# Paper Title: Dynamic Random Testing of Web Services: A Methodology and Evaluation

***Manuscript ID: TSC-2019-01-0031***

Dear Editor-in-Chief,

Thank you for your email on May 8, 2019, regarding our paper “Dynamic Random Testing of Web Services: A Methodology and Evaluation,” which was submitted to IEEE Transactions on Services Computing (Manuscript ID: TSC-2019-01-0031).

We are submitting a new version of the paper, in which we have made revisions to address and respond to each comment from the reviewers. Below are the detailed responses to the comments.

We look forward to hearing from you.

Yours sincerely,  
Chang-ai Sun, Hepeng Dai, Guan Wang, Dave Towey, Kai-Yuan Cai, and Tsong Yueh Chen

# Response to comments of associate editor and reviewers

In the following, unless otherwise specified, all comments refer to the revised version of the paper.

## Associate editor’s comments

***E1C1: Having analyzed in detail the reviews and the manuscript, I have matured my recommendation to the EIC, which is that it undergoes a major revision.***

Response: Thank you for your recommendation.

Action: None.

***E1C2: I believe the reviewers provided useful comments for you to improve the manuscript. Should you choose to revise your manuscript, pay attention to address all reviewers' concerns, especially those on: (1) the novelty of this paper with respect to own previous publication; (2) the improvements to the writing of various Sections; (3) the repeatability of experiments; (4) the definition of failure rate in the 2nd research question and the revision of the answer to the 3rd research question; (5) more details about the mutants generated; (6) insight as to why CP testing was able to uncover faults; (7) the quality of the reported graphics.***

Response: Thank you for summarizing the reviewers’ comments. We have carefully examined and responded to all these comments, and have made the corresponding revisions, accordingly. Please refer to our response to R3C2 for the first concern; responses to R1C2, R2C2, R2C3, R2C11, R2C12, R3C2, R3C4, R3C5, R3C6, and R3C7 for the second concern; response to R2C5 for the third concern; responses to R2C9 and R2C10 for the fourth concern; response to R1C5 for the fifth concern; response R1C5 for the sixth concern; and response to R2C11 for the seventh concern.

Action: None.

## Reviewer 1's comments

***R1C1:*** ***The paper is relevant to the services computing community. However, the specific representations for the SOA used in the paper (WSDL) etc sound a bit dated.***

Response: Thank you for the comment. Although web services may be considered somewhat dated, they are still adopted in the development of various applications. For instance, a representative web service repository (https://github.com/ouniali/WSantipatterns) contains 226 realistic web services in various domains. Furthermore, researchers from academia are still exploring the performance improvement of web services, and reporting their research results in top conferences and journals in this area. For example, some such work related to web services has been recently published in TSC, including: S. Wang, Y. Ma, B. Cheng, F. Yang, & R. N. Chang, *"Multi-Dimensional QoS Prediction for Service Recommendations", IEEE TSC, 2019, 12(1):47-57; and P. Wang & X. Du. "QoS-Aware Service Selection Using an Incentive Mechanism", IEEE TSC, 2019, 12(2):262-275.*

Action: None.

***R1C2:*** ***The probability distribution computation model described in section 3.2 is too detailed. Some of the proofs etc and moved into an appendix without taking away from the core message.***

Response: Thank you for the suggestion.

Action: In the revised manuscript, we have followed the suggestion and moved the proofs from Section 3.2 to the appendix.

***R1C3:*** ***The technique also requires a user to provide category partition details along with an initial test profile. This could be an unreasonable expectation from a test practitioner.***

Response: Thank you for the comment. When applying a testing technique, it is very common that certain inputs are expected to be provided by the tester. In the context of partition testing, the tester needs to construct the partitions based on the specification of the software under test. Furthermore, the partition details can be derived in various ways, including through use of the equivalent class method or the category partition method. Because our technique is based on [illegible] it is necessary that the partition details be provided (by the tester), which can easily be done through analysis of the input parameters and their constraints, as described in the specifications of the web service under test. Once the partition details are available, then it is not difficult to set an initial test profile: The tester can, for example, simply use a uniform probability distribution ($P_1$=$P_2$=…=$P_k$=$1/k$, where $k$ denotes the number of partitions, and $P_i$

($i$=$1..k$) denotes the probability of selecting the $i^{th}$ partition). Our technique, therefore, only requires the tester to provide additional information on the initial test profile, which should be relatively affordable and easy.

Action: In the revised manuscript, we have added a discussion (in the last paragraph of Section 3.1) of the inputs required from the tester when using our technique.

Intended details are as follows:

Generally speaking, DRT test case generation is influenced by both the probability distribution (for selection of the relevant partition), and the principles of RT — combining the effectiveness of PT with the ease of RT. Because our technique is based on PT, it is necessary that the partition details be provided (by the tester), which can easily be done through analysis of the input parameters and their constraints, as described in the specifications of the web service under test. Once the partition details are available, then it is not difficult to set an initial test profile: The tester can, for example, simply use a uniform probability distribution ($P_1$=$P_2$=…=$P_m$=$1/m$, where $m$ denotes the number of partitions, and $P_i$ ($i$=$1…m$) denotes the probability of selecting the $i^{th}$ partition). In Section 3.2, we provide some guidance for how to set the DRT parameters. Furthermore, many of the components in the DRT for web services framework can be automated. To enhance the efficiency and practical applicability of DRT for web services, we also developed a prototype that will be described in Section 3.3.

批注 **[DT1]:** Will a prototype enhance the efficiency and practical applicability?

***R1C4:*** ***The applications used in the study are fairly small (~100 SLOC). While the authors say it is not possible to gain access to service request implementations, they do seem to have access to the implementation of the web services used in the experiments in order to create the faulty mutants.***

Response: We do understand the reviewer’s concern regarding the size of the subject web services used in the study. Certainly, it would be preferable to include larger open-source services for evaluation. However, to the best of our knowledge, no such services are available for study. As noted, our evaluation needed access to the web services’ source code in order to be able to seed in the faults. However, due to commercial interests (including protection of technical secrets), it seems that most real web service owners are unwilling to make the source code available.

In order to overcome this lack of realistic open-source web services, we developed the subject applications ourselves, based on the real-life specifications. In this way, we were obviously able to access the source code for evaluation.

Action: In the revised manuscript, we have added some further explanation (in the first paragraph of Section 4.2) to clearly state that the subject web services were developed in our laboratory based on the real-life specifications.

***R1C5:*** ***Authors should provide more details about the actual mutants generated and provide insight as to why the specification based testing (CP testing) was able to uncover those faults.***

Response: Thank you for the suggestion. We agree that the suggested details would be helpful for understanding the experimental settings. Accordingly, we have included more details about the actual mutants generated, including the tool that was used to generate them, the mutation operators used, and how the evaluated mutants were finally selected.

We also agree that it is necessary to provide some insight into why the specification-based testing was able to uncover the faults. In our study, as described in Section 4.4.1, we employed a decision table (DT rather than CP) to construct the partitions for each studied web service. Because DT considers all parameters and identifies their invalid combinations, it can provide a systematic and efficient way to partition an input domain into disjoint subdomains, and then generate test cases. In practice, each DT rule condition entry corresponds to a partition in which test cases cover some paths — thus, the faults in those paths have a chance of being detected.

Action: In the revised manuscript, we have followed the suggestion to provide more details about the mutants generated (the first paragraph of Section 4.2), and provide insight as to why specification-based testing was able to reveal the faults (the fourth paragraph of Section 4.4.1).

Intended description of mutant generation as follows:

We used the tool muJava [31] to conduct mutation analysis [12], [28]–[30], generating a total of 1563 mutants. Each mutant was created by applying a syntactic change (using a mutation operator) to the original program, and all mutation operators provided by muJava were used in the experiments. Equivalent mutants, and those that were too easily detected (requiring less than 20 randomly generated test cases), were removed. To ensure the statistical reliability, we obtained 50 different test suites using different random seeds, then testing all mutants with all test suites, calculating the average number of test cases needed to kill (detect) a mutant.

批注 **[DT2]:** Is this corect? We used all the different operators?

Intended insights are follows:

As can be seen from the description above, because DT considers all parameters and identifies their invalid combinations, it can provide a systematic and efficient way to partition an input domain into disjoint subdomains, and then generate test cases. In practice, each DT rule condition entry corresponds to a partition in which test cases cover some paths — thus, the faults in those paths have a chance of being detected

## Reviewer 2's comments

***R2C1:*** ***The problem addressed in this manuscript is interesting. The proposed solution is appealing because of its simplicity and low applicability effort; moreover it improves significantly the performance of random testing and partition testing (commonly used in web services testing).***

Response: Thank you for the endorsement.

Action: None.

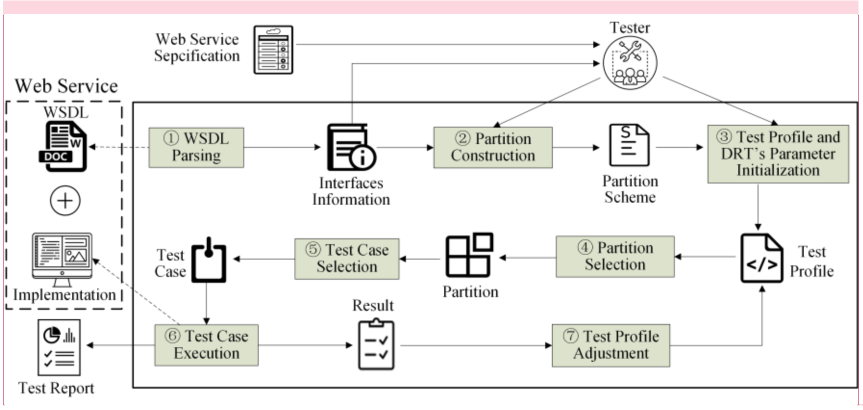
***R2C2:*** ***In the model in Figure 1, the human interaction is not represented, although it is important not only for DRT parameters configuration, but also for partitions construction as specified in Section 3.3 "The tool provides two options for the partitions and test suites: either to manually specify the partitions (and test cases); or to upload the predefined partitions and test suites".***

Response: We apologize for having omitted the human interactions from the figure in the original paper. Thank you for drawing our attention to this.

feel sorry that human interactions are missing in the framework in Figure 1. The main interactions are the partition scheme construction and the setting of the initial test profiles and DRT parameters.

Action: In the revised manuscript, we have followed the suggestion and added the human interactions to the figure (Figure 1). We have also revised the framework description in Section 3.1.

Intended details and updated framework are as follows:



批注 [DT3]: Web Service Specification

Test Profile and DRT Parameter Initialization: Testers need to initialize the test profile, a simple way of doing which would be to use a uniform probability distribution ($P_1=P_2=\ldots=P_k=1/k$, where k denotes the number of partitions, and $p_i$ (i=1…k) denotes the probability of selecting the ith partition). They also need to set the DRT parameters (guidelines for which are introduced in Section 3.2).

Test Case Selection: DRT selects a test case from the selected partition $s_i$ according to a uniform distribution.

***R2C3:*** ***The first contribution reported by authors in Section 3.1 is "a DRT framework that addresses key issues for testing web services, and a prototype that partly automates the framework". From the description in section 3.1, the focus is more on positive features than on issues, in fact the WSDL document allows to automate part of the testing process, that is a common practice in web services testing. The prototype description in Section 3.3 is very thin and too similar to the one in the conference paper. Further details are desirable.***

Response: Thank you for the comment. We agree that the original description was very thin, and very similar to the conference paper's version, which was mainly due to the page limitation. We have decided to follow the suggestion to add more details about the prototype, and have made substantial efforts to enhance the prototype reported in the conference paper. The enhancements include: (1) a guidance feature to guide usage of this prototype; (2) a configuration feature to set the DRT parameters, partitions, and test case generations; and (3) an execution feature to provide information about the WSUT execution process, and the testing results.

Action: In the revised manuscript, we have updated the description of the enhanced prototype (Section 3.3).

Intended details are as follows [这些界面设计太粗糙，无法使用!!!]:

(1) Guidance. This feature describes the steps the tester should follow when testing a web service.

(2) Configuration. This feature interacts with the testers to obtain and set the information related to testing the web service, including: the address of the web service under test; the DRT parameters and partition details; and the test case generation. The detailed steps are as follows:

- Inputting and parsing the URL: We integrate the WSDL parsing functionality provided by MT4WS (*C-A. Sun et al. MT4WS: an automated metamorphic testing system for web services, IJHPCN 9(1/2): 104-115, 2016*). This enables all the (WSDL) parameters and their types to be automatically obtained.
- Parameter setting: The tester is responsible for selecting which operations of the current WSUT are to be tested, and for partitioning each parameter into disjoint choices.

**批注 [DT4]:** "preparation"?

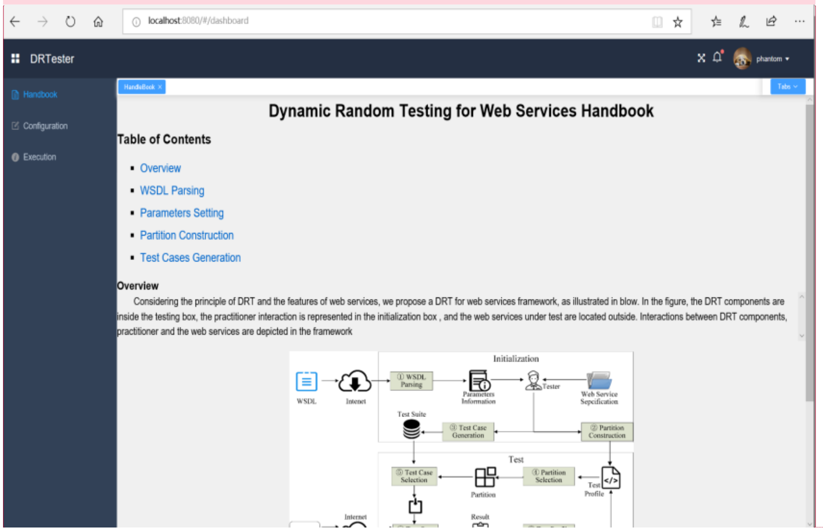
**批注 [DT5]:** Is this OK (we haven't used the acronym before here)

- Partition setting: The tester is responsible for specifying the partitions by combining the choices associated with each parameter.
- Test case generation: The tester is responsible for specifying the mode of test case generation (either randomly generating test cases based on the parameters, or uploading test cases generated using other techniques).

(3) Execution. This feature presents a summary of the testing results, including details of the test case execution (number, input, expected output, partition, and result (pass or fail). For randomly generated tests, the tester has to check each individual result. Otherwise, when all tests have been completed, a report is generated in a downloadable file.

批注 **[DT6]:** Is this correct? Why?

批注 **[DT7]:** Are there captions/explanations for these figures?



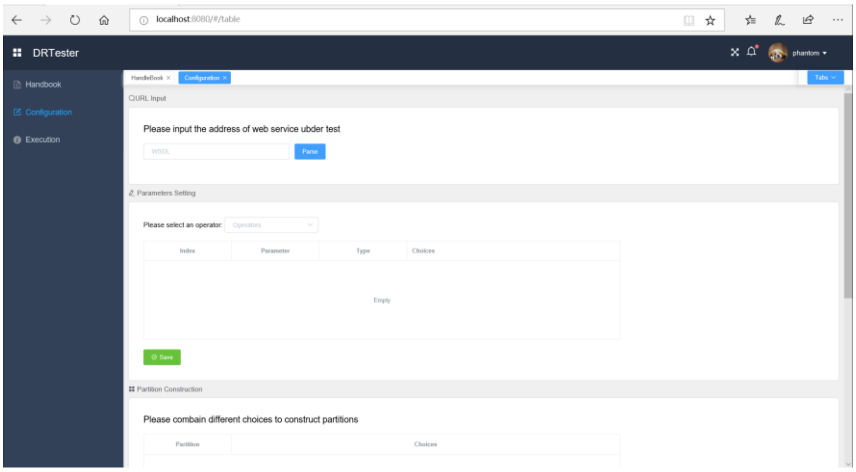
## Dynamic Random Testing for Web Services Handbook

Table of Contents

- Overview
- WSDL Parsing
- Parameters Setting
- Partition Construction
- Test Cases Generation

Overview

Considering the principle of DRT and the features of web services, we propose a DRT for web services framework, as illustrated in blow. In the figure, the DRT components are inside the testing box, the practitioner interaction is represented in the initialization box , and the web services under test are located outside. Interactions between DRT components, practitioner and the web services are depicted in the framework



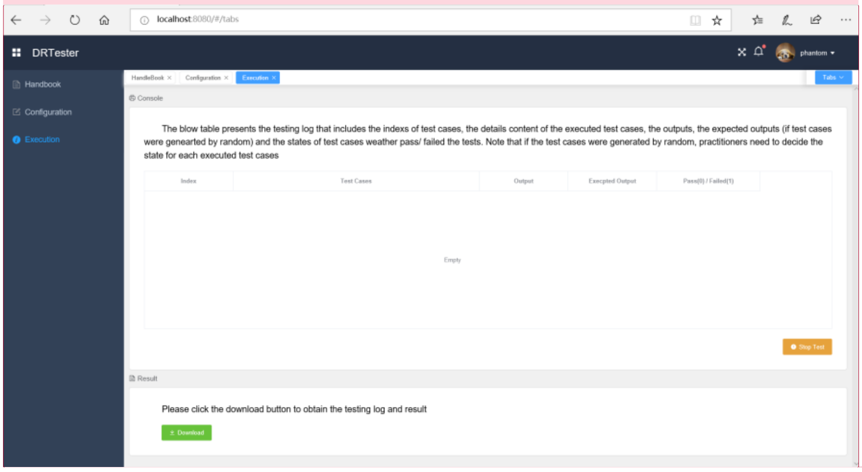
Please input the address of web service ubder test

Please select an operator:

| Index | Parameter | Type | Choices |
|---|---|---|---|
| Empty | | | |

Please combain different choices to construct partitions

| Partition | Choices |
|---|---|



The blow table presents the testing log that includes the indexs of test cases, the details content of the executed test cases, the outputs, the expected outputs (if test cases were genearted by random) and the states of test cases weather pass/ failed the tests. Note that if the test cases were generated by random, practitioners need to decide the state for each executed test cases

| Index | Test Cases | Output | Execpted Output | Pass(0) / Failed(1) |
|---|---|---|---|---|

Empty

Please click the download button to obtain the testing log and result

批注 [DT8]: Some English problems in this figure ...

***R2C4:*** ***The experimentation is conducted generating mutants of three web services. The authors remove equivalent mutants, and mutants that can be detected with less than 20 randomly generated test cases. This latter criterion is strongly influenced by luck: a mean of the number of test cases generated through multiple repetitions is more robust.***

Response: Thank you for this comment. The detailed mutant selection process was not clearly described in the original submission — we apologise for this. Actually, we obtained 50 different test suites using different random seeds, and then tested all mutants with all test suites, calculating the average number of test cases needed to kill (detect) a mutant. Those that could be detected too easily were removed.

In other words, our experiments DID already consider multiple repetitions to address the randomness. This selection process can be seen in our test scripts, available here: https://github.com/phantomDai/evidenceDRT.

Action: In the revised manuscript (Section 4.2), we have provided more details about the mutant selection process for the experiments.

Intended changes are as follows:

To ensure the statistical reliability, we obtained 50 different test suites using different random seeds, then testing all mutants with all test suites, calculating the average number of test cases needed to kill (detect) a mutant. Based on this, those mutants that could be detected too easily were removed.

***R2C5:*** ***Some defections make the repeatability of experiments impossible, more details are required:***

***- (1) The test profile initialization is not explicitly reported, although the authors, in the Section 4.4.2, indicate that "a feasible method is to use a uniform probability distribution as the initial testing profile. On the other hand, testers may also use past experience to guide a different probability distribution as the initial profile".***

Response: We apologize for the test profile initialization details not being clearly stated in the original submission. The initialization can be done in different ways, depending on the tester's knowledge of software under test (SUT). In the experiments, we used a uniform probability distribution as the initial test profile.

批注 **[DT9]:** Are we sure that we want to say this? Are we sure that the test profile initialization depends on the tester's knowloedge of the SUT?

Action: In the revised manuscript (in Section 4.4.2), we have clearly reported the test profile initialization used in the experiments.

Intended details are as follows:

Add the following sentence in the end of Section 4.4.2: "In our experiments, we used a uniform probability distribution for the initial test profile. The initial test profiles of each web service are summarized in Table 12, where <$S_\mathrm{i}$, $P_i$> means that the probability of selecting partition $S_\mathrm{i}$ is $P_i$ ."

Table 12 Summary of Initial Test Profiles of Subject Web Services

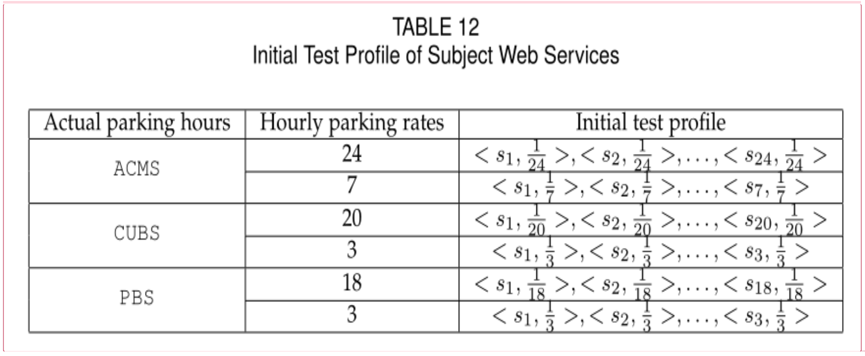


TABLE 12

Initial Test Profile of Subject Web Services

| Actual parking hours | Hourly parking rates | Initial test profile |
|---|---|---|
| ACMS | 24 | $< s_1, \frac{1}{24} >, < s_2, \frac{1}{24} >, \ldots, < s_{24}, \frac{1}{24} >$ |
| | 7 | $< s_1, \frac{1}{7} >, < s_2, \frac{1}{7} >, \ldots, < s_7, \frac{1}{7} >$ |
| CUBS | 20 | $< s_1, \frac{1}{20} >, < s_2, \frac{1}{20} >, \ldots, < s_{20}, \frac{1}{20} >$ |
| | 3 | $< s_1, \frac{1}{3} >, < s_2, \frac{1}{3} >, \ldots, < s_3, \frac{1}{3} >$ |
| PBS | 18 | $< s_1, \frac{1}{18} >, < s_2, \frac{1}{18} >, \ldots, < s_{18}, \frac{1}{18} >$ |
| | 3 | $< s_1, \frac{1}{3} >, < s_2, \frac{1}{3} >, \ldots, < s_3, \frac{1}{3} >$ |

批注 **[DT10]:** Table 12

Initial Test Profiles for Subject Web Services

***R2C6:***- ***(2) The authors set the partitions by making use of decision table, obtaining two partition schemes for each application. The decision tables used in the experiments for the partitioning are not reported.***

Response: We agree that it is necessary to describe the decision tables used to set the partitions. These decision tables were not included in the previous version mainly due to the page limitations. We have now included the decision tables in the revised manuscript (Tables 7 to 9).

批注 **[DT11]:** Where? What table numbers?

Action: In the revised manuscript (before the last paragraph of Section 4.4.1), we have clearly reported the partitions, determined by decision tables, for each web service.

批注 **[DT12]:** I'm not sure that I have captured the intended meaning here. Please check.

Intended details are as follows:

Add the following sentences before the last paragraph of Section 4.4.1: “The decision tables for ACMS, CUBS, and PBS are shown in Tables 7 to 9, respectively. In the tables, $r_i$ (i = 1,…,n) denotes the identified i^th^ *rule*; n is total number of *rules*; and the checkmark under each *rule* indicates that corresponding action should be taken.”

Table 7 ACMS Decision Table

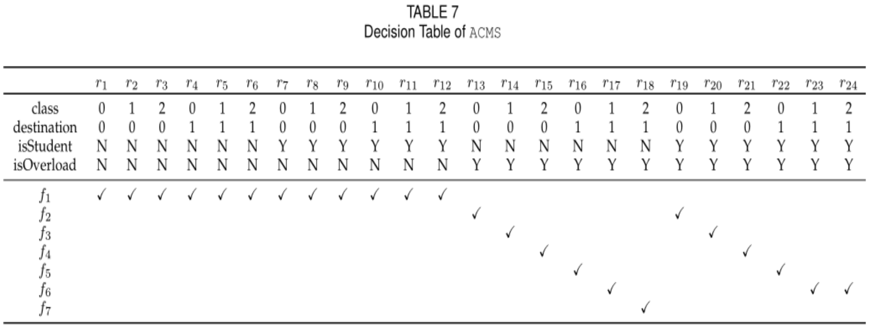


TABLE 7
Decision Table of ACMS

| | $r_1$ | $r_2$ | $r_3$ | $r_4$ | $r_5$ | $r_6$ | $r_7$ | $r_8$ | $r_9$ | $r_{10}$ | $r_{11}$ | $r_{12}$ | $r_{13}$ | $r_{14}$ | $r_{15}$ | $r_{16}$ | $r_{17}$ | $r_{18}$ | $r_{19}$ | $r_{20}$ | $r_{21}$ | $r_{22}$ | $r_{23}$ | $r_{24}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| class | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 |
| destination | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| isStudent | N | N | N | N | N | N | Y | Y | Y | Y | Y | Y | N | N | N | N | N | N | Y | Y | Y | Y | Y | Y |
| isOverload | N | N | N | N | N | N | N | N | N | N | N | N | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| $f_1$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | | | | | | | |
| $f_2$ | | | | | | | | | | | | | ✓ | | | | | | ✓ | | | | | |
| $f_3$ | | | | | | | | | | | | | | ✓ | | | | | | ✓ | | | | |
| $f_4$ | | | | | | | | | | | | | | | ✓ | | | | | | ✓ | | | |
| $f_5$ | | | | | | | | | | | | | | | | ✓ | | | | | | ✓ | | |
| $f_6$ | | | | | | | | | | | | | | | | | ✓ | | | | | | ✓ | ✓ |
| $f_7$ | | | | | | | | | | | | | | | | | | ✓ | | | | | | |

Table 8 CUBS Decision Table

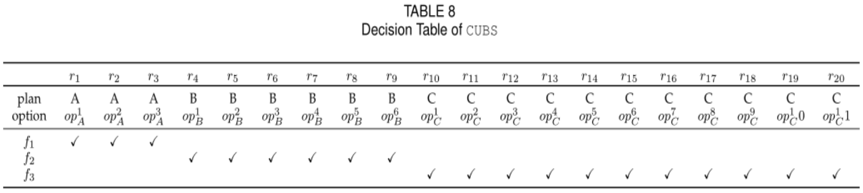


TABLE 8
Decision Table of CUBS

| | $r_1$ | $r_2$ | $r_3$ | $r_4$ | $r_5$ | $r_6$ | $r_7$ | $r_8$ | $r_9$ | $r_{10}$ | $r_{11}$ | $r_{12}$ | $r_{13}$ | $r_{14}$ | $r_{15}$ | $r_{16}$ | $r_{17}$ | $r_{18}$ | $r_{19}$ | $r_{20}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| plan | A | A | A | B | B | B | B | B | B | C | C | C | C | C | C | C | C | C | C | C |
| option | $op_A^1$ | $op_A^2$ | $op_A^3$ | $op_B^1$ | $op_B^2$ | $op_B^3$ | $op_B^4$ | $op_B^5$ | $op_B^6$ | $op_C^1$ | $op_C^2$ | $op_C^3$ | $op_C^4$ | $op_C^5$ | $op_C^6$ | $op_C^7$ | $op_C^8$ | $op_C^9$ | $op_C^1 0$ | $op_C^1 1$ |
| $f_1$ | ✓ | ✓ | ✓ | | | | | | | | | | | | | | | | | |
| $f_2$ | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | | | | | | |
| $f_3$ | | | | | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table 9 PBS Decision Table

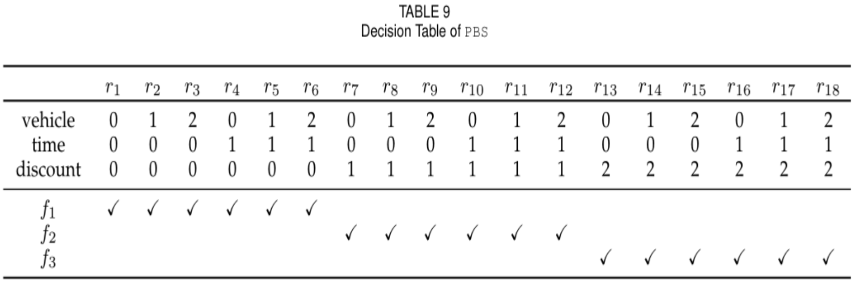


TABLE 9
Decision Table of PBS

| | $r_1$ | $r_2$ | $r_3$ | $r_4$ | $r_5$ | $r_6$ | $r_7$ | $r_8$ | $r_9$ | $r_{10}$ | $r_{11}$ | $r_{12}$ | $r_{13}$ | $r_{14}$ | $r_{15}$ | $r_{16}$ | $r_{17}$ | $r_{18}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| vehicle | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 |
| time | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| discount | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| $f_1$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | | | | | | | |
| $f_2$ | | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| $f_3$ | | | | | | | | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table 7: In order to calculate the costs associated with the additional baggage, we identified those conditions that impact the fee, and the options for each condition (Table 10 summarizes the details, using some simplified symbols for ease of presentation). Table 7 lists all possible actions: the seven formulas for calculating the additional baggage fee. These seven formulas are defined in Table 11, where an “0” indicates that the passenger does not have any additional baggage. The main

批注 **[DT13]:** I'm not sure if this is the correct intended meaning. please check.

difference among the formulas relates to the values of the free baggage, as determined by the destination being domestic or international, and whether or not the passenger is a student (Table 2).

**批注 [DT14]:** I'm not sure that have captured the intended meaning. Please check

Table 10 Conditions and Corresponding Options for the three Web Services Under Study

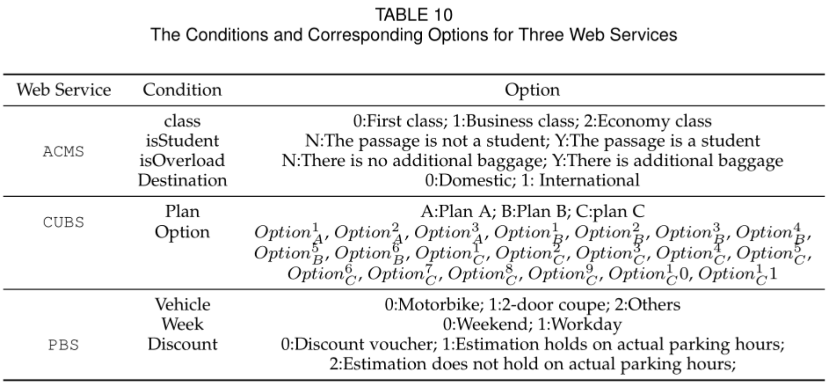


TABLE 10
The Conditions and Corresponding Options for Three Web Services

| Web Service | Condition | Option |
|---|---|---|
| ACMS | class | 0:First class; 1:Business class; 2:Economy class |
| | isStudent | N:The passage is not a student; Y:The passage is a student |
| | isOverload | N:There is no additional baggage; Y:There is additional baggage |
| | Destination | 0:Domestic; 1: International |
| CUBS | Plan | A:Plan A; B:Plan B; C:plan C |
| | Option | $Option^1_A$, $Option^2_A$, $Option^3_A$, $Option^1_B$, $Option^2_B$, $Option^3_B$, $Option^4_B$, $Option^5_B$, $Option^6_B$, $Option^1_C$, $Option^2_C$, $Option^3_C$, $Option^4_C$, $Option^5_C$, $Option^6_C$, $Option^7_C$, $Option^8_C$, $Option^9_C$, $Option^1_C0$, $Option^1_C1$ |
| PBS | Vehicle | 0:Motorbike; 1:2-door coupe; 2:Others |
| | Week | 0:Weekend; 1:Workday |
| | Discount | 0:Discount voucher; 1:Estimation holds on actual parking hours; 2:Estimation does not hold on actual parking hours; |

Table 11 Formulas to Calculate the Costs for the three Web Services Under Study

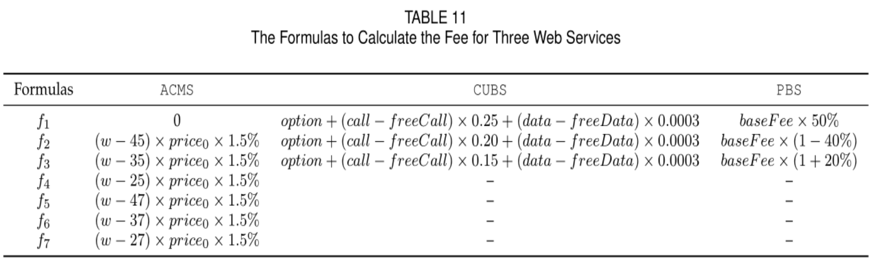


TABLE 11
The Formulas to Calculate the Fee for Three Web Services

| Formulas | ACMS | CUBS | PBS |
|---|---|---|---|
| $f_1$ | 0 | $option + (call - freeCall) \times 0.25 + (data - freeData) \times 0.0003$ | $baseFee \times 50\%$ |
| $f_2$ | $(w - 45) \times price_0 \times 1.5\%$ | $option + (call - freeCall) \times 0.20 + (data - freeData) \times 0.0003$ | $baseFee \times (1 - 40\%)$ |
| $f_3$ | $(w - 35) \times price_0 \times 1.5\%$ | $option + (call - freeCall) \times 0.15 + (data - freeData) \times 0.0003$ | $baseFee \times (1 + 20\%)$ |
| $f_4$ | $(w - 25) \times price_0 \times 1.5\%$ | – | – |
| $f_5$ | $(w - 47) \times price_0 \times 1.5\%$ | – | – |
| $f_6$ | $(w - 37) \times price_0 \times 1.5\%$ | – | – |
| $f_7$ | $(w - 27) \times price_0 \times 1.5\%$ | – | – |

Table 8: In order to help customers calculate their cell-phone bills, we identified those conditions that impact the bill, and the options for each condition (Table 10 summarizes the details, using some simplified symbols for ease of presentation, and "$\mathrm{option}_X^Y$" to indicate option y of Plan X, where $X \in \{A, B, C\}$, and $Y \in \{R|Y \neq 0\}$). Table 8 lists all possible actions: the three formulas for calculating the phone bill, as defined in Table 11. The main difference among the formulas relates to the fee for a one-minute call.

Table 9: In order to calculate the parking fee, we identified those conditions that impact the fee, and the options for each condition (Table 10 summarizes the details, using some simplified symbols for ease of presentation). Table 9 lists all possible actions: the three formulas for calculating the parking fee, as defined in Table 11, where the *baseFee* is calculated based on the type of vehicle, the day of week, and the

total time the car is parked for. The main difference among the formulas relates to the value of the discount.

***R2C7:- (3) The applications are well described, but there is no reference to where they are taken (open source repository, private repository, …).***

Response: We apologize for this omission. The subject web services were developed in our laboratory, based on the real-life specifications.

Action: In the revised manuscript, we have clearly explained where the subject applications are from (Section 4.2).

Intended details are as follows:

We selected three real-life web services as the subject programs for our study, and implemented them ourselves, based on their official specifications: Aviation Consignment Management Service (ACMS); China Unicom Billing Service (CUBS); and Parking Billing Service (PBS).

批注 **[DT15]:** please confirm

***R2C8: In the first Research Question, DRT is evaluated by comparing the effectiveness (in terms of F-, F2- and T-measure) with that of RT and RPT, also if DRT is described in references as an improvement of RT and PT. An additional comparison with a more competitive technique is more significant to appreciate effectiveness of DRT. Some techniques that improve RT and PT are reported by authors in Section 6.2.***

Response: Thank you for the suggestion. As described in Section 6.2, there are two related techniques: adaptive random testing (ART) and adaptive testing (AT). ART attempts to improve the fault detection ability of RT by evenly spreading the test cases throughout the input domain. AT attempts to do so by introducing feedback into the RT process. Based on the suggestion, and due to their similar rationales, we have decided to include AT as an additional benchmark technique.

Action: In the revised manuscript , we have added AT as an additional benchmark technique (Section 4.3.1 and Section 5).

For discussion: Although we have completed additional experiments with AT. I am still not sure whether this is should add or not. Please suggest.

批注 **[DT16]:** What will be the impact on the original results and conclusion?

***R2C9: In the second Research Question (Section 5.2), the definition of failure rate is not clear: it is defined as the ratio between k and ki, where k is the number of test cases until revealing a fault and ki is the total number of test cases in si, that could be infinite. For instance, if a parameter can take all real values between 1 and 10, the number of inputs is countable infinite. This formulation of failure rate is more proper of test case "selection" algorithm. In this case, a solution is to consider ki as***

***the total number of test cases performed to reveal an error and k equal to 1. Failure rate should be defined as #number of failure/#number of executed tests.***

Response: We apologize for the confusing description in the original submission. ~~Thank you for the comment: we agree that the failure rate should be defined as #number of failure/#number of executed tests. Unfortunately~~, because the studied web services all have infinite input domains, it is not possible to follow this definition in our experiments. In order to approximate the failure rate of each partition, we agree with the solution suggested by the reviewer: the failure rate $\theta_i$ of partition $s_i$ can be calculated as 1/ $k_i$ (where $k_i$ is the total number of test cases executed before revealing a fault). We have checked the results presented in the previous submission, and can confirm that this solution was actually what we used. For further details, please refer to the test scripts that are available on: https://github.com/phantomDai/evidenceDRT

Action: In the revised manuscript, we have rewritten the definition of the failure rate (the second paragraph of Section 5.2).

Intended details are as follows:

Before starting the test, it is necessary to know the failure rate $\theta_i$ of partition $s_i$. From Tables 2-6, it can be observed that the values of some parameters (such as the baggage weight, the call duration, and parking duration) are such that the total number of test case values in a partition could be infinite. For such a situation, we approximate the failure rate $\theta_i$ of $s_i$ by 1/$k_i$ (where $k_i$ is the total number of test cases executed before revealing a fault).

***R2C10:*** ***The purpose of the Third Research Question is to validate that DRT requires linear time to generate test case through empirical examination of the actual test case generation and execution. Instead, in Section 5.3 the focus is on the comparison among the three techniques, without any reference to the temporal complexity. For this reason, the answer of the Research Question must be revised.***

Response: Thank you for the comment. We agree that in the original submission, the third research question (Section 4.1) did not match the answer (Section 5.3). Originally, we wanted to evaluate the fault detection efficiency of DRT in terms of its time cost. Compared with RT and PT, DRT incorporates the selection of partitions and test cases within a partition. Thus, we are interested in comparing the fault detection efficiency of DRT, RT and PT, in terms of their time costs. We have decided to rephrase the third research question (Section 4.1) to match our original intention.

批注 **[DT17]:** What about AT?

Action: In the revised manuscript, we have rephrased the third research question (Section 4.1) and the title of Section 5.3.

Intended details are as follows:

Changes to Section 4.1:
"RQ3 Compared with the baseline techniques, how efficient is DRT at detecting web service faults in terms of time cost?
Compared with RT and PT, DRT incorporates the selection of partitions and test cases within a partition. Thus, we are interested in comparing the fault detection efficiency of DRT, RT and PT, in terms of their time costs. To answer this, we compare the time cost of DRT with that of the baseline techniques."
Changes to Section 5.3:
"Section 5.3 RQ3: Fault detection efficiency"

批注 **[DT18]:** What about AT?

***R2C11:*** ***In some cases, the results graphics could be scaled better, because the difference between the various elements is not very appreciable in the printing of paper.***

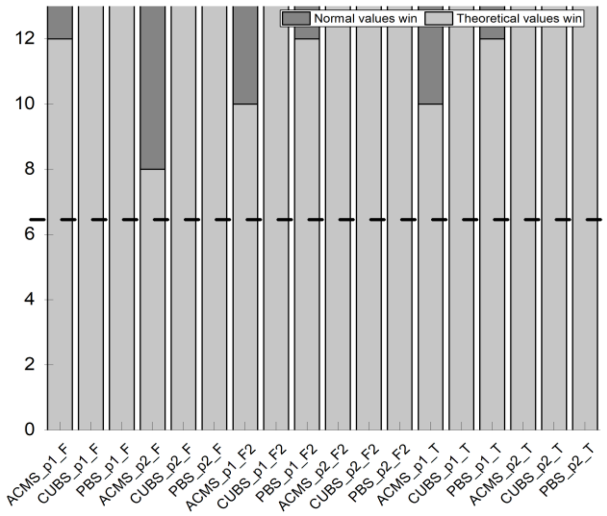
Response: Thank you for the comment. We agree the some of the figures in the original submission (including Figures 3~9) were a bit crowded, which was done in an effort to save the space. This resulted in, amongst other things, some points overlapping in Figure 6, making it difficult to distinguish among them. To improve the presentation, we have decided to take the following actions: (1) Figures 3~9 have been extended with more details and scaled better; (2) A stacking bar chart has been added to summarize the fitness of the experimental results with the theoretical analysis; (3) Tables 9~11 have been moved to the appendix.

批注 **[DT19]:** I'm not sure that I have captured

Action: In the revised manuscript, we have followed the reviewer's suggestion to redraw all figures to improve the presentation. Significant changes have been made to Section 5.

The following will be added to summarize the fitness of the experimental results with the theoretical analysis:

批注 **[DT20]:** I'm not sure that I have captured



***R2C12:*** ***The article is well written and it is easy to read. An incorrect tables layout is evident: Table 5 is printed after Table 2 and before Tables 3, 4 and 6.***

Response: Thank you for the suggestion.

Action: In the revised manuscript, we have followed the suggestion to rearrange the layout of Tables 2~6.

Reviewer 3’s comments

***R3C1:* The paper investigates the quality of service-based applications and proposes a dynamic random testing (DRT) technique for web services that improves existing methods. The authors present the technique, a framework for its usage and a prototype implementation. An empirical study including three different case studies is also included to show the effectiveness of the proposed approach.**

Response: Thank you for the endorsement.

Action: None.

***R3C2:* The paper deals with an interesting topic and the proposed method extends an idea already proposed in [1]. I have a few comments about the current version of the paper.**
***-The authors should explicitly state in the Introduction the novelty of this paper with respect to their previous publication. The method and the prototype seem to be already present in [1] together with some empirical studies. Besides the original idea has already been presented in [7]. Even if a list is provided in the cover letter, a clear statement about the differences is necessary here.***

Response: Thank you for the comment. We agree that a revision of the Introduction section to include an explicit statement of the novelty of the paper (with respect to the previous publications [1] and [7]) should enhance the paper. We have therefore moved the description of substantial extensions to previous work from the cover letter to the main paper.

Action: In the revised manuscript, we have followed the suggestion to add a clear explanation of the differences between this and the earlier work (third last paragraph of Section 1).

Replace “we examine key issues … in terms of fault detection efficiency” with the following:

We examine key issues of such an adaptation, and, accordingly, propose a framework for testing web services that combines the principles of DRT [7] and the features of web services. To validate the fault detection effectiveness and efficiency of the proposed DRT method in the context of SOA, we conduct a comprehensive empirical study. We also explore the impact factors of the proposed DRT, and provide guidelines for setting DRT parameters based on a theoretical analysis. Finally, we compare the performance of the proposed DRT with other baseline techniques.

Replace “The contributions of this work include” with the following:

This paper extends our previous work [1] in the following aspects. Firstly, this paper extensively examines the challenges and practical situations related to testing web services (Section 2.2). It also extensively discusses the limitations of RT,

**批注 [DT21]:** We may need to come back later and make a decision on using present or past tense for this text.

**批注 [DT22]:** Is this the first time to use RT?

Partition Testing (PT), and Random Partition Testing (RPT), when they are used for testing web services (Section 1). Secondly, although previous work [1] provided a coarse-grained framework for DRT of web services, PT was not studied. In contrast, this paper provides a comprehensive solution based on partitioning (Section 4.4.1). Thirdly, based on a theoretical analysis (Section 3.2), this paper provides guidelines for setting DRT parameters. Such guidelines are crucial to enhance the practical application of DRT, which was not covered in previous work [1]. Fourthly, previous work [1] only evaluated the fault detection effectiveness and efficiency of the proposed approach (DRT) in terms of the F-measure and T-measure, and only two small web services (ATM Service and Warehouse Service) were used in the evaluation of its performance. This paper, in contrast, provides a comprehensive evaluation that not only evaluates the fault detection effectiveness of the proposed approach in terms of the F-measure, F2-measure, and T-measure (Section 5.1), but also evaluates its efficiency in terms of F-time, F2-time, and T-time (Section 5.3). Furthermore, we also examine three real-life web services, comparing the fault-detection effectiveness and efficiency of the proposed approach with those of RT, RPT, and AT. Statistical analyses were used to validate the significance of the empirical evaluations and comparisons (Sections 5.1 and 5.3), which was not covered in previous work [1]. Extending again the previous work [1], we also examine the relationship between the number of partitions and the optimal control parameter settings for DRT, evaluating the usefulness of guidelines provided by the theoretical analysis (Section 5.2). Fifthly, we substantially extend the previous literature review [1] (Section 6). The contributions of this work, combined with previous work [1], include:

**批注 [DT23]:** Anything about AT?

**批注 [DT24]:** Has this been introduced already?

**批注 [DT25]:** This feels like a comment for a response letter, not something that we should put in the actual paper. Please consider…

***R3C3:* The cover letter describes a set of improvements concerning the writing (points (i), (ii) and (vi)) and some other major extensions concerning the presentation of the framework, the definition of guidelines about parameters settings in DRT and a more thorough empirical evaluation. From my point of view, the most consistent improvement is the evaluation part. The other ones need more clarification, as described below**

Response: Thank you for the comments. We have added further clarifications, as suggested. For more details, please refer to our responses to R3C4, R3C5, R3C6, and R3C7.

Action: None.

***R3C4:* Section 3 describes the application of DRT to web services. The novel part described in Section 3.2 needs some rewriting. I understand the importance of parameters setting and the need of mathematical treatment, but as it is now it is not easily understandable. A high-level description of the procedure and of the findings is necessary, together with a description of the followed procedure. The mathematical demonstrations and theorem should be moved to an Appendix. At**

***present these details disrupt the reading flow and at the end of Section 3.2, it is not clear how to practically set the parameters.***

Response: Thank you for the suggestion. We agree that a high-level description of the procedure and the findings would help to enhance the paper's readability. Accordingly, we now provide some general guidance for better understanding how the findings are achieved. We have also moved the proof to the appendix, to avoid disrupting the reading flow. We examine each aspect of setting the DRT parameters setting and provide further guidelines for using them individually.

**批注 [DT26]:** I'm not sure what the ordinally intended message/meaning is: could you rephrase? What are the "elements in the parameter setting"?

Action: In the revised manuscript, we have added a high-level description of the findings and moved the related proofs to the appendix. We have also provided detailed guidelines for how to set the parameters.

Intended high-level description of the parameter settings:
In order to achieve the best performance, the probability of selecting the partition $s_M$ (which has the maximum failure rate) is expected to increase. To achieve this, the increase in probability of $s_M$ being selected for the next round should be larger than that for other partitions. The further analysis of sufficient conditions can result in an interval of ε.

**批注 [DT27]:** I don't understand the intended meaning here. Can you rephrase?

Intended practical guidelines for setting parameters:
For a given partition scheme with a total number of $m$ partitions, identification of the partition with the minimum failure rate ($\theta_{min}$) first requires calculation of the failure rates of each partition, then identification of the minimum. Each partition's failure rate can be obtained in two ways: (1) it can be calculated directly as (#number of failures)/(#number of executed tests), if the test history of the web service under test is accessible; or (2) it can be approximated by $1/k_i$ (where $k_i$ is the total number of test cases executed before revealing a fault).

**批注 [DT28]:** Please confirm that we are talking about the minimum, not maximum

***R3C5:*** ***Section 3.3 describing the prototype should be expanded. Now it looks similar to the description in [1], while I would expect a more detailed description here together with the possibility to experiment with the tool for the sake of replicability.***

**批注 [DT29]:** The responses to this are basically identical to those for R2C3

Response: Thank you for the comment. We agree that the original description was ~~very~~ thin, and ~~very~~ similar to the conference paper's version, which was mainly due to the page limitation. We have decided to follow the suggestion to add more details about the prototype, and have made substantial efforts to enhance the prototype reported in the conference paper. The enhancements include: (1) a guidance feature to guide usage of this prototype; (2) a configuration feature to set the DRT parameters, partitions, and test case generations; and (3) an execution feature to provide information about the WSUT execution process, and the testing results.

Action: In the revised manuscript, we have updated the description of the enhanced prototype (Section 3.3).

Intended details description and usage of prototype are as follows:

(1) Guidance. This feature describes the steps the tester should follow when testing a web service.

(2) Configuration. This feature interacts with the testers to obtain and set the information related to testing the web service, including: the address of the web service under test; the DRT parameters and partition details; and the test case generation. The detailed steps are as follows:

**批注 [DT30]:** "preparation"?

   a) Inputting and parsing the URL: We integrate the WSDL parsing functionality provided by MT4WS (*C-A. Sun et al. MT4WS: an automated metamorphic testing system for web services, IJHPCN 9(1/2): 104-115, 2016*). This enables all the (WSDL) parameters and their types to be automatically obtained.

   b) Parameter setting: The tester is responsible for selecting which operations of the current WSUT are to be tested, and for partitioning each parameter into disjoint choices.

**批注 [DT31]:** Is this OK (we haven't used the acronym before here)

   c) Partition setting: The tester is responsible for specifying the partitions by combining the choices associated with each parameter.

   d) Test case generation: The tester is responsible for specifying the mode of test case generation (either randomly generating test cases based on the parameters, or uploading test cases generated using other techniques).

(3) Execution. This feature presents a summary of the testing results, including details of the test case execution (number, input, expected output, partition, and result (pass or fail). For randomly generated tests, the tester has to check each individual result. Otherwise, when all tests have been completed, a report is generated in a downloadable file.

**批注 [DT32]:** Is this correct? Why?

The back-end logic is composed of several Restful APIs and Java classes: The APIs are responsible for communicating HTTP messages to and from the front-end interface. The controller class is responsible for updating the test profile according to the test results, and for selecting test cases from the partitions. The selected test cases are wrapped in SOAP messages and sent to the web service under test through the proxy class, which also intercepts the test results.

***R3C6:*** ***Section 4 reports the empirical studies conducted to evaluate the performance of the proposed method. A set of research questions are described and then answers in the results section. The authors may consider the possibility to anticipate the research questions as a way to motivate the paper in the Introduction or in a section describing the research approach followed in this paper.***

Response: Thank you for the suggestion. Originally, we had presented the research questions in the *Empirical Study* section, and the answers in the *Experimental Results* section, to provide a structured presentation. However, we agree with the suggestion that moving the research questions to the Introduction may help improve the flow and presentation. Accordingly, we have now embedded the research questions in the Introduction as goals to motivate the work of the paper. Please also refer to our response to R3C2.

Action: In the revised manuscript, we have followed the suggestion and restructured the Introduction, adding the research motivation before presenting the contributions (Section 1).

***R3C7: The empirical study itself is quite interesting. I would suggest adding a subsection summarising the results and possible limitations discovered during the experimentation. Besides, since the three different case studies have more or less the same dimensions in terms of LOC, it would be nice understanding the scalability of the proposed approach.***

Response: Thank you for the suggestion. We agree that adding a subsection summarizing the results and limitations would enhance the paper. Our approach is a kind of black-box testing technique that is an enhanced version of random and partition testing. Because random testing and partition testing are widely applied in practice, we believe that our proposed approach can also be applied to larger subject applications. As explained earlier, because our evaluation requires to access to the source code of the web services (in order to seed faults), the current evaluation only involves relatively small web services. Nonetheless, the actual approach itself is not limited to small services.

Action: In the revised manuscript, we have followed the suggestion and added a new subsection (Section 5.4) summarizing the results and possible limitations.

Intended results and limitations are as follows: (in Section 5.4)

5.4 Summary

Based on the evaluation, we have the following observations:

- DRT outperforms RPT and RT, according to all the applied metrics for all three studied web services. DRT performs slightly less well than AT in terms of the F-, F2-, and T-measures, for all the studied web services. However, AT incurs heavier computation overheads, and takes a significantly longer time. (For instance, AT takes … haven't counted the experimental data yet.). This indicates that among RT, RPT, and AT, DRT should be chosen.
- DRT is more effective in terms of F-, F2-, and T-measures when the parameter settings are optimal (according to the theoretical analysis): In most cases, DRT has the best performance for all three object web services,

批注 **[DT33]:** ?

批注 **[DT34]:** Does the evidence support this statement? We may need to elaborate a bit more.

批注 **[DT35]:** I'm not sure that I understand. More effective than what? The previous point says that AT outperforms DRT according to F-, F2, and T-measure … Please clarify.

according to these three metrics (F- measure, F2- measure, and T-measure) when following the guidelines for the parameter settings. This highlights the usefulness of the parameter-setting guidelines.

We also note the following limitations:

- While DRT outperforms RT and RPT in terms of fault detection effectiveness and efficiency, this is achieved at the cost of the additional effort required to set the partitions and test profiles.
- Applying DRT involves setting parameters, which may not be trivial. Even when following the theoretical guidelines.

批注 **[DT36]:** No mention of AT?