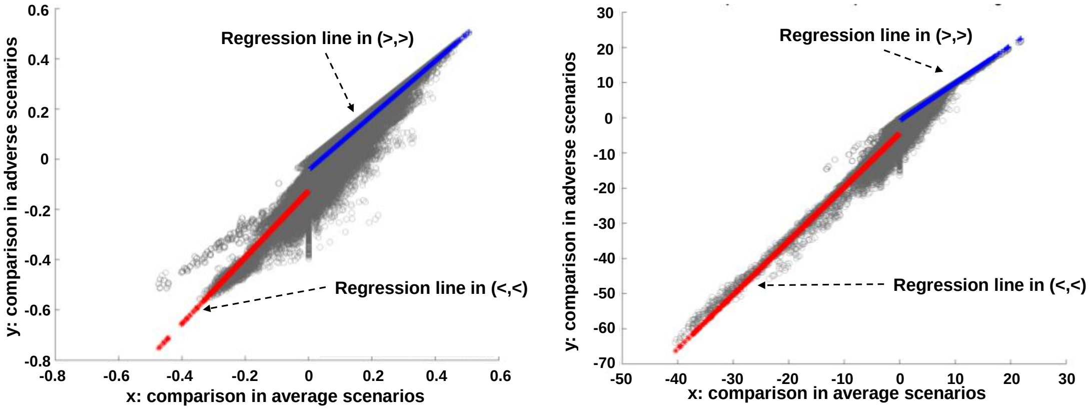
The proportions of cases (i.e., Case 1, Case 2, and Case 3) in which the studied techniques are more effective than random ordering in the average scenarios are 78% for APFD and 79% for HMFD. Among these three cases, the proportion of Case 3 takes up to 23.1% for APFD and 11.4% for HMFD. This finding indicates that even when the studied TCP techniques have been measured to be more effective than random ordering in the average scenarios, there is still a noticeable chance of having a less effective result when applying the techniques. Measuring the effectiveness of a technique in the average scenarios only is unlikely to reliably guarantee the effectiveness of applying the technique with high confidence (e.g., 80–100% of chances). However, it does provide a moderate confidence (e.g., 40–79% of chances).

To further explore the relationship between the effectiveness in the average scenarios and that in the adverse scenarios, we plotted all the comparison data points in a scatter plot as shown in Figure 1, in which each data point is a comparison result (*a*, *b*) summarized in Table 5. The *x*-axis and *y*-axis of the figure represent the effectiveness comparison results in the average scenarios and in the adverse scenarios, respectively.

We observe from Figure 1 that no matter for APFD or HMFD, the effectiveness of TCP techniques in the average scenarios seem to have a strong linear correlation with the effectiveness in the adverse scenarios. To verify our observation, we performed a Pearson correlation coefficient test using Matlab on all the data points. The test results in terms of APFD and HMFD are shown in the first column entitled 'All' under the section entitled *Pearson's Correlation Coefficient* in Table 6 and Table 7, respectively. There is no golden criterion to interpret the result of Pearson test in software engineering and services computing research. Thus, we followed existing work (e.g., Wang et al. (2014)) to interpret the result as follows: If the absolute value of the Pearson's Correlation Coefficient is greater than 0.8, the correlation is regarded as strong. If the absolute value is more than 0.1 but less than 0.5, the correlation is considered as mild. If the absolute value is at most 0.1, there is no correlation. Otherwise, the correlation is said to be moderate.

We find that for all the subjects except B (i.e., buybook), the overall correlation coefficient is larger than 0.9 for both APFD and HMFD. In fact, for the subject B, the overall correlation is more than 0.79. These Pearson test data indicate that overall speaking, there is a strong linear correlation between the effectiveness measures of the prioritization techniques in the two scenarios.

We also investigate whether the techniques that are effective (or ineffective, respectively) in the average scenarios are also effective (or ineffective) in their corresponding the adverse scenarios. Hence, we studied the correlations in Case (>, >) and Case (<, <).



(a) APFD

(b) HMFD

Figure 1. Statistical comparisons of TCP techniques for all subjects in average scenarios and in adverse scenarios

Table 6. Correlations of effectiveness measures of all TCP techniques in average scenarios and adverse scenarios for each subject in terms of APFD

Ref	**Pearson's Correlation Coefficient**			**Linear Regression (*y* = *ax* + *b*)**						
				Value of *a*				**Value of *b***		
	All	Case (>, >)	Case (<, <)	All	Case (>, >)	Case (<, <)	a(>, >) – a(<, <)	All	Case (>, >)	Case (<, <)
A	0.900375	0.892824	0.620642	1.238222	0.910829	1.822192	–0.91136	–0.05819	–0.02327	–0.05193
B	0.795791	0.856073	0.730196	1.152342	0.987222	0.653001	0.334221	–0.16123	–0.07582	–0.18668
C	0.976206	0.937613	0.970091	1.497027	1.051372	1.438022	–0.38665	–0.11067	–0.02299	–0.11902
D	0.935073	0.92394	0.808715	1.446076	0.969249	1.451735	–0.48249	–0.09624	–0.02538	–0.13787
E	0.975306	0.95563	0.903943	1.205387	1.147328	1.187479	–0.04015	–0.06119	–0.05346	–0.06169
F	0.954217	0.925365	0.553224	1.414656	1.096258	1.435644	–0.33939	–0.12958	–0.05628	–0.11976
G	0.96767	0.948118	0.732413	1.328643	1.162229	1.155755	0.006474	–0.10218	–0.06103	–0.12465
H	0.991916	0.971094	0.92489	1.369702	1.132112	1.0309	0.101212	–0.11834	–0.04655	–0.11544
All	0.952259	0.961681	0.905938	1.389617	1.084976	1.322813	–0.23784	–0.10858	–0.04226	–0.12571

Table 7. Correlations of effectiveness measures of all TCP techniques in average scenarios and adverse scenarios for each subject in terms of HMFD

Ref	**Pearson Correlation Coefficient**			**Linear Regression ($y = ax + b$)**						
				Value of *a*				**Value of *b***		
	All	Case (>, >)	Case (<, <)	All	Case (>, >)	Case (<, <)	a(>, >) – a(<, <)	All	Case (>, >)	Case (<, <)
A	0.92987	0.893379	0.560859	1.546947	1.276856	0.644731	0.632125	–2.18133	–1.31753	–1.96731
B	0.792614	0.747389	0.646295	1.45697	0.873583	0.835877	0.037706	–4.97821	–1.12333	–5.32191
C	0.989152	0.990626	0.990136	1.575833	1.087219	1.524913	–0.43769	–3.22828	–0.6807	–4.13576
D	0.958961	0.858727	0.705028	1.868274	1.153052	1.288721	–0.13567	–5.42358	–1.87845	–8.53117
E	0.987114	0.96721	0.865683	1.174191	1.165036	1.047905	0.117131	–0.98564	–0.94917	–1.12298
F	0.938369	0.981216	0.484756	1.532698	1.063368	1.519201	–0.45583	–3.62574	–0.68343	–2.8275
G	0.960469	0.991917	0.87947	1.367318	1.028599	2.128536	–1.09994	–3.54536	–0.45145	–5.27834
H	0.969589	0.995733	0.763302	1.667766	1.014701	1.214153	–0.19945	–6.41994	–0.30329	–6.45247
All	0.956734	0.981556	0.961	1.511497	1.065811	1.531579	–0.46577	–3.80313	–0.80661	–4.44408

Table 6 and Table 7 also show the Pearson test data for the two cases in terms of APFD and HMFD, respectively. We observe from Table 6 that the correlation coefficient in Case (>, >) is consistently higher than that in Case (<, <) for all the subjects except for subject C. In Table 7, this trend is consistent for all the subjects. In particular, in Case (>, >), all the subjects in these two tables except subject B in Table 7 have a correlation value larger than 0.85. In Case (<, <), only four and three subjects in these two tables have a correlation value larger than the threshold of strong correlations (0.80).

These observations indicate that for those test suites where the techniques are effective in the average scenarios, their effectiveness in the adverse scenarios can be more reliably predicted. However, for those test suites where the techniques are ineffective in the average scenarios, their effectiveness in the adverse scenarios is less reliably predictable.

We also computed the linear regression lines ($y = ax + b$) for all the data points (All), the data points in Case (>, >), and those in Case (<, <) for taking all subjects as a whole as well as for each individual subject.

Figure 1 shows the two linear regression lines for taking all subjects as a whole in Case (>, >) and Case (<, <), respectively. We observe that the slopes of the two regression lines are not the same (nor nearly so). Figure 2 and Figure 3 show the scatter plots and the regression lines for the data points of each subject in terms of APFD and HMFD, respectively. We also find similar observations that the slopes of the two regression lines are not the same (nor nearly so).

Table 6 and Table 7 show the estimated parameters (a and b) for the regression lines of all data points, the data points in Case (>, >), and the data points in Case (<, <) for each individual subject and taking all subjects as a whole (i.e., the last row). When taking all the subjects as a whole, we observe from the last row of each of Table 6 and Table 7 that the three values of 'a' are larger than 1, which means the effectiveness in the average scenarios is amplified in the adverse scenarios. The value of 'a' in Case (<, <) is larger than that in Case (>, >), which indicates that when techniques are ineffective in the average scenarios for some test suites, these techniques will amplify the ineffectiveness cases more in the adverse scenarios for the same test suites.

We then subtract the value of a in Case (<, <) from the value of a in Case (>, >) to facilitate our observation. The results are shown in Table 6 and Table 7 under the column entitled 'a(>, >) – a(<, <)'. We observed that most subjects (namely, 5 out of 8) had negative values in this column. This indicates that for most subjects, the above amplification effect *does* exist.

We also observe from Table 6 and Table 7 that the estimated values of *b* are negative for all subjects. This means that when techniques show no significant difference in effectiveness in the average scenarios of the test suites, the techniques are very likely to be ineffective in the adverse scenarios of these test suites as well.

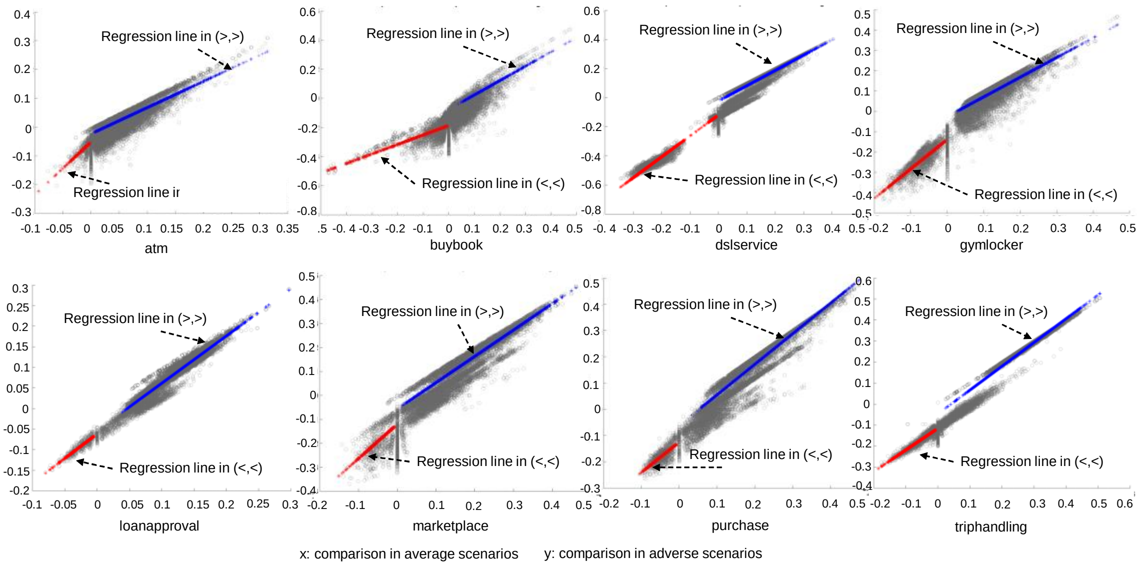


Figure 2 Statistical comparisons of TCP techniques in average scenarios and adverse scenarios in terms of APFD

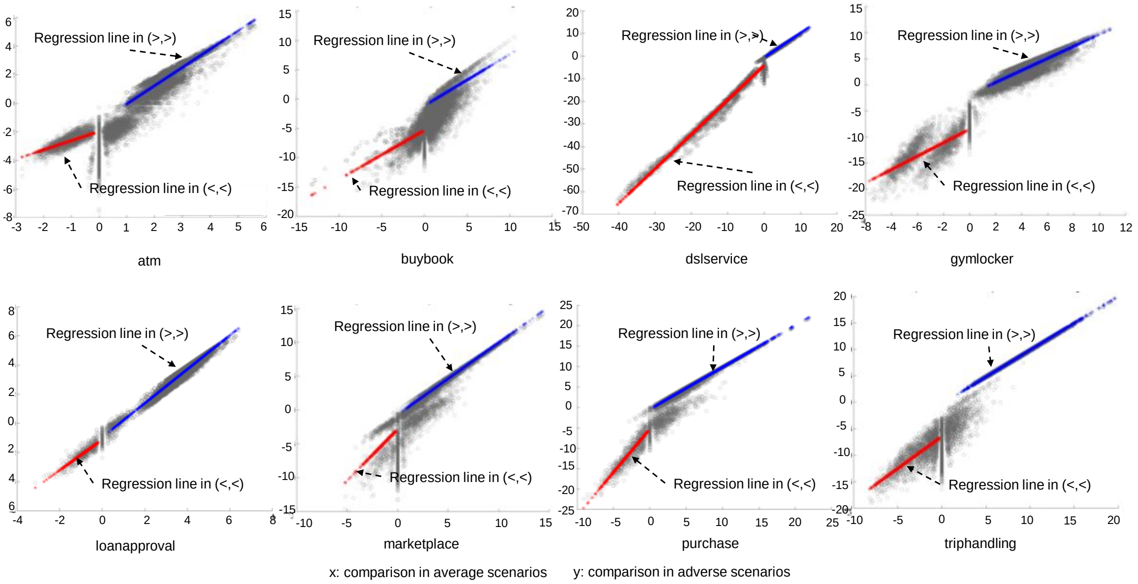


Figure 3 Statistical comparisons of TCP techniques in average scenarios and adverse scenarios in terms of HMFD

Summary: We find a strong linear correlation between the effectiveness of the techniques in the average scenarios and that in the adverse scenarios. This linear correlation is stronger for the test suites in Case (>, >) than those in Case (<, <). Also, comparing between the amplification results of the effectiveness measures in the average scenarios and the adverse scenarios, the ineffectiveness cases are amplified more in the adverse scenarios. The studied prioritization techniques also have a noticeable chance of being less effective than random ordering in terms of APFD and HMFD.

Answering RQ2

In RQ2, we investigate to what extent each of the five factors affects the prioritization effectiveness in different regions. We recall that Table 4 has shown all the technique-pairs for each factor at each pairwise combination of the corresponding factor levels, and the whole effectiveness spectrum of each technique on the prioritized test suite is also mapped to five regions defined at the end of the Experimental Procedure section.

As stated in the experimental procedure, in each region, for each factor at each pairwise combination of factor levels, we performed a multiple mean comparison for all the techniques. We then checked whether any pair of techniques is significantly different in each case, which is described in detail as follows.

Table 8 shows the statistical comparison results using the APFD dataset. In each region, there are three possible outcomes: M*i* is significantly more effective than (>), has no significant difference from (=), or is significantly less effective than (<) M*j*. The value in each cell in Table 8 is the proportion of technique-pairs M*i*–M*j* of the corresponding row in Table 4 falling within a particular outcome. For instance, from Table 4, there are 16 pairs of techniques grouped in the BPEL-XRG row. Under Region All in Table 8, 50% of them are marked as ">". It means that 50% of these techniques are identified to be more effective using BPEL than XRG. Other cells can be interpreted similarly.

Table 8. Statistics of technique pair comparisons in terms of APFD for each level value pair of each factor

Factor	Pairwise Comparison	Region 1 (Least Effective)			Region 2			Region 3			Region 4			Region 5 (Most Effective)			Region All		
		>	=	<	>	=	<	>	=	<	>	=	<	>	=	<	>	=	<
Artifact type	BPEL-XRG	0.44	0.00	**0.56**	**0.69**	0.06	0.25	**0.69**	0.00	0.31	0.25	0.00	**0.75**	0.38	0.06	**0.56**	0.50	0.06	0.44
	BPEL-WSDL	0.00	0.00	**1.00**	0.38	0.00	**0.63**	0.88	0.00	0.13	0.75	0.00	0.25	0.75	0.00	0.25	1.00	0.00	0.00
	XRG-WSDL	0.00	0.00	**1.00**	0.25	0.00	**0.75**	0.75	0.00	0.25	0.88	0.00	0.13	0.88	0.00	0.13	1.00	0.00	0.00
Strategy	Total-Additional	**0.61**	0.02	0.37	**0.47**	0.00	**0.53**	0.41	0.00	0.59	**0.53**	0.00	**0.47**	0.29	0.02	0.69	0.29	0.00	0.71
	Total-Iterative	0.14	0.00	**0.86**	0.07	0.00	**0.93**	0.61	0.00	0.39	0.89	0.00	0.11	0.71	0.00	0.29	0.89	0.00	0.11
	Additional-Iterative	0.07	0.00	**0.93**	0.14	0.04	**0.82**	0.86	0.00	0.14	0.89	0.00	0.11	0.79	0.00	0.21	0.96	0.00	0.04
Order direction	Ascending-Descending	**0.84**	0.03	0.13	**0.94**	0.00	0.06	0.28	0.00	0.72	0.00	0.00	1.00	0.00	0.00	1.00	0.03	0.00	0.97
Coverage data	Executable-Nonexecutable	0.19	0.00	**0.81**	0.19	0.03	**0.78**	0.66	0.00	0.34	0.81	0.00	0.19	0.69	0.00	0.31	0.88	0.00	0.13
ROLE hierarchy	CM1–CM2	**1.00**	0.00	0.00	**1.00**	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00
	CM1–CM3	**1.00**	0.00	0.00	**1.00**	0.00	0.00	**0.50**	0.00	**0.50**	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00
	CM1–CM4	**1.00**	0.00	0.00	**1.00**	0.00	0.00	**0.50**	0.00	**0.50**	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00
	CM1–CM5	**0.50**	0.00	**0.50**	**1.00**	0.00	0.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00
	CM2–CM3	**0.50**	**0.50**	0.00	**1.00**	0.00	0.00	0.00	**0.50**	**0.50**	**0.50**	0.00	**0.50**	0.00	0.00	1.00	0.00	0.00	1.00
	CM2–CM4	**0.50**	0.00	**0.50**	**1.00**	0.00	0.00	**0.50**	0.00	**0.50**	**0.50**	0.00	**0.50**	0.00	0.00	1.00	0.00	0.00	1.00
	CM2–CM5	**0.50**	0.00	**0.50**	**1.00**	0.00	0.00	0.00	0.00	1.00	**0.50**	0.00	**0.50**	**0.50**	0.00	**0.50**	0.00	0.00	1.00
	CM3–CM4	0.00	0.50	0.50	0.00	0.50	0.50	**0.50**	**0.50**	0.00	0.00	0.50	0.50	**0.50**	**0.50**	0.00	0.00	0.50	0.50
	CM3–CM5	0.00	**0.50**	**0.50**	0.00	**0.50**	**0.50**	0.00	**0.50**	**0.50**	0.50	0.50	0.00	0.50	0.50	0.00	0.50	0.50	0.00
	CM4–CM5	0.00	**0.50**	**0.50**	0.00	**0.50**	**0.50**	0.00	**0.50**	**0.50**	0.50	0.50	0.00	0.50	0.50	0.00	0.50	0.50	0.00

For ease of presentation, we further highlight certain cells in color. Specifically, in each region, for each row in Table 8, if one of the three cells ">", "=", and "<" dominates the percentage of techniques, we highlight the dominating cell by setting its background color to ***yellow***[1]. For

[1] The yellow cells are rendered in light gray if this paper is printed in black and white.

instance, in Region All for the BEPL-WSDL artifact type, the cell “>” dominates. On the other hand, if multiple cells have the same or similar values in the same region (defined as the difference in proportion being less than 10%), we highlight these multiple cells by setting their background color to ***green***[2]. As such, for each region, at each row, at least one cell is highlighted.

Moreover, for any entry in any specific region, if *either* (a) a yellow cell is under a column different from the yellow cell in Region All, *or* (b) there is a yellow cell but the corresponding cell in Region All is green, it indicates that the effect of prioritization in the former region is different from the overall effect observed from Region All. In this case, we further highlight the text in the yellow cell of the former region in **bold**. For instance, in the BPEL-WSDL row, the yellow cell in Region 1 is under the column “<” whereas the yellow cell in Region All is under the column “>”. Hence, the text in the former cell is highlighted in bold. We also do the highlighting for the green cells in a similar way.

We first discuss the overall results. In Region 1, the texts in all but one row (CM3–CM4) are highlighted in bold. On the other hand, in Region 5, the texts in much fewer rows (only three) are highlighted in bold.

In fact, moving from Region 1 to Region 5, we find that the texts in fewer and fewer cells are highlighted in bold. The results show that the overall effectiveness results (Region All) are increasingly consistent toward the results of those regions having higher effectiveness. It indicates that the results of TCP techniques obtained in the average scenarios are more likely to be extrapolated successfully to the more favorable end of the effectiveness spectrum.

We also study to what extent each factor is sensitive to affecting the prioritization effectiveness when switching from one factor level to another in different regions. The factor of ROLE strategy has the highest proportion (0.62 = 31 / 50) of regions that have at least one cell highlighted in bold, followed by the factor of artifact type (0.6 = 9 / 15) and strategy (0.47 = 7 / 15). The factors of order direction and coverage data have the same smallest proportion (0.4 = 2 / 5). This proportion can act as a measure of how sensitive a factor affects the prioritization effectiveness in the whole spectrum. Thus, the above data show that the most sensitive factor is the ROLE strategy, followed by the factors of artifact type and prioritization strategy. The factors of order direction and coverage type are the most insensitive.

We next discuss the results on individual factors. We examine the four pairs of regions in each row: (Region 1, Region 2), (Region 2, Region 3), (Region 3, Region 4), and (Region 4, Region 5). Specifically, if the change in relative effectiveness between two levels of the same factor (in the same row) is significant enough, then the highlighted cells in the corresponding pair of regions will change from one pattern (such as a yellow cell under the “>” column of one region in the pair) to another pattern (such as a yellow cell under the “=” column or a pair of green cells in another region of the same pair).

We observe that for each row, there is at least one pair of regions such that the highlighted cells are placed under different columns. It indicates that for each factor, a change of level can lead to a difference in effectiveness in a statistically significant way in the corresponding region.

For artifact type and strategy as a factor, among the three levels, there are, respectively, 4 and 6 out of 12 possible consecutive pairs of regions such that a change of level in a given level pair (indicated by a row) can result in moving the highlighted cells from one set of columns (either

[2] The green cells are rendered in dark gray if this paper is printed in black and white.

one or two columns depending on the color code) to another set of columns. Similarly, one out of four pairs of consecutive regions in either order direction or coverage data can result in such a change. For the CM*i* series, we find that 22 out of 40 possible consecutive pairs can result in such changes.

Hence, roughly speaking, in 50% of the cases, a change of factor level in these five factors can result in a significant change of the TCP effectiveness.

Lastly, we find it quite interesting that the two ends of the effectiveness spectrum of each technique exhibit such a distinguishing behavior from the factor's perspective. Specifically, for artifact type as a factor, we find that using WSDL is attractive in Regions 1 and 2 (the ineffective end), but toward Region 5 (the effective end), it loses its grounds to using XRG. For strategy as a factor, the Iterative strategy is attractive when used in Region 1, but using the Additional strategy is better in Region 5. For order direction as a factor, we find that using ascending ordering is effective in Region 1, but using descending ordering is better in Region 5. For the factor of coverage data, using non-executable coverage artifact is better in Region 1, but using executable coverage artifact is better in Region 5. For the CM*i* techniques, in Region 1, CM1 is more effective, but in Region 5, CM3 becomes the most effective.

Table 9. Statistics of technique pair comparisons in terms of HMFD for each level value pair of each factor

Factors	**Pairwise Comparison**	**Region 1 (Least Effective)**			**Region 2**			**Region 3**			**Region 4**			**Region 5 (Most Effective)**			**Region All**		
		>	=	<	>	=	<	>	=	<	>	=	<	>	=	<	>	=	<
Artifact type	BPEL-XRG	1.00	0.00	0.00	0.94	0.00	0.06	0.63	0.00	0.38	0.31	0.00	**0.69**	–	–	–	0.75	0.00	0.25
	BPEL-WSDL	1.00	0.00	0.00	1.00	0.00	0.00	0.63	0.13	0.25	0.50	0.00	**0.50**	–	–	–	1.00	0.00	0.00
	XRG-WSDL	0.75	0.00	0.25	**0.50**	0.00	**0.50**	0.63	0.00	0.38	0.63	0.00	0.38	–	–	–	0.75	0.00	0.25
Strategy	Total-Additional	0.22	0.00	0.78	0.20	0.06	0.73	0.02	0.00	0.98	**0.57**	0.00	0.43	–	–	–	0.31	0.00	0.69
	Total-Iterative	0.71	0.00	0.29	0.75	0.00	0.25	0.36	0.00	**0.64**	0.86	0.00	0.14	–	–	–	0.79	0.00	0.21
	Additional-Iterative	0.93	0.00	0.07	0.86	0.00	0.14	0.75	0.04	0.21	0.71	0.00	0.29	–	–	–	0.96	0.00	0.04
Order direction	Ascending-Descending	0.19	0.06	0.75	0.25	0.00	0.75	**0.75**	0.03	0.22	0.00	0.00	1.00	–	–	–	0.03	0.00	0.97
Coverage data	Executable-Nonexecutable	0.69	0.00	0.31	0.66	0.00	0.34	0.69	0.03	0.28	0.63	0.00	0.38	–	–	–	0.78	0.00	0.22
ROLE strategy	CM1–CM2	0.50	0.00	0.50	0.50	0.00	0.50	0.50	0.00	0.50	0.00	0.00	**1.00**	–	–	–	0.50	0.00	0.50
	CM1–CM3	0.00	0.00	1.00	0.00	0.00	1.00	**1.00**	0.00	0.00	0.00	0.00	1.00	–	–	–	0.00	0.00	1.00
	CM1–CM4	**0.50**	0.00	**0.50**	0.00	0.00	1.00	**1.00**	0.00	0.00	0.00	0.00	1.00	–	–	–	0.00	0.00	1.00
	CM1–CM5	**0.50**	0.00	**0.50**	**0.50**	0.00	**0.50**	**1.00**	0.00	0.00	0.00	0.00	1.00	–	–	–	0.00	0.00	1.00
	CM2–CM3	**0.50**	0.00	**0.50**	0.00	**0.50**	**0.50**	**1.00**	0.00	0.00	0.00	0.00	1.00	–	–	–	0.00	0.00	1.00
	CM2–CM4	**0.50**	0.00	**0.50**	**0.50**	0.00	**0.50**	**1.00**	0.00	0.00	0.00	0.00	1.00	–	–	–	0.00	0.00	1.00
	CM2–CM5	0.50	0.00	0.50	0.50	0.00	0.50	**1.00**	0.00	0.00	0.00	0.00	**1.00**	–	–	–	0.50	0.00	0.50
	CM3–CM4	0.50	0.50	0.00	0.50	0.50	0.00	0.50	0.50	0.00	0.00	**0.50**	**0.50**	–	–	–	0.50	0.50	0.00
	CM3–CM5	0.50	0.50	0.00	0.50	0.50	0.00	0.50	0.50	0.00	0.50	0.50	0.00	–	–	–	0.50	0.50	0.00
	CM4–CM5	0.50	0.50	0.00	0.50	0.50	0.00	0.00	**0.50**	**0.50**	0.50	0.50	0.00	–	–	–	0.50	0.50	0.00

Table 9 shows the comparison data in terms of HMFD, which can be interpreted similarly as the APFD data in Table 8. No technique has any data falling in Region 5. Hence, we mark the cells in Region 5 with the symbol '–'.

As to the sensitivity of factors affecting the effectiveness spectrum, similar to the observation in Table 8, we find that the factor of ROLE strategy is still the most sensitive one, where 42.5% (= 17 / 40) of the regions have at least one cell highlighted in bold. The next one is the factor of order direction (33% = 1 / 4), followed by artifact type (25% = 3 / 12), and then strategy (0.17 = 2 / 12). The factor of coverage data is still the most insensitive one, which has no cell highlighted in bold in each region for the HMFD metric. This is consistent with the conclusion drawn from the APFD dataset.

We also find that for each factor, a change of level can result in many significant changes in the highlighted cells between two consecutive regions. For instance, 4, 3, 2, 0, and 19 changes out of 9, 9, 3, 3, and 30 possible pairs have their highlighted cells moved, or about 52% of all cases (or 42% in weighted average). This also supports the conclusion drawn from the APFD dataset that in a noticeable proportion of cases, the TCP effectiveness changes significantly as we change the factor level.

For individual factors, most level-switch pairs have at least one pair of regions between Regions 1 and 4 such that the highlighted cells are under different columns. This also indicates that the two ends of the effectiveness spectrum of each technique in terms of HMFD also exhibit distinguishable behavior from the factor's perspective.

Summary: The five factors have significant effects on prioritization effectiveness with respect to changes in factor levels. The ROLE strategy and coverage data are still the most sensitive and insensitive factor, respectively, in affecting the prioritization effectiveness spectrum.

Threats to Validity

In this section, we discuss the threats to validity of the experiments.

Threats to construct validity arise when the metric cannot adequately reflect the features that they should measure. In our experiments, we used two metrics APFD and HMFD to measure the effectiveness of test case prioritization techniques. Other measures such as FATE (Yu and Lau 2012) and APSC (Li et al. 2007) may also be used to evaluate a prioritization technique. We used mutants to simulate real faults in this study. Existing work (Andrews et al. 2005) shows that the detection of mutation faults can simulate the detection of real faults in the same program. Many studies have used these mutation faults to evaluate TCP techniques.

Threats to internal validity affect the ability to make causal conclusions between experiment variables. The major internal threat is the correctness of data collection for each test case prioritization technique and data analyses for answering research questions. To assure their correctness, we have carefully implemented our tools and test suite prioritization techniques in Java. We also have carefully verified the Matlab scripts for data analyses.

Threats to external validity relate to the degree of extending our results and conclusions to more general populations, including subjects and techniques. To minimize any bias in our experiments, we have chosen subjects and techniques widely researched and used in existing work. Even so, the use of other subjects, test cases, faults, test oracles, and techniques may yield different results. We define the effectiveness spectrum into five regions with the results of a random ordering technique. The use of alternative decompositions of the effectiveness spectrum may also yield other results.

RELATED WORK

This section discusses existing work related to this paper.

Regression testing is widely used in the industry (Onoma et al. 1998). It is a testing process performed after the modification of a program (Leung and White 1989). Leung and White pointed out that it is not a simple testing process by just rerunning all the test cases. Regression testing can be more effective by selecting only those test cases relevant to the modified components. Test case prioritization is one of major tasks in regression testing, enabling test cases to be executed in selected order to achieve specific testing purposes, such as a higher fault detection rate (Do et al. 2004; Elbaum et al. 2002; Srivastava and Thiagarajan 2002).

Leung and White (1989) provided a principle of retests by dividing the regression testing problem into two subproblems: *test selection* and *test plan update*. Yoo and Harman (2012) reported that there are an increasing number of papers that study regression testing techniques.

Generally, there are two kinds of test case prioritization, namely *general* test case prioritization and *version-specific* test case prioritization (Elbaum et al. 2002). For the former, a test suite T for a program P is sorted with the intent of being useful over the subsequent modified versions of P. For the latter, the test suite T is prioritized to be useful on a specific version P' of P. Such a test suite may be more effective at meeting the goal of the prioritization for P'. Our study in this paper focuses on the former kind.

Many coverage-based prioritization techniques (such as Do et al. 2004; Elbaum et al. 2002; Kim and Porter 2002; Mei et al. 2011; Mei et al. 2014b; Rothermel and Harrold 1996; Rothermel et al. 2001; Rothermel et al. 2002; Srivastava and Thiagarajan 2002) have been proposed, including prioritizing test cases by the total statement or branch coverage achieved by individual test cases, and by additional statement or branch coverage (or additional cost) achieved by not-yet-selected test cases. Zhang et al. (2013) generalized the total-and-additional test case prioritization strategies. Some techniques are not purely based on code coverage data of test cases such as prioritization based on test costs (Srivastava and Thiagarajan 2002), fault severities (Yoo and Harman 2012), ability to detect specification-based faults (Yu and Lau 2012), data from the test history (Huang et al. 2012), or fault-exposing-potential (Yoo and Harman 2012). The effects of granularity and compositions of test suites have been reported (Elbaum et al. 2002). Srivastava and Thiagarajan (2002) built an *Echelon* system to prioritize test cases according to the potential change impacts of individual test cases between versions of a program to cover maximally the affected programs. Most of the existing experiments are conducted on procedural (Elbaum et al. 2002) and object-oriented (Do et al. 2004) programs.

In addition, studies on prioritizing test cases using input domain information (Zhai et al. 2014) and service discovery mechanisms (Tsai et al. 2005) have been explored. Moreover, methods to reveal internal state transitions through testing techniques have also been developed (Bartolini et al. 2011; Ye and Jacobsen 2013).

Xu and Rountev (2007) proposed a regression test selection technique for AspectJ programs. They use a control-flow representation for AspectJ software to capture aspect-related interactions and develop a graph comparison algorithm to select test cases. Martin et al. (2007) gave a framework that generates and executes web-service requests, and collects the corresponding responses from web services. Using such request-response pairs, they test the robustness aspect of services. They discuss the potential of using request-response pairs for regression testing. Tsai et al. (2005) proposed an adaptive group testing technique to address the challenges in testing a service-oriented application with a large number of web services simultaneously.

Using the mathematical definitions of XPath constructs (XPath 2.0, 2007) as rewriting rules, Mei et al. (2008) developed a data structure known as XPath Rewriting Graph (*XRG*). They propose an algorithm to construct XRGs and a family of unit testing criteria to test WS-BPEL applications. Their research group has also developed test case prioritization techniques for service testing (Mei et al. 2011; Mei et al. 2014b). However, they do not study the factors that may affect the fault detection effectiveness in the adverse scenarios or the whole effectiveness spectrum. This paper complements the study of TCP technique performance on the whole spectrum.

Most importantly, to the best of our knowledge, all the above reviewed work have not studied the relationship between the average scenarios that they study and other scenarios as what we have presented in RQ1 (connecting to the adverse scenarios) and in RQ2 (connecting among five sequences of regions in the whole effectiveness spectrum with respect to random ordering).

CONCLUSION

In this paper, we have analyzed test case prioritization for WS-BPEL applications in both the average scenarios and adverse scenarios. We have found a strong linear correlation between the effectiveness in the average and adverse scenarios. We have also studied the influence of using different levels of the same design factors on prioritization effectiveness. We have found that switching the levels of various different design factors of various prioritization techniques could significantly affect the prioritization effectiveness in at least 50% and 52% of all cases in terms of APFD and HMFD, respectively.

ACKNOWLEDGMENT

This research is supported in part by the Early Career Scheme and the General Research Fund of the Research Grants Council of Hong Kong (project numbers 111313, 125113, 123512, 716612, and 717811). W.K. Chan is the corresponding author. A preliminary version of this paper was published as a Research Track paper in the *Proceedings of the 2014 IEEE International Conference on Web Services* (*ICWS '14*) (Jia et al. 2014).

REFERENCES

Andrews, J.H., Briand, L.C., & Labiche, Y. (2005). Is mutation an appropriate tool for testing experiments? *Proceedings of the 27th International Conference on Software Engineering* (*ICSE '05*), St. Louis, MO, USA, 402–411.

Bartolini, C., Bertolino, A., Elbaum, S.G., & Marchetti, E. (2011). Bringing white-box testing to service oriented architectures through a service oriented approach. *Journal of Systems and Software*, *84*(4), 655–668.

Do, H., Rothermel, G., & Kinneer, A. (2004). Empirical studies of test case prioritization in a JUnit testing environment, *Proceedings of the 15th International Symposium on Software Reliability Engineering* (*ISSRE '04*), Saint-Malo, Bretagne, France, 113–124.

Elbaum, S.G., Malishevsky, A.G., & Rothermel, G. (2002). Test case prioritization: A family of empirical studies. *IEEE Transactions on Software Engineering*, *28*(2), 159–182.

Huang, Y.-C., Peng, K.-L., & Huang, C.-Y. (2012). A history-based cost-cognizant test case prioritization technique in regression testing. *Journal of Systems and Software*, *85*(3), 626–637.

Jia, C., Mei, L., Chan, W.K., Yu, Y.T., & Tse, T.H. (2014). Is XML-based test case prioritization for validating WS-BPEL evolution effective in both average and adverse scenarios? *Proceedings of the IEEE International Conference on Web Services* (*ICWS '14*), Anchorage, AK, USA, 233–240.

Kim, J.-M. & Porter, A. (2002). A history-based test prioritization technique for regression testing in resource constrained environments, *Proceedings of the 24th International Conference on Software Engineering* (*ICSE '02*), Orlando, FL, USA, 119–129.

Leung, H.K.N. & White, L.J. (1989). Insights into regression testing, *Proceedings of the IEEE International Conference on Software Maintenance* (*ICSM '89*), Miami, FL, USA, 60–69.

Li, Z., Harman, M. & Hierons, R.M. (2007). Search algorithms for regression test case prioritization. *IEEE Transactions on Software Engineering*, *33*(4), 225–237.

Martin, E., Basu, S., & Xie, T. (2007). Automated testing and response analysis of Web services, *Proceedings of the IEEE International Conference on Web Services* (*ICWS '07*), Salt Lake City, UT, USA, 647–654.

Mei, L., Cai, Y., Jia, C., Jiang, B., Chan, W.K., Zhang, Z., & Tse, T.H. (2014a). A subsumption hierarchy of test case prioritization for composite services. *IEEE Transactions on Services Computing*, doi: 10.1109/TSC.2014.2331683.

Mei, L., Chan, W.K., & Tse, T.H. (2008). Data flow testing of service-oriented workflow applications, *Proceedings of the 30th International Conference on Software Engineering* (*ICSE '08*), Leipzig, Germany, 371–380.

Mei, L., Chan, W.K., Tse, T.H., Jiang, B., & Zhai, K. (2014b). Preemptive regression testing of workflow-based Web services. *IEEE Transactions on Services Computing*, doi: 10.1109/TSC.2014.2322621.

Mei, L., Chan, W.K., Tse, T.H., & Merkel, R.G. (2011). XML-manipulating test case prioritization for XML-manipulating services. *Journal of Systems and Software*, *84*(4), 603–619.

Onoma, A.K., Tsai, W.-T., Poonawala, M., & Suganuma, H. (1998). Regression testing in an industrial environment. *Communications of the ACM*, *41*(5), 81–86.

Rothermel, G., Elbaum, S.G., Malishevsky, A.G., Kallakuri, P., & Davia, B. (2002). The impact of test suite granularity on the cost-effectiveness of regression testing, *Proceedings of the 24th International Conference on Software Engineering* (*ICSE '02*), Orlando, FL, USA, 130–140.

Rothermel, G. & Harrold, M.J. (1996). Analyzing regression test selection techniques. *IEEE Transactions on Software Engineering*, *22*(8), 529–551.

Rothermel, G., Untch, R.H., Chu, C., & Harrold, M.J. (2001). Prioritizing test cases for regression testing. *IEEE Transactions on Software Engineering*, *27*(10), 929–948.

Srivastava, A. & Thiagarajan, J. (2002). Effectively prioritizing tests in development environment, *Proceedings of the 2002 ACM SIGSOFT International Symposium on Software Testing and Analysis* (*ISSTA '02*), Rome, Italy, 97–106.

Tsai, W.-T., Chen, Y., Paul, R.A., Huang, H., Zhou, X., & Wei, X. (2005). Adaptive testing, oracle generation, and test case ranking for Web services, *Proceedings of the 29th Annual International Computer Software and Applications Conference* (*COMPSAC '05*), vol. 1, Edinburgh, UK, 101–106.

Wang, H., Chan, W.K., & Tse, T.H. (2014). Improving the effectiveness of testing pervasive software via context diversity. *ACM Transactions on Autonomous and Adaptive Systems*, *9*(2), 9:1–9:28.

Web Services Business Process Execution Language Version 2.0: OASIS Standard (2007). Organization for the Advancement of Structured Information Standards (OASIS). Retrieved from http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf.

Wong, W.E., Horgan, J.R., London, S., & Agrawal, H. (1997). A study of effective regression testing in practice, *Proceedings of the 8th International Symposium on Software Reliability Engineering* (*ISSRE '97*), Albuquerque, NM, USA, 264–274.

XML Path Language (*XPath*) *2.0: W3C Recommendation* (2007). W3C. Retrieved from http://www.w3.org/TR/xpath20/.

Xu, G. & Rountev, A. (2007). Regression test selection for AspectJ software, *Proceedings of the 29th International Conference on Software Engineering* (*ICSE '07*), Minneapolis, MN, USA, 65–74.

Ye, C. & Jacobsen, H.-A. (2013). Whitening SOA testing via event exposure. *IEEE Transactions on Software Engineering*, *39*(10), 1444–1465.

Yoo, S. & Harman, M. (2012). Regression testing minimization, selection and prioritization: A survey. *Software Testing, Verification and Reliability*, *22*(2), 67–120.

Yu, Y.T. & Lau, M.F. (2012). Fault-based test suite prioritization for specification-based testing. *Information and Software Technology*, *54*(2), 179–202.

Zhai, K., Jiang, B., & Chan, W.K. (2014). Prioritizing test cases for regression testing of location-based services: Metrics, techniques, and case study. *IEEE Transactions on Services Computing*, *7*(1), 54–67.

Zhang, L., Hao, D., Zhang, L., Rothermel, G., & Mei, H. (2013). Bridging the gap between the total and additional test-case prioritization strategies, *Proceedings of the 2013 International Conference on Software Engineering* (*ICSE '13*), San Francisco, CA, USA, 192–201.

Changjiang Jia is a PhD student at Department of Computer Science, City University of Hong Kong. He received the BEng and MEng degrees from National University of Defense Technology, China. His research interest is testing and analysis of concurrent and service-based software.

Lijun Mei received the PhD degree from The University of Hong Kong. He is a research staff member at IBM Research—China. His current research interests include program analysis and application management in the business environment.

W.K. Chan is an associate professor at Department of Computer Science, City University of Hong Kong. His current main research interest is program analysis and testing for concurrent software and systems. He is on the editorial board of the Journal of Systems and Software.

Yuen Tak Yu is an associate professor at Department of Computer Science, City University of Hong Kong. His research interests include software testing, e-commerce and computers in education. His publications have appeared in scholarly journals and leading international conferences, such as ACM Transactions on Software Engineering and Methodology, IEEE Transactions on Software Engineering, Information and Software Technology, Information Research, Computers and Education, ICSE, FSE, ISSRE, ICCE and others. He is a past chair of the IEEE Hong Kong Section Computer Society Chapter.

T.H. Tse received the PhD degree from the London School of Economics and was a visiting fellow at the University of Oxford. He is an honorary professor in computer science at The University of Hong Kong after retiring from the full professorship in July 2014. His research interest is in program testing, debugging, and analysis. He is the steering committee co-chair of QRS and an editorial board member of the Journal of Systems and Software, Software Testing, Verification and Reliability, and Software: Practice and Experience. He also served on the search committee for the editor-in-chief of the IEEE Transactions on Software Engineering in 2013. He is a fellow of the British Computer Society, a fellow of the Institute for the Management of Information Systems, a fellow of the Institute of Mathematics and Its Applications, and a fellow of the Hong Kong Institution of Engineers. He was awarded an MBE by The Queen of the United Kingdom.