A Dissimilarity with Dice-Jaro-Winkler Test Case Prioritization Approach for Model-Based Testing in Software Product Line

R. Aduni Sulaiman[1], Dayang N. A. Jawawi[1*], and Shahliza Abdul Halim[2]

[1] Software Engineering Department, School of Computing
Faculty of Engineering Universiti Teknologi Malaysia
81300 Skudai Johor, Malaysia
[e-mail: raduni2@live.utm.my, dayang@utm.my]

[2] Faculty of Computer Science and Information System
Universiti Tun Hussein Onn Malaysia
86400 Parit Raja, Johor, Malaysia
[e-mail: shahliza@utm.my]

[*]Corresponding author: Dayang N.A. Jawawi

*Received November 16, 2020; revised January 18, 2021; accepted February 15, 2021;
published March 31, 2021*

Abstract

The effectiveness of testing in Model-based Testing (MBT) for Software Product Line (SPL) can be achieved by considering fault detection in test case. The lack of fault consideration caused test case in test suite to be listed randomly. Test Case Prioritization (TCP) is one of regression techniques that is adaptively capable to detect faults as early as possible by reordering test cases based on fault detection rate. However, there is a lack of studies that measured faults in MBT for SPL. This paper proposes a Test Case Prioritization (TCP) approach based on dissimilarity and string based distance called Last Minimal for Local Maximal Distance (LM-LMD) with Dice-Jaro-Winkler Dissimilarity. LM-LMD with Dice-Jaro-Winkler Dissimilarity adopts Local Maximum Distance as the prioritization algorithm and Dice-Jaro-Winkler similarity measure to evaluate distance among test cases. This work is based on the test case generated from statechart in Software Product Line (SPL) domain context. Our results are promising as LM-LMD with Dice-Jaro-Winkler Dissimilarity outperformed the original Local Maximum Distance, Global Maximum Distance and Enhanced All-yes Configuration algorithm in terms of Average Fault Detection Rate (APFD) and average prioritization time.

Keywords: Model-based Testing, Software Product Line, Software Testing, Test Case Prioritization

http://doi.org/10.3837/tiis.2021.03.007 **ISSN : 1976-7277**

1. Introduction

A Software Product Line (SPL) is one of the paradigms for systematic reuse that guides organization to develop products from core assets rather than develop products from scratch. There are two major activities in SPL that focus of core assets development in domain engineering and product development in application engineering. Development of core assets is based on identification of reusable assets. In order to develop reusable core assets, SPL must have ability to exploit commonality and manage variability. Due to the explosion in the number of products, SPL requires an exhaustive testing technique to manage products. SPL testing is aimed to minimize testing effort while at the same time produce effective testing results. One of the most promising techniques is Model-Based Testing (MBT), which offers systematic automation in test cases generation [1]. There are two main steps, which are to obtain requirements to be presented in the test model and derive test cases. MBT offers automated, rigorous and systematic testing early in the software life cycle stage (modelling stage).

The quality of test cases covers two main aspects, which are the cost of testing and effectiveness of test cases. In recent years, multi-objective technique has been proposed to cover multiple test case quality measures in SPL. Effectiveness of testing in MBT for SPL is commonly measured by using coverage criteria. However, there are lack of studies that implemented a multi-objective criterion that proved effective in fault detection rate in test suites of MBT statechart for SPL. Effectiveness in fault detection can be discovered efficiently by using Test Cases Prioritization (TCP) Technique [2]. However, the lack of TCP implemented by using multi-objective caused the fault to be unable to be revealed earlier. Thus, it highlights the need for TCP in order to balance a trade-off between cost and effectiveness measure. Studies [3-5] defined optimization based on prioritization problem by using Search-Based and Heuristic-Based Technique. However, there are arguments concerning which approach can give the best quality of test case in terms of cost and effectiveness measure for SPL testing. It showed the importance of optimization approach that can give a minimal cost of testing and maximize effectiveness for MBT in SPL testing.

These two quality measures require a balanced trade-off that is significant to produce a good test case quality. Based on our experience working in MBT for SPL, after a test case has been generated, the test case still needs to be validated in terms of existing faults to ensure that the good test case in terms of cost and effectiveness can be considered first. Test case generation was the focus in our previous work [6] where we proposed a test case generation technique. The result of the study produced a set of test suites with consideration of cost and effectiveness measure. Here, the proposed TCP approach is used to reorder test case in test suite by using dissimilarity technique. Our goal is to reorder test cases based on higher faults detected with minimal execution time. In this paper, we explore the TCP based on dissimilarity measure for test case from MBT in SPL domain perspective. We present TCP technique based on prioritization algorithm with string distance measure. There are also performance comparison against existing TCP approach for SPL to establish the viability of LM-LMD with Dice-Jaro-Winkler Dissimilarity. In the evaluation part, we evaluate the cost and effectiveness of TCP approach by using SPL e-shop test object. Specifically, our contribution is to create a test case prioritization approach LM-LMD with Dice-Jaro-Winkler Dissimilarity that based on enhanced string distance called as Dice-Jaro-Winkler Distance to measure dissimilarity distance between test cases in test suite generated from statechart MBT for SPL and enhanced Local Maximum Distance algorithm that is integrated with enhanced string distance. The rest of this study is organized as follows. Section 2 presents the theoretical background. Section 3 presents the related work, while Section 4 presents the overview of proposed approach. Section 5 presents the empirical study and Section 6 describes the example used in this paper. Section 7 presents the result and threat to validity is presented in Section 8. Finally, the conclusion in presented in Section 10.

2. Theoretical Background

This section will present the background of the concept of SPL, MBT and TCP implemented in this study.

2.1 Overview of Software Product Line Testing

SPL is defined as a collection of related software products that share same functionalities, but at the same time still differ from each other based on specific features. New products can be derived by reusing and combining the software assets in an effective way. Assets that appear in all products are known as commonalities, whereas assets that only existed in some products are known as variability. In SPL testing context, the explosion in the number of possible products caused exhaustive testing to be infeasible. This issue gives challenge to select relevant subset of product for testing. A basic way to conduct testing for SPL is by using standard techniques which are used in single system to be applied for SPL products. However, this takes higher cost and time consumption to evaluate every single product.

2.2 Model-based Testing for Software Product Line

The basic idea of MBT is to systematically minimize effort by exploiting knowledge of core assets. MBT for SPL is used to capture behavior of SPL. MBT process starts with the test model development which is built based on requirement specifications. Then, test selection criteria will be defined to derive good test cases. A good test case is one that can detect faults earlier with higher effectiveness measure such as structural-based criteria [7]. In the scope of SPL testing, previous implementation of MBT faced issues of test model development [8-12], redundancy in test suites [3, 12, 13], and quality measure, which are effectiveness of test cases [15-18] and cost of testing [19-21].

In this study, we focus on the cost and effectiveness measure in MBT for SPL. These two measures can be considered as an important evaluation, since they are related to each other in order to produce a good testing result. For example, once the tester has achieved the maximum target to find defects, studies have been unable to ensure that these activities significantly reduce the testing effort and cost. Furthermore, as highlighted by Inozemtseva and Holmes [22], a good coverage test suite does not guarantee that the test suite is effective enough. In order to improve effectiveness in fault detection in test cases, another technique, which is Test Case Prioritization (TCP), is applied to enhance the fault detection rate. The regression testing based on TCP can be implemented to improve effectiveness based on earlier fault detection. The lack of TCP implementation made it difficult for faults to be revealed as soon as possible, thus making it possible for the wrong products to be executed.

2.3 Test Case Prioritization for Model-based Testing for Software Product Line

Yoo and Harman [23] divided regression testing into three main categories, which are test selection, minimization and prioritization. Each category has different goals and different implementation process. The goal is to test the software after changes and to ensure the software still works correctly. In the scope of MBT in SPL, regression testing is used to utilize reusability technique to minimize test effort. Adoption of regression testing, for example retest, tests artefacts of variability and can lead to a reduction of testing effort [24]. The combination between two testing approaches, which are MBT and regression testing, helps to reduce a set of test cases to be re-executed. It required integration between test model to facilitate test case generation and regression testing is retest test selection in order to stimulate reusability of test cases. By considering the fault problem in test cases, TCP is the technique that is most suitable to be selected for further use in this research study [4].

While conducting TCP for generated test cases from MBT in SPL, the goals of testing are identified. This is related to the effectiveness of the algorithm to measure faults and at the same time, demand to reduce time to conduct TCP [25]. The goal in TCP for MBT in SPL can be divided into three, which are effectiveness measure (fault detection), minimize execution time and successful algorithm used. Effectiveness of test cases is not only based on coverage. Another important element is fault detection capability in proposed TCP technique. It is a good means to ensure that the SPL testing considers not only functional testing result (coverage), but that it is also validated based on test cases [17]. This is because the demand of industry in SPL that is focused on maximization for certain testing property, for example, coverage and early fault detection. Furthermore, reusability of test cases can be achieved by reusing potential test cases in reuse-optimize process of TCP in measuring faults. This is helpful to improve reusability in generated test cases.

One cost measure that is very important in TCP is execution time. Kazmi et al. [26] discussed that the execution time can be minimized by reducing other goals, for example fault and coverage criteria. Test case execution time can differ depending on the number of states. Another goal of TCP is to adapt a good technique at the fastest rate with increased fault detection. It is also important in TCP because a good technique is the main contributor to other measurements, for example cost or effectiveness measure. It leads to a different type of technique proposed to handle prioritization, for example search-based algorithm and similarity-based algorithm. The technique proposed will be used to arrange the configuration of testing depending on the specific objectives.

2.4 Dissimilarity Technique for Test Case Prioritization

In order to implement TCP for SPL, many different types of approaches have been proposed including similarity and dissimilarity measure. Testers have demanded that faults be revealed as soon as possible by using different techniques. The concept of similarity and dissimilarity are based on the measurement distance between two test cases. This approach will evaluate test cases by comparing two similarities between test cases. Similarity and dissimilarity approach was introduced due to scalability issue in existing prioritization algorithm. This technique offers a simple, scalable and effective way to reduce the number of test cases and to prioritize it.

Researchers have also highlighted that for SPL, this concept can be used to detect fault faster compared with other approaches [5, 26]. Similarity is required as a core element to measure dissimilarity. Henard et al. [5] proposed similarity-based technique to maximize dissimilarity measures in a set of test cases. Jaccard Distance and Dice anti Dice were used measure distance between two test cases. Two techniques, which are Local Maximum Distance and Global Maximum Distance, are proposed to list the results of dissimilarity measured. This technique is very useful since it can improve the time for execution and different techniques have been discussed to implement dissimilarity measure. Distance measure is a common method to handle prioritization. In the single system testing, string distance is commonly used in record association to recognize the duplication in computerized files. It is also used to detect redundancy to improve accuracy level. It can be considered as a new method to evaluate prioritization based on SPL.

3. Related Work

In order to enhance effectiveness of testing, earlier fault detection needs to be measured. TCP is the one of the techniques that can reveal faults earlier by reordering test cases based on fault detection rate. Existing studies proposed TCP approach based on Feature Model and mathematical model (Markov Chain) [4, 5, 24, 27]. However, there is a lack of TCP approaches used to evaluate test cases from UML statechart test model artefacts. This is due to the previous studies that highlighted the effectiveness measure of UML statechart test model artefacts by using coverage criteria, for example structural-based coverage. This has led to faults to not be considered an effectiveness measure for test cases generated from MBT in SPL.

In addition, there are different TCP based technique implemented to handle prioritization for SPL testing. Weight-based prioritization is implemented for SBT. This approach will assign weight for each test case based on objectives defined. Then, test cases would be prioritized based on weight assigned. Wang et al. [28] proposed a multi-objective test case prioritization based on (1+1) Evolutionary Algorithm (EA). The fitness functions are constructed based on the cost-effectiveness measure. Four different weight values are manually assigned by the tester for the fitness function proposed. Many TCP techniques have been introduced to improve TCP in MBT for SPL. Schaefer et al. [29] introduced dissimilarity measure for delta-oriented architecture test model, whereas Lity [24] implemented similarity measure for delta-oriented architecture test model for SPL. Wang et al. [30]applied SBT based on (1+1) Evolutionary Algorithm to solve TCP issue in MBT for SPL, whereas Devroey [31] used statistical prioritization algorithm for feature transition system test model.

Delta-oriented based prioritization is implemented once the tester has used architecture model as a test model. Schaefer et al. [29] integrate component-based, delta-oriented prioritization and dissimilarity values. Weight value is assigned with dissimilarity measure to prioritize test cases and formula is constructed based on dissimilarity in component-based method. This work is still vague as only a small number of improvements for fault detection are found compared to existing study. Al-Hajjaji et al. [25] perform delta-oriented prioritization based on dissimilarity between the first and second products by means of deltas. The new function of delta similarity is based on Hamming Distance and common deltas are used to represent the number of common deltas. Statistical prioritization is used to improve the prioritization technique by using test model of Markov Chain and Feature Transition System (FTS). Devroey et al. [32] implemented statistical prioritization by using FTS and probability to reorder test cases. However, this technique is dependent on the domain-case study, which caused difficulties to adapt the technique to other test scenarios.

Papadakis et al. [27] used similarity measure for prioritization, but it was implemented for mutation testing process. This proposed approach only focuses on prioritization based on similarity algorithm without any specific technique to choose the test cases in the configuration. Sahak et al. [33] proposed similarity-based TCP based on enhancement of string distance and prioritization algorithm. It was used to improve the enhancement of existing algorithm to achieve faster rate of fault detection and to minimize testing effort. Devroey et al. [34] combine the implementation of statistical-based and dissimilarity-based prioritization. The proposed algorithm will sort list of test cases based on coverage criteria or weight that has been assigned in the test model. This study is based on family-based prioritization for variability-based test model for model-checking purpose. Al-Hajjaji et al. [25] combined TCP approaches, which are delta-oriented based and configuration-based, that measure based on weight factors assigned. Egyed et al. [4] used NSGA-II algorithm to prioritize test cases. Objective function based on dissimilarity of test cases is proposed to measure the distance of test case results. The results of the proposed algorithm show that the combination metrics between dissimilarity and faults provide the best results in experiments compared with other comparison metrics.

It is shown that the distance measure technique is one of the most important techniques to improve the performance of effectiveness results, for example faults and execution time. Based on previous studies, string distance has proven to be the one of the best techniques used in similarity / dissimilarity-based prioritization. This motivates researchers to apply the string distance for similarity / dissimilarity-based as a technique for prioritization of test cases from statechart test generation. As per result of existing study, it shows lack of works that focus on implementation of string distance based for the scope of test case format of statechart test model. As summary, techniques of prioritization have been discussed [24, 34]. However, previous work focused on prioritizing products based on configuration generated from FM. In the scope of other models, especially in test model of MBT, there is a lack of existing study that has applied prioritization technique in SPL context.

The implementation of similarity test cases concept in prioritization technique has been discussed widely by researchers. However, recently some pieces of work have shown that the dissimilarity test cases have proven to be one of the techniques that can accelerate higher fault detection rather than the similarity technique.Two string distance techniques, which are Dice Similarity Measure and Jaro-Winkler, have been selected as techniques to evaluate distance between test cases since they can produce best distance functions compared with other techniques. In software testing, many algorithms have been proposed, for example, Local Maximum Distance, Global Maximum Distance, Farthest-first Ordered Sequence and All-Yes Configuration. The effectiveness of these algorithms is still under investigation since there is a lack of work that has implemented it, especially in SPL context.

4. Overview of the Proposed Approach

The proposed approach utilized string distance with dissimilarity measure as a technique to prioritized test case. The proposed TCP approach is illustrated in **Fig. 1**. The process of proposed prioritization algorithm starts by extracting test suite based on the results from the proposed generation algorithm. Then, the mutant test suite is created. The mutant test suite is generated based on mutant versions of test model based on test objects used. The information between original result and mutant result are extracted in order to prioritize test cases. The proposed approach is divided into two main processes, which are (i) measurement of test case dissimilarity by using string distance Dice-Jaro-Winkler, and (ii) picking the order of test cases by using the proposed approach.

First process starts with measurement of dissimilarity distance between test cases generated. The selection of distance techniques is based on the best results of distance measurement from previous studies. The proposed string distance is also carried out in this part. The enhanced Dice and Jaro-Winkler Distance is implemented to improve dissimilarity measure distance between two test cases. Dissimilarity measured technique is used to evaluate distance of original test suite and the mutant test suite. The prioritization algorithm based on enhancement from Local Maximum Distance is used to evaluate distance results proposed in LM-LMD with Dice-Jaro-Winkler Dissimilarity results. Here, the multi-objectives prioritization is concerned with two objectives based on minimization of execution time, while at the same time focused on maximization of fault detection capability rate. APFD metric is used to as indicator to select the highest fault detected.

4.1 Phase (I): Dissimilarity Measure: Dice with Jaro-Winkler Algorithm

At first, in order to measure distance between two test cases, dissimilarity technique is implemented. Dissimilarity measure is a technique used to evaluate the distance between two test cases. The concept of dissimilarity distance has been proposed by Henard [35] and used to evaluate dissimilar test suite in MBT. It became a technique that can increase fault detection rate compared with similarity distance technique.

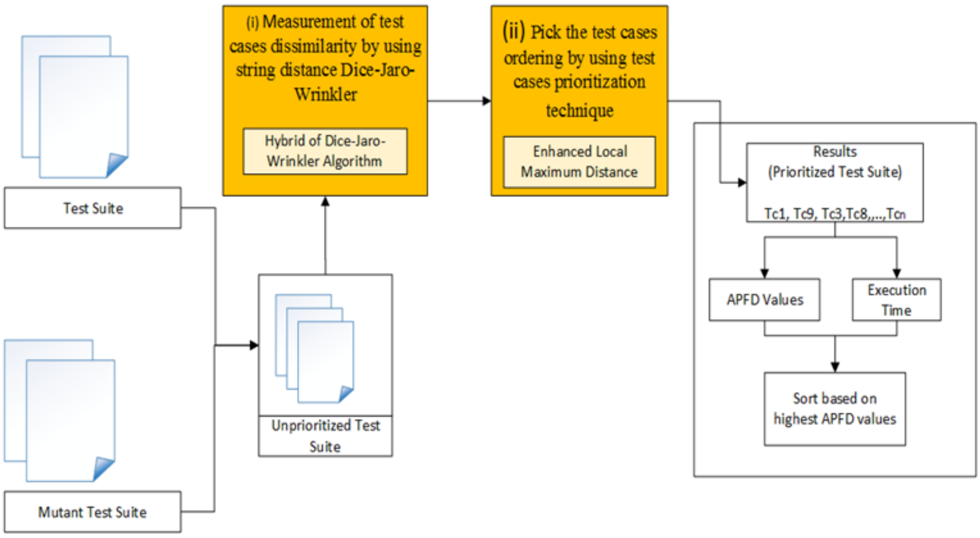

Test Suite
Mutant Test Suite
(i) Measurement of test cases dissimilarity by using string distance Dice-Jaro-Winkler
Hybrid of Dice-Jaro-Wrinkler Algorithm
Unprioritized Test Suite
(ii) Pick the test cases ordering by using test cases prioritization technique
Enhanced Local Maximum Distance
Results (Prioritized Test Suite)
Tc1, Tc9, Tc3,Tc8,...,Tcn
APFD Values
Execution Time
Sort based on highest APFD values

Fig. 1. The proposed approach

The dissimilar measure technique is based on the theory that most dissimilar test cases can accelerate higher coverage, thus faults can be detected as early as possible. The aim of the enhancement of the string distance technique is to improve the rate of fault detection with low execution time taken. The enhancement of the proposed string distance is based on the string distance by Sahak et al. [36]. This study has focused on enhancement of Jaro-Winkler with all-yes config algorithm by focusing on highest number of features as the one that will be selected in prioritize list. However, once the dissimilarity is implemented, it requires a technique that can focus on highest distance between two test cases rather than focus on the total maximum number of paths in test cases.

The Dice Similarity Measure is a one of the measurement techniques that can be used to measure similarity between test cases. The concept of Dice Similarity is based on the set-based distance generalized from the idea of Jaccard Distance formula. It is based on balance between the positive value and false positive values obtained from measurement. Similar to Jaro-Winkler, it gains the best results of similarity measure in software testing domain [37-39]. This study enhanced existing Dice Similarity of formula in (1) and Jaro-Winkler in (2) and (3).

$$d_n(T_1,T_2) = \frac{0.5|T_1 \cap T_2|}{|T_1 + T_2|} \tag{1}$$

$$JW(T_1,T_2)_j = Jaro(T_1,T_2) + \gamma p(1 - Jaro(T_1,T_2)djw) \tag{2}$$

$$Jaro(T_1, T_2)_{djw} = \frac{1}{3}\left(\frac{m}{T_1} + \frac{m}{T_2} + \frac{m-t}{m}\right) \tag{3}$$

As illustrated in Fig. 2, the proposed measurement technique involves two main components that consist of extensive usage of Dice Similarity Measure with small additional part of Jaro-Winkler equation. Existing Jaro-Winkler consists of integration between two equations of finding matching test cases. The combination with Dice Similarity helps to improve the selection of matching test cases. At first step of measurement, this study considered Jaro Distance equality of d_j that is used to calculate deselected test cases. Then, it will consider the Degree of Difference, *df* that is used to calculate the difference between two test cases based on string length. The degree of difference is proposed by Tumeng [39] that used *df* to replace transposed character in test cases. *df* values is used to replace ℓ in Winkler formula. The implementation of *df* in the equation helps to provide more accurate and consistent similarity values. Based on the proposed string distance technique, next subsection will discuss the example of calculation to measure dissimilarity test cases.

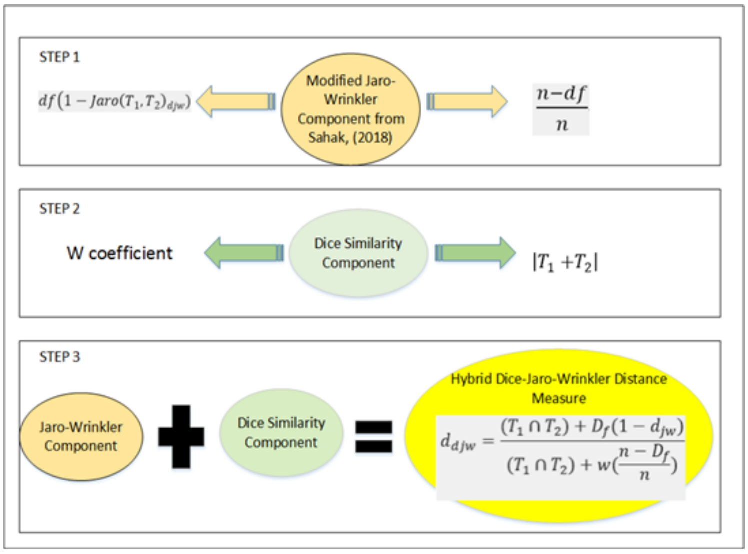

STEP 1
$df(1 - Jaro(T_1, T_2)_{djw})$
Modified Jaro-Wrinkler Component from Sahak, (2018)
$\frac{n-df}{n}$
STEP 2
W coefficient
Dice Similarity Component
$|T_1 + T_2|$
STEP 3
Jaro-Wrinkler Component
Dice Similarity Component
Hybrid Dice-Jaro-Wrinkler Distance Measure
$d_{djw} = \frac{(T_1 \cap T_2) + D_f(1 - d_{jw})}{(T_1 \cap T_2) + w(\frac{n - D_f}{n})}$

Fig. 2. The Proposed LM-LMD with Dice-Jaro-Winkler Dissimilarity

In addition, this study used n parameter described by Sahak [36]. *n* parameter is taken from Hamming Distance equation that considers calculation of deselected state between two test cases. The results also show good improvement in terms of highest APFD value. The result of existing study motivates this study to further enhance the proposed equation. The process of formula enhancement starts with deselected states between two test cases based on Hamming Distance equation. Take out $\frac{S}{T1} \cap \frac{S}{T2}$ based on Hamming Distance equation. The total of deselected feature is represented as *n* value.

$$Jaro(T_1, T_2)_{djw} = \frac{1}{3}\left(\frac{m}{T_1} + \frac{m}{T_2} + \frac{n-df}{n}\right) \tag{4}$$

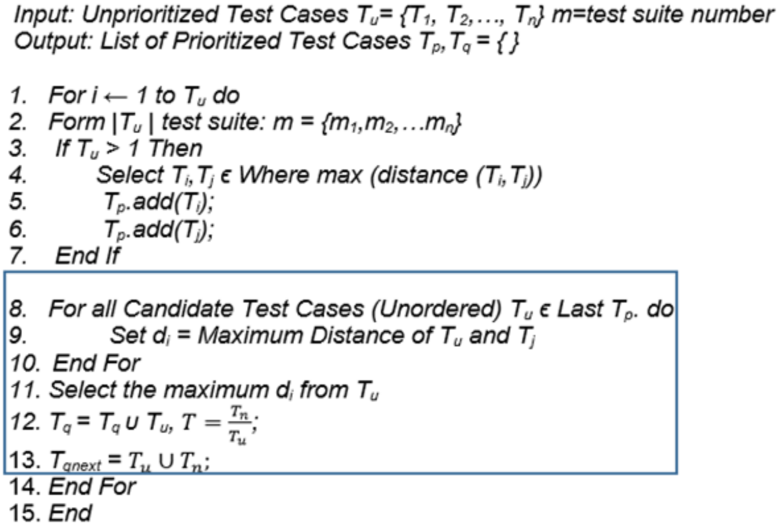
$$JW(T_1, T_2)_j = Jaro(T_1, T_2) + df(1 - Jaro(T_1, T_2)djw) \tag{5}$$

$$d_{djw} = 1 - \left(\frac{(T_1 \cap T_2) + df(1 - d_{jw})}{(T_1 \cap T_2) + w\left(\frac{n - df}{n}\right)} \right) \tag{6}$$

Then, the original Jaro is transformed into modified Jaro with d_j value that have been proposed by Tumeng, (2017). n value is selected and inserted into Jaro formula. Equation (4) presented the modified Jaro. Then, Modified Jaro component is split into four parts, which are $\frac{1}{3}, \frac{m}{T_1}, \frac{m}{T_2}$, and $\frac{n-d_f}{n}$. The original Winkler component γρ is converted into modified Winkler *df*. The modified Winkler is stated in (5). Next, the Dice Similarity formula is split into three main parts, which are 2.0, $|T_1 \cap T_2|$ and $|T_1+T_2|$. The Modified Jaro Winkler component will be combined with Dice Similarity in order to calculate dissimilarity value. The final formula of this enhancement is defined in (6).

4.2 Phase (I): Dissimilarity Measure: Dice with Jaro-Winkler Algorithm

The proposed enhancement is to improve the selection of the next test cases. In order to select next test cases into prioritized list, the last test case in the prioritized list will be compared with test cases in the unordered list. The comparison is based on the minimum distance calculated from the list of unordered test cases. The maximum distance is taken as a measurement since it can detect fault as early as possible. The process will be repeated until all test cases have been inserted into prioritized list. The pseudocode of enhance Local Maximum Distance is discussed in **Fig. 3**.



```
Input: Unprioritized Test Cases $T_u$= {$T_1$, $T_2$,…, $T_n$} m=test suite number
Output: List of Prioritized Test Cases $T_p$,$T_q$ = { }

1.  For i ← 1 to $T_u$ do
2.  Form |$T_u$| test suite: m = {$m_1$,$m_2$,…$m_n$}
3.   If $T_u$ > 1 Then
4.        Select $T_i$,$T_j$ ϵ Where max (distance ($T_i$,$T_j$))
5.         $T_p$.add($T_i$);
6.         $T_p$.add($T_j$);
7.   End If
8.  For all Candidate Test Cases (Unordered) $T_u$ ϵ Last $T_p$. do
9.          Set $d_i$ = Maximum Distance of $T_u$ and $T_j$
10. End For
11. Select the maximum $d_i$ from $T_u$
12. $T_q$ = $T_q$ ∪ $T_u$, $T = \frac{T_n}{T_u}$;
13. $T_{qnext}$ = $T_u \cup T_n$;
14. End For
15. End
```

Fig. 3. The Proposed LM-LMD with Dice-Jaro-Winkler Dissimilarity

Line 1, *i←1* to T_u is used to represent the list of prioritized test cases. Line 2 is used to form test cases $|T_u|$ based on maximum distance. Then *if else* condition in Line 3 to 7 is used to select the distance of test cases. It will be added into the list based on the maximum distance measure. Then, after the first two test cases have been included in prioritized list, it will proceed with the pseudocode in the shaded blue box. Line 8 is used to measure distance of unordered list T_u. Line 9, d_i is to evaluate distance between test case of T_u and T_j. Line 11 is to get the maximum distance of d_i. In Line 12, T_q=$T_q \cup T_u$ is used to get next distance measure between the current last prioritize test case with the unprioritized list. Then, $T = T_n/T_u$ is used to get the total of test cases in the unprioritized list.

The box shows the enhancement of the Local Maximum Distance algorithm. Here, the enhancement is conducted due to lack of measurements of dissimilarity distance between the maximum value of the last prioritized test cases with the list of unprioritized ordered. The comparison between the last test cases in prioritized order with unprioritized can help to ensure the prioritization process time can be improved based on the balance of total test cases in unordered list.

5. Empirical Study Design

The goals of the experiment evaluation are: (1) to investigate how LM-LMD with Dice-Jaro-Winkler Dissimilarity measure fares against existing TCP approach in SPL; (2) to investigate LM-LMD with Dice-Jaro-Winkler based on cost and effectiveness measure for e-shop test object. In line with the goals defined above, we focus on answering the following research questions:
RQ1: How well can the LM-LMD with Dice-Jaro-Winkler prioritize test cases for MBT test model in SPL domain context?
RQ2: In what ways can the LM-LMD with Dice-Jaro-Winkler improve existing TCP in SPL domain context?

5.1 Empirical Study Setup

We implemented our work by using Netbeans of Java programming language and there is no standard framework used in the implementation. So, it can be concluded that the experiment is run fairly for all existing TCP approaches in SPL. We run our experiments based on Windows 8.1 Intel Core i5 2.3Ghz 4GB RAM. For statistical significance, we have executed LM-LMD with Dice-Jaro-Winkler for 20 times to get the result trend of proposed approach. For RQ1 and RQ2, the test case from e-shop test object generated from SPL MBT test case generator is taken as benchmark test case. For RQ2, the comparison between existing TCP approach is considered. Existing TCP approach in SPL proposed by Henard [35] is taken as comparison. In order to ensure a fair comparison, we re-implement existing TCP approaches by using the same settings and programming language similar to our proposed approach. In order to evaluate the performance of proposed prioritization algorithm, five types of mutant version test cases are created based on test objects as shown in **Table 1**. Each type of mutant version will be compared with original version in order to evaluate faults.

6. Example

As an example, the e-shop test object based on statechart is illustrated in **Fig. 4**. Statechart consists of 31 states and 44 transitions, and includes the behavior process of educational robotics learning tools to enhance students experience through hands-on nature with implementation of the robotics technology in learning process. Based on e-shop test scenario, the model-based testing test case generator will be used to generate the set of test cases. With consideration of existing constraints, test case is generated.

Table 1. Mutant version for test model

Mutant Version	Description
V1	Add one new state
V2	Remove one transition
V3	Change start state

7. Experimental Result

7.1 Answering RQ1: How well can the LM-LMD with Dice-Jaro-Winkler prioritize test cases for MBT test model in SPL domain context?

This section provided an analysis result of the proposed TCP approach data collected based on test case from e-shop test object. First, dissimilarity distance based on the proposed measurement, which is LM-LMD with Dice-Jaro-Winkler Dissimilarity results, is obtained. This distance result will be used in the proposed prioritization algorithm in order to reorder test cases based on the maximum dissimilarity results. Table 2 show results of dissimilarity distance measure of LM-LMD with Dice- Jaro-Winkler Dissimilarity based on e-shop test case objects. 15 test cases generated from test model artefacts are used as indicator to represent TCP. The matric form is constructed in order to evaluate dissimilarity distance results by comparing each test case. It is interesting to note that dissimilarity measure score for some test cases is less than the others. But, for some comparison of test cases, it shows that dissimilarity is greater than 0.5. This means that for some test case comparisons, the dissimilarity measure is higher.

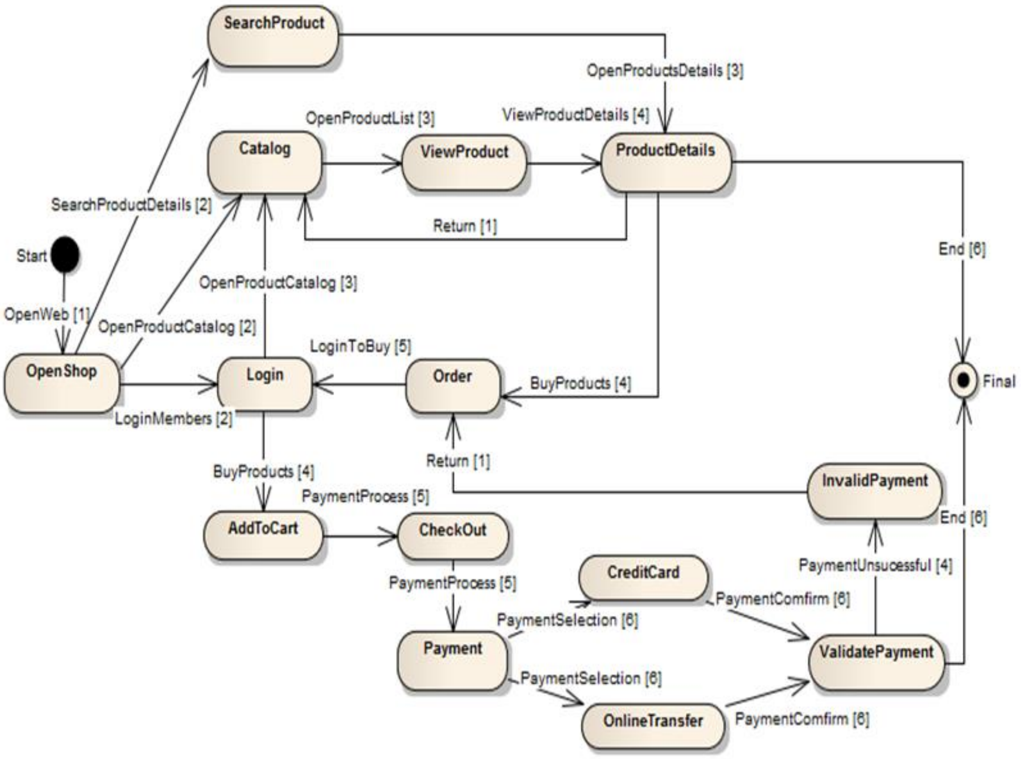


Fig. 4. e-shop test object

After dissimilarity of test cases have been measured, the results of dissimilarity will be used to reorder position of test cases. Here, the proposed LM-LMD with Dice-Jaro-Winkler Dissimilarity based on enhanced Local Maximum Distance based on the maximum dissimilarity measure results will be used to find the maximum dissimilarity. The reason that the proposed algorithm is used to evaluate based on maximum dissimilarity is that the greater the dissimilarity between two test cases, the more different they are from each other. The test cases with higher similarity will be located in the bottom of the test cases.

This measurement is based on 20 executions to evaluate faults. **Table 2** represents the results of average APFD for five mutant versions. As per result, the average APFD obtained is 0.907. This result is expected since the nature of e-shop object consists of large size of test cases and there are many looping transitions with substates in the test model. Then, the prioritization algorithm proposed is analyzed based on five versions of test case mutants.

Table 2. Mutant version for test model

APFD	V1	V2	V3	V4	V5	Average Values
Mean Value	0.902	0.955	0.906	0.945	0.986	0.939
Std. Deviation Value	0.066	0.164	0.034	0.998	0.091	0.271
Maximum Value	1.074	1.420	0.950	1.080	1.180	1.141
Minimum Value	0.820	0.880	0.850	0.900	0.920	0.874
Time (s)	0.902	0.955	0.906	0.945	0.986	0.939

Table 3 shows the average of mutant killed at 20%, 40%, 60%, 80% and 100% level across e-shop test objects. The rate of fault detected formulation is based on Tumeng [39]. Results shows that the proposed prioritization algorithm based on Dice-Jaro-Winkler dissimilarity is able to compete competitively in order to increase rate of fault detected. It shows that the proposed approach can detect 100% of faults in only 80% of mutants.

Table 3. Summary Rate of Fault Detected

Mutant Rate	Fault Rate
20	24
40	52
60	62
80	100
100	100

7.2 Answering RQ2: In what ways can the LM-LMD with Dice-Jaro-Winkler improve existing TCP in SPL domain context?

In order to answer RQ2, the comparison between the proposed algorithms with previous studies that implement test cases prioritization in SPL are evaluated. Study by Henard et al. [35] and Sahak [36] are used as a comparison to evaluate results of prioritization technique. There are two existing prioritization algorithms proposed by Henard et al.[35], which are Local Maximum Distance and Global Maximum Distance. In terms of dissimilarity measure, this proposed technique used Jaccard Distance to evaluate distance between two test cases. Sahak [36] proposed enhanced all-yes configuration by measured dissimilarity by using hybrid between Hamming and Jaro-Winkler distance. Performance comparison is based on two types of metrics, which are APFD values and average execution time.

In **Fig. 5** is shown the analysis comparison of average APFD values for e-shop test object. The results are evaluated based on average prioritization based on five mutant versions by using different prioritization algorithm. As per result, the proposed approach outperformed other approaches in e-shop test object. It shows improvement in the results with 0.937 improved followed by enhanced all-yes config with 0.815, Local Maximum Distance 0.721 and 0.833 for Global Maximum Distance. It can be overall concluded that the APFD values obtained from the proposed algorithm can improve the fault detection rate compared with previous studies. The proposed algorithm can produce the good result of APFD for e-shop object.

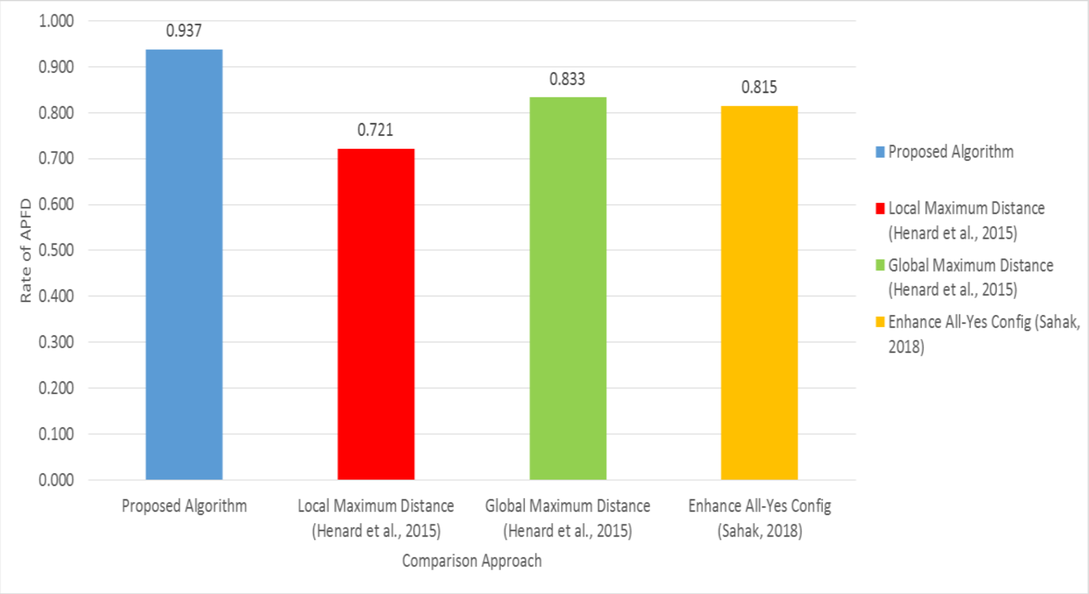


Fig. 5. Comparison Results of Proposed TCP Approach against Existing Approach based on average APFD

This showed the proposed string distance based on dissimilarity measure is effective in discovering faults on average. This is due to the capability of the Dice-Jaro-Winkler to diversify the maximum dissimilarity distance of test cases. Thus, more defects can be revealed as soon as possible.The improvement of Local Maximum Distance in the proposed algorithm that measures the next prioritization list by considering maximum distance from last test cases in prioritized order with unprioritized list can maximize fault detection based on highest dissimilarity values.

Another comparison is based on testing cost. Here, cost is evaluated based on total execution time. **Fig. 6** shows analysis of average execution time of fault detection measure for APFD values. Results showed that the proposed algorithm takes most minimal average execution time with 0.312 seconds followed by 0.650, 0.980 and 1.377 for Local Maximum Distance, Enhanced All-yes Config and Global Maximum Distance, respectively. It can be concluded that the proposed algorithm reflects the best average execution time. The proposed LM-LMD with Dice-Jaro-Winkler Dissimilarity is able to achieve faster prioritization time. This is because the string distance proposed is directly focused on dissimilarity string distance value. It allows the proposed enhanced Local Maximum Distance to be able to look for the highest dissimilarity. Furthermore, the enhanced Local Maximum Distance process that covers the maximum value between last test cases in the prioritize ordered with unprioritized order helps to reduce the measurement process in unprioritized list. This leads to improve prioritization time for the proposed prioritization algorithm.

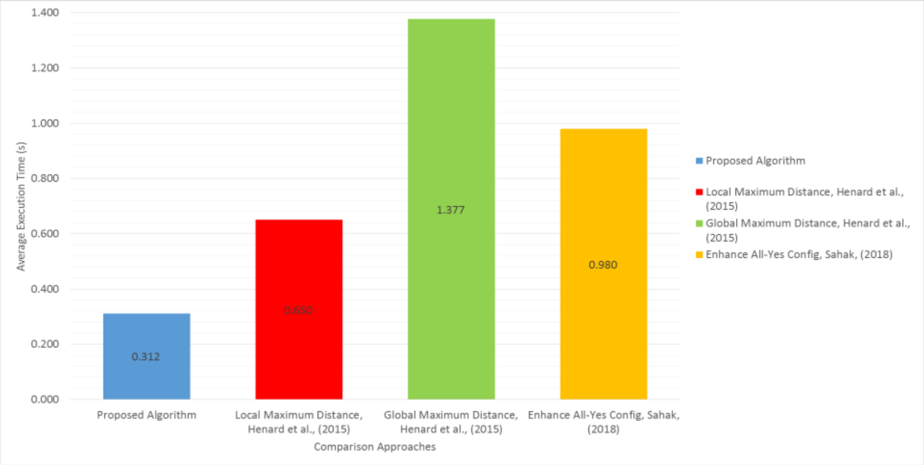


Fig. 6. Comparison results of proposed TCP approach against existing approach based on average execution time

7.3 Statistical Analysis

We conduct our statistical analysis for all measures in **Fig. 5** and **Fig. 6**. This evaluation is conducted based on Kruskal Wallis used as a non-parametric test. The measurement is based on the median of the population as some occasions are different from others. This testing process is based on detailed hypothesis testing as defined in **Table 4**. The Kruskal Wallis test states that for null hypothesis, all strategies are equivalent while a rejection of this hypothesis implies there are significance differences among approaches performance. The results analysis is based on 20 times of execution against all goal measurements defined.

Using Kruskal-Wallis, the average mean value of each validation metric is presented in **Table 5**. The mean rank of each validation metric is presented. Average prioritization time reflects cost as the lowest values show the best result, whereas for the effective measure, average APFD shows the highest values reflect the good results. In summary, the Kruskal-Wallis test presents a statistically significant difference between comparison types with validation metric of p = 0.000. The p-value shows an average statistical significance of (0.000 < 0.05). Therefore, the results show that the comparison types against all validation metrics is statistically significant. However, there is no clue as to which comparison type is statistically significant to others. Thus, it requires investigation on the comparison types by using mean rank post hoc test.

Table 4. Kruskal Wallis Hypothesis Summary

Null Hypothesis	**The distribution of AVG prioritization time and AVG APFD is the same across categories of comparison types.**
Asymptotic Significance	0.000
Significance Level	0.05

Based on the mean values obtained, the Kruskal-Wallis Post Hoc analysis was conducted in order to evaluate the statistical significance of validation metrics. The post-hoc hypothesis are defined in **Table 6** as basement to evaluate the comparison types. **Table 8** shows the results of post-hoc analysis. Values will reflect statistical difference once results of p-values is less than 0.05; thus the null hypothesis can be rejected. Whereas if the p-values is more than 0.05, then the null hypothesis will be accepted.

Table 5. Kruskal-Wallis Test Mean Ranks

Null Hypothesis	Average APFD (%)	Average Prioritization Time(s)	p-value	Hypothesis (Reject / Accept)
LM-LMD with Dice-Jaro-Winkler	0.927	0.233	0.00	Reject
Local Maximum Distance	0.733	0.745	0.00	Reject
Global Maximum Distance	0.832	1.243	0.00	Reject
Enhance All-Yes Config	0.810	0.912	0.00	Reject

By using Kruskal-Wallis mean rank post hoc analysis, three validation metrics are compared. For three validation metrics defined, p-values for all validation metrics are statistically different. Thus, it shows that for average APFD value, hypothesis${}_0^1H$ is accepted and for average prioritization value, hypothesis ${}_0^2H$ is accepted. In conclusion, the proposed LM-LMD with Dice-Jaro-Winkler is analyzed and compared with existing approach. By comparing results, it implies that the proposed LM-LMD with Dice-Jaro-Winkler is highly and significantly improved for both validation metrics.

Table 6. Kruskal-Wallis with Post Hoc hypothesis

Null Hypothesis	Descriptions
${}_0^1H$	There is no significant difference of execution time to prioritized test suite values between the test cases generation and prioritization approach
${}_0^2H$	There is no significant difference of rate of fault in APFD values between the test cases generation and prioritization approach

8. Discussion

Based on work undertaken, some observation can be elaborated as lessons learned. The observations are divided into two parts. The first is the design and advantages of LM-LMD with Dice-Jaro-Winkler, whereas the second part is related to the performance of LM-LMD with Dice-Jaro-Winkler.

Concerning the first part, LM-LMD with Dice-Jaro-Winkler consists of integration between LM-LMD as prioritization algorithm and Dice-Jaro-Winkler used to measure distance between test cases. Integration between Dice and Jaro-Winkler helps to produce more intense dissimilarity values. This is because integration considers both aspects of similar and dissimilar test case sequences. The LM-LMD with Dice-Jaro-Winkler is used in order to evaluate the distance between test cases and to reorder test case ranking in test suite. Here, the proposed Dice-Jaro-Winkler string distance is able to reduce distance value measurement since the dissimilarity of test cases is calculated in this phase. For LM-LMD, it will validate faults based on values produced from string distance without any additional formulation to evaluate dissimilarity between two test cases. The LM-LMD helps to detect higher faults by considering maximum distance of last test cases in prioritized list with unordered distance of test cases. Implementing this measurement can improve the existing SPL prioritization approach in fault detection process based on dissimilarity measure. The lack of diversity in existing TCP approach for SPL can be improved by enhanced algorithm and measures distances between the last test cases in prioritized order with maximum dissimilarity distance.

Table 8. Kruskal Wallis with Post Hoc Analysis

Comparison	**Validation Metrics**	**Test Statistics**	**Std. Error**	**p-value**	**Conclusion Significant p-value = (<0.05)**	**Validation Metrics (Accept/ Reject)**
LM-LMD with Dice-Jaro-Winkler vs Local Maximum Distance	Average APFD (%)	64.921	0.1805	0.00	Significant	Reject ${}_0^1H$
	Average Prioritization Time(s)	104.11	0.0397	0.00	Significant	Reject ${}_0^2H$
LM-LMD with Dice-Jaro-Winkler vs Global Maximum Distance	Average APFD (%)	64.921	0.1805	0.00	Significant	Reject ${}_0^1H$
	Average Prioritization Time(s)	104.11	0.0397	0.00	Significant	Reject ${}_0^2H$
LM-LMD with Dice-Jaro-Winkler vs Enhance All-Yes Config	Average APFD (%)	64.921	0.1805	0.00	Significant	Reject ${}_0^1H$
	Average Prioritization Time(s)	104.11	0.0397	0.00	Significant	Reject ${}_0^2H$

For the second part, the results show that the proposed prioritization algorithm outperforms existing approaches based on effectiveness (APFD) and cost (prioritization time). One interesting point is the proposed algorithm can maximize effectiveness and minimize the cost of testing in MBT for SPL. A comparison of the results shows that the proposed LM-LMD with Dice-Jaro-Winkler outperformed existing studies that can produce a good fault detection rate with minimal prioritization time. This helps to give algorithm more diversity in searching process for maximum distance to reorder test cases. The proposed approach can minimize cost of testing without requiring large computational memory to execute the process faced by existing approaches. By having the minimal prioritization time and maximal APFD rate, it reflects that the proposed approach can improve existing result by having cost effective TCP in MBT for SPL. In summary, a good quality of test cases refers to test cases that can balance trade-off between cost and effectiveness. This study shown that TCP in MBT for SPL can handle multi-objective optimization problems with cost and effectiveness measure. By having a minimal test case prioritization time and maximal APFD rate, it shows that the proposed TCP approach can help to handle optimization problem in terms of cost and effectiveness measure.

9. Threat to Validity

Empirical and experimental studies often encounter threat to validity. Effort has been undertaken to minimize the threats. In the context of this research, several threats have been identified.

1. Internal Validity. Internal validity in this study is related to the configuration setting for proposed and existing approaches. To handle this threat, we utilized the similar default setting for all approaches. Similar stop criteria, initial default parameters and maximum populations is similar for traditional and the proposed LLH. Furthermore, in each LLH and traditional SBT implemented, specific parameters are required, and similar tune value is implemented based on values defined in existing literature.

2. Construct Validity. Construct validity involves measurement metrics for average APFD and average prioritization time. In order to ensure fairness, this study used similar evaluation formula to evaluate the results of test cases. In order to have trends of evaluation, the execution is repeated 20 times.

10. Conclusion and Future Work

We present LM-LMD with Dice-Jaro-Winkler, a TCP approach with measurement based on maximal distance of dissimilarity measure for SPL. We used e-shop, a medium scale of test object for SPL domain context to evaluate the result. The evaluation shows that the LM-LMD with Dice-Jaro-Winkler is efficient in terms of average prioritization time and average APFD rate. A good quality of test cases refers to a test case that has balanced trade-off between cost and effectiveness measure. This study has shown that TCP in MBT for SPL can handle multi-objective optimization problems with cost and effectiveness measure. By having a minimal test case prioritization time and maximal APFD rate, it has been shown that the proposed TCP approach can help to handle optimization problems in terms of balance trade-off between cost and effectiveness measure. As our finding has been promising, we plan to improve our work further to validate this work based on real test object. In addition, the incorporation of different techniques in similarity-based prioritization for SPL can be further improved to evaluate fault for statechart test model artefacts. Furthermore, more empirical study can be conducted to evaluate the similarity-based prioritization based on the current approach.

Acknowledgement

This research is fully funded by Ministry of Higher Education Malaysia (MOHE) for FRGS Grant Vot No.5F117 and University Teknologi Malaysia for UTM-TDR Grant Vot No.06G23, which made this research endeavor possible.

References

[1] A. Reuys, E. Kamsties, K. Pohl, and S. Reis, "Model-Based System Testing of Software Product Families," in *Proc. of International Conference Advanced Information Systems Engineering*, pp. 519-534, 2010. Article (CrossRef Link)

[2] R. Kazmi, D. N. A. Jawawi, R. Mohamad, and I. Ghani, "Effective Regression Test Case Selection," *ACM Computing Surveys*, vol. 50, no. 2, pp. 1-32, 2017. Article (CrossRef Link)

[3] A. Arrieta, G. Sagardui, and L. Etxeberria, "A model-based testing methodology for the systematic validation of highly configurable cyber-physical systems," in *Proc. of the 6th International Conference on Advances in System Testing and Validation Lifecycle*, pp. 66-72, 2014. Aritcle (CrossRef Link)

[4] A. Egyed, S. Segura, R. E. Lopez-Herrejon, A. Ruiz-Cortés, J. A. Parejo, and A. B. Sánchez, "Multi-objective test case prioritization in highly configurable systems: A case study," *Journal of Systmes and Software*, vol. 122, pp. 287-310, 2016. Article (CrossRef Link)

[5] C. Henard, M. Papadakis, G. Perrouin, J. Klein, P. Heymans, and Y. Le Traon, "Bypassing the Combinatorial Explosion : Using Similarity to Generate and Prioritize T-wise Test Configurations for Software Product Lines," *IEEE Transactions on Software Engineering*, vol. 40, no. 7, pp. 650-670, 2016. Article (CrossRef Link)

[6] R. A. Sulaiman and D. N. A. Jawawi, "Derivation of Test Cases for Model-based Testing of Software Product Line with Hybrid Heuristic Approach," in *Proc. of International Conference of Reliable Information and Communication Technology*, vol. 1073, pp. 199-208. Article (CrossRef Link)

[7] B. Utting, M. Pretschner, and A. Legeard, "A taxonomy of model-based testing approaches," *Software: Testing, Verification and Reliablity*, vol. 22, no. 5, pp. 291-312, 2012. Article (CrossRef Link)

[8] A. Abbas, I. F. Siddiqui, and S. U. Lee, "Multi-Objective Optimization of Feature Model in Software Product Line : Perspectives and Challenges," *Indian Journal of Science and Technology*, vol. 9, no. 45, pp. 1-7, Dec. 2016. Article (CrossRef Link)

[9] S. A. Ajila and A. B. Kaba, "Using Traceability Mechanisms to Support Software Product Line Evolution," in *Proc. of International Conference on Information Reuse and Integration*, pp. 157-162, 2004. Article (CrossRef Link)

[10] F. Heidenreich, J. Kopcsek, and C. Wende, "FeatureMapper: mapping features to models," in *Proc. of Companion of the 30th International Conference on Software Engineering*, vol. 30, pp. 943-944, 2008. Article (CrossRef Link)

[11] K. Czarnecki and M. Antkiewicz, "Mapping Features to Models : A Template Approach Based on Superimposed Variants Background : Feature Modeling," in *Proc. of the 4th International Confernce on Generative Programming and Component Engineering*, pp. 422-437, 2005. Article (CrossRef Link)

[12] L. Shen, X. Peng, and W. Zhao, "A Comprehensive Feature-Oriented Traceability Model for Software Product Line Development," in *Proc. of ECMDA Traceability Workshop Proceedings*, pp. 77-86, 2008. Article (CrossRef Link)

[13] M. S. H. Lackner and M. Schmidet, "Towards the Assessment of Software Product Line Tests : A Mutation System for Variable Systems," in *Proc. of the 18th International Software Product Line Conference*, pp. 62-69, 2014. Article (CrossRef Link)

[14] G. Perrouin, Y. Traon, B. Baudry, S. Sen, J. Klein, and S. Oster, "Pairwise testing for software product lines: comparison of two approaches," *Software Quality Journal*, vol. 20, pp. 605-643, 2012. Article (CrossRef Link)

[15] S. Lity, M. Al-Hajjaji, T. Thüm, and I. Schaefer, "Optimizing product orders using graph algorithms for improving incremental product-line analysis," in *Proc. of the 11th International Workshop on Variability Modelling of Software-intensive Systems*, pp. 60-67, 2017. Article (CrossRef Link)

[16] X. Devroey, G. Perrouin, and P. Y. Schobbens, "Abstract test case generation for behavioural testing of software product lines," in *Proc. of the 18th International Software Product Line Conference: Companion Volume for Workshops*, vol 2, pp. 86-93, 2014. Article (CrossRef Link)

[17] I. Machado, "Fault Model-Based Variability Testing," Ph. D. Thesis, Universidade Salvador, Brazil, 2014. Article (CrossRef Link)

[18] S. Weissleder, F. Wartenberg, and H. Lackner, "Testing Software and Systems," in *Proc. of IFIP International Confernece on Testing Software and Systems*, vol. 9447, pp. 86-101, 2015. Article (CrossRef Link)

[19] F. Ensan, E. Bagheri, and D. Gasevic, "Evolutionary Search-based Test Generation for Software Product Line Feature Models," in *Proc. of International Conference on Advanced Information Systems Engineering*, pp. 613-628, 2012. Article (CrossRef Link)

[20] C. Henard, M. Papadakis, G. Perrouin, J. Klein, and Y. L. Traon, "Multi-objective Test Generation for Software Product Lines," in *Proc. of the 17th International Software Product Line Conference*, pp. 62-71, 2013. Article (CrossRef Link)

[21] H. Hemmati, A. Arcuri, and L. Briand, "Reducing the Cost of Model-Based Testing through Test Case Diversity," in *Proc. of IFIP International Confernece Federation for Information Processing*, vol. 6435, no. 2, pp. 63-78, 2010. Article (CrossRef Link)

[22] L. Inozemtseva and R. Holmes, "Coverage Is Not Strongly Correlated with Test Suite Effectiveness," in *Proc. of the 36th International Conference on Software Engineering*, pp. 435-445, 2014. Article (CrossRef Link)

[23] S. Yoo and M. Harman, "Regression testing minimization, selection and prioritization : a survey," *Software Testing Minimization, Selection and Prioritization: a survey*, vol. 22.2, pp. 67-120, 2012. Article (CrossRef Link)

[24] S. B. Lity, "Model-Based Product-Line Regression Testing of Variants and Versions of Variants," *Technische Universitat Braunschweig*, 2019. Article (CrossRef Link)

[25] M. Al-Hajjaji, S. Lity, R. Lachmann, T. Thüm, I. Schaefer, and G. Saake, "Delta-Oriented Product Prioritization for Similarity-Based Product-Line Testing," in *Proc. of IEEE/ACM 2nd International Workshop on Variablity and Complexity in Software Design*, pp. 34-40, 2017. Article (CrossRef Link)

[26] R. Kazmi, D. N. A. Jawawi, R. Mohamad, and I. Ghani, "Effective Regression Test Case Selection Technique using Weighted Average Scoring," *ACM Computing Survery*, vol. 50, no. 2, 2017. Article (CrossRef Link)

[27] C. Henard, M. Papadakis, G. Perrouin, J. Klein, and Y. Le Traon, "Assessing Software Product Line Testing via Model-based Mutation: An Application to Similarity Testing," in *Proc. of the 6th International Conference on Software Testing, Verification and Validation Workshops*, pp. 188-197, 2013. Article (CrossRef Link)

[28] S. Wang, D. Buchmann, S. Ali, A. Gotlieb, D. Pradhan, and M. Liaaen, "Multi-Objective Test Prioritization in Software Product Line Testing : An Industrial Case Study," in *Proc. of the 18th International Software Product Line Confernece*, vol. 1, pp. 32-41, 2014. Article (CrossRef Link)

[29] I. Schaefer, M. Al-Hajjaji, R. Lachmann, F. Fürchtegott, and S. Lity, "Fine-grained test case prioritization for integration testing of delta-oriented software product lines," in *Proc. of the 7th International workshop on Feature-Oriented Software*, vol. 1, no. 212, pp. 1-10, 2016. Article (CrossRef Link)

[30] S. Wang, S. Ali, and A. Gotlieb, "Minimizing Test Suites in Software Product Lines Using Weight-based Genetic Algorithms," in *Proc. of the 15th Annual Conference on Genetic and Evolutionary Computation*, pp. 1493-1500, 2013. Article (CrossRef Link)

[31] X. Devroey, "Behavioural Model Based Testing of Software Product Lines," Ph. D. Thesis, University of Namur, Belgium, 2014. Article (CrossRef Link)

[32] X. Devroey, G. Perrouin, M. Cordy, H. Samih, A. Legay, and P. S. Patrick, "Statistical prioritization for software product line testing : an experience report," *Software and Systems Modeling*, vol. 16, no. 1, pp. 153-171, 2017. Article (CrossRef Link)

[33] M. Sahak, S. A. Halim, D. N. A. Jawawi, and M. A. Isa, "Evaluation of Software Product Line Test Case Prioritization Technique," *International Journal on Advanced Science*, vol. 7, no. 4-2, pp. 1601-1608, 2017. Article (CrossRef Link)

[34] X. Devroey, G. Perrouin, A. Legay, P. Heymans, and P. Heymans, "Dissimilar Test Case Selection for Behavioural Software Product Line Testing," 2017. Article (CrossRef Link)

[35] C. Henard, "Enabling Testing of Large Scale Highly Configurable Systems with Search-based Software Engineering : The Case of Model-based Software Product Lines" Ph. D. Dissertation, Universite Du Luxemborg, 2015. Article (CrossRef Link)

[36] M. Sahak, "Effective similarity based test case prioritization technique for software product lines," Universiti Teknologi Malaysia, 2018. Article (CrossRef Link)

[37] A. Emília and V. Barbosa, "Similarity-based test suite reduction in the context of Model-Based Testing," *Universidade Federal de Campina Grande*, 2015. Article (CrossRef Link)

[38] S. Halim, D. N. Jawawi, and M. Sahak, "Similarity Distance Measure and Prioritization," *Journal of Informaion and Communication Technology*, vol. 18, no. 1, pp. 57-75, 2019. Article (CrossRef Link)

[39] R. A. Tumeng, "Test case prioritization with requirements change using string metrics," Universiti Teknologi Malaysia, 2017.

Rabatul Aduni Sulaiman received PhD degree in computer science from Universiti Teknologi Malaysia, Johor, Malaysia. Her master degree in computer science (database technology) and bachelor degree in computer science (database management) from Universiti Teknikal Malaysia Melaka (UTeM), Melaka, Malaysia. She was a system engineer at Infineon Technologies Malaysia before she joined the Department of Software Engineering at Universiti Tun Hussein Onn. Currently, her research focuses on software engineering, software testing, artificial intelligence techniques.

Dayang N. A. Jawawi is a Professor in School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia (UTM). She received her B.Sc. in Software Engineering from Sheffield Hallam University, UK, and her M.Sc. and Ph.D. in the field of Software Engineering from UTM. Her research interests cover software reuse, software quality, software testing and computing and software engineering education. A major part of her research projects focuses on rehabilitation and mobile robotics, healthcare application, real-time embedded systems and precision farming applications.

Shahliza Abd Halim received the B.Sc., M.Sc., and Ph.D. degrees from Universiti Teknologi Malaysia, Johor Bahru, Malaysia, in 1997, 1998, and 2016, respectively. She is currently a Senior Lecturer at the same university, where she is currently involved in the Software Product Line and Engineering Education research area.