ADAPTIVE REGRESSION TESTING STRATEGIES FOR COST-EFFECTIVE REGRESSION TESTING

A Dissertation
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Amanda Jo Schwartz

In Partial Fulfillment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY

Major Department:
Computer Science

June 2013

Fargo, North Dakota

North Dakota State University

Graduate School

Title

Adaptive Regression Testing Strategies for Cost-Effective Regression Testing

By

Amanda Schwartz

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

DOCTOR OF PHILOSOPHY

SUPERVISORY COMMITTEE:

Dr. Hyunsook Do
Chair

Dr. Gursimran Walia

Dr. Simone Ludwig

Dr. Sean Sather-Wagstaff

Approved:

6/19/2013
Date

Dr. Brian Slator
Department Chair

ABSTRACT

Regression testing is an important part of the software development life-cycle. However, it is also very expensive. Many different techniques have been proposed for reducing the cost of regression testing. To date, much research has been performed comparing regression testing techniques, but very little research has been performed to aid practitioners and researchers in choosing the most cost-effective technique for a particular regression testing session. One recent study investigated this problem and proposed Adaptive Regression Testing (ART) strategies to aid practitioners in choosing the most cost-effective technique for a specific version of a software system. The results of this study showed that the techniques chosen by the ART strategy were more cost-effective than techniques that did not consider system lifetime and testing processes. This work has several limitations, however. First, it only considers one ART strategy. There are many other strategies which could be developed and studied that could be more cost-effective. Second, the ART strategy used the Analytical Hierarchy Process (AHP). The AHP method is subjective to the weights made by the decision maker. Also, the AHP method is very time consuming because it requires many pairwise comparisons. Pairwise comparisons also limit the scalability of the approach and are often found to be inconsistent. This work proposes three new ART strategies to address these limitations. One strategy utilizing the fuzzy AHP method is proposed to address imprecision in the judgment made by the decision maker. A second strategy utilizing a fuzzy expert system is proposed to reduce the time required by the decision maker, eliminate inconsistencies

due to pairwise comparisons, and increase scalability. A third strategy utilizing the Weighted Sum Model is proposed to study the performance of a simple, low cost strategy. Then, a series of empirical studies are performed to evaluate the new strategies. The results of the studies show that the strategies proposed in this work are more cost-effective than the strategy presented in the previous study.

ACKNOWLEDGMENTS

First of all, I would like to thank my advisor, Dr. Hyunsook Do, for all of the time she spent meeting with me, helping revise my paper, and providing me with advice on this journey. Her support and advice was invaluable in completing my dissertation.

I would also like to thank the National Science Foundation for supporting some of my work. Also, a thank you to each of my committee members Dr. Gursimran Walia, Dr. Simone Ludwig, and Dr. Sean Sather-Wagstaff for agreeing to serve on my committee.

And last, but not least, a special thank you to my husband, who without his support and encouragement, I would have never been able to do this.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Software maintenance is a large part of the software development lifecycle. Maintaining a software system includes many different tasks such as fixing defects, adding new features, or modifying the software to accommodate different environments. After the software system has been modified, it needs to be tested, to ensure the changes did not have any adverse effects on the previously validated code. Regression testing is the process of testing a modified software system to ensure its continued quality.

Regression testing is often performed by re-running existing tests from previous versions along with new tests to test new features. However, as software systems grow, the size of the test suite can become too large, making it too time-consuming and costly to run all of the tests. For example, one study [14], mentions a company that has a software product with a regression test suite containing over 30,000 test cases that requires over 1000 machine hours to execute. To ensure continued quality of the system, when maintenance is performed on the system, it needs to be tested. However, requiring 1000 hours to run all of the test cases is not likely a feasible option. This example shows how reducing the time, and ultimately then, the cost, required by regression testing sessions has considerable importance.

Many regression testing techniques (e.g. [10, 53, 81]) and maintenance approaches have been proposed to reduce the time and cost of regression testing. These techniques are often grouped into three categories: test case prioritization, test case selection, and test case minimization. Test case prioritization [53, 81] techniques reorder test cases to meet a certain goal. For example, one commonly used goal is to improve the rate of fault detection. To achieve a high rate of fault detection, the test cases are reordered to find the highest number of faults in the shortest amount of time. Test case selection techniques [28, 71] select a subset of test cases that

focus on testing the parts of the system that have changed. Test case minimization techniques [29, 39] seek to identify and eliminate obsolete or redundant test cases.

In order to evaluate the numerous proposed techniques, many empirical studies [16, 22, 51, 53, 63] have been performed. In early studies, evaluation of the techniques focused on very simple metrics such as the number of test cases in the test suite, the time required for testing, and the rate of fault detection. These evaluations are limited, however, because they do not consider costs associated with the regression testing techniques themselves. The costs related to applying the regression testing techniques should be considered to obtain a more practical cost-benefit analysis of the techniques.

To address this problem, recent empirical studies [14, 52, 73] began to include costs related to environment and testing factors (e.g. cost of test setup and cost of identifying obsolete tests). These studies have shown that the environment and testing factors affect the cost-effectiveness of the regression testing techniques. Further, the studies show that the cost-benefits differ based on the particular software release and different techniques are most cost-effective in different regression testing sessions. The technique which is most cost-effective for one version may not be the most cost-effective for every version of a software system. Therefore, there is no single regression testing technique that is the most cost-effective for every version of a software system.

Empirical studies which investigate factors that affect the cost-effectiveness of techniques have helped researchers and practitioners understand different factors that affect the cost-effectiveness of the techniques. However, very little research has been performed to aid researchers and practitioners in utilizing this knowledge to select the most cost-effective technique for a particular regression testing session. To address this problem, a recent study proposed Adaptive Regression Testing (ART) strategies [5] to help identify the most cost-effective regression testing technique for

each regression testing session. This work proposed and empirically studied the analytical hierarchy process (AHP) [55] as one ART strategy focusing on test case prioritization techniques. The results indicated that the prioritization techniques selected by the AHP method can be more cost-effective than those that do not consider system lifetime and testing processes.

Although this study showed promising results, there are several limitations with the study and the proposed strategy. The study is limited because only one strategy was studied and evaluated. There are many other decision making strategies which could be considered that have the potential to be even more cost-effective. There are also several limitations with the proposed strategy. The strategy used the AHP method, which has many disadvantages. First, the AHP method is frequently criticized for being subjective to the judgments made by the decision makers [60, 67, 75]. Thus, the results can be inaccurate if the decision makers are inexperienced or if they lack knowledge about the application domain. Further, the study only used one decision maker, so the results are dependent upon the judgments made by one individual. A second weakness of the AHP method is that the comparisons made by the decision maker during the pairwise comparison process are often inconsistent [7, 41]. Judgements made in one comparison often contradict judgements made in another comparison. A third weakness of the AHP method is that it is very time-consuming for the decision makers. Empirical studies have shown that decision makers prefer other methods because of the time required by the pairwise comparisons [3, 27]. A fourth limitation of the AHP method is the use of pairwise comparisons is not scalable. Because of the work required by the pairwise comparisons, there is a limit to the number of criteria and alternatives that can be considered [54]. To address these problems, other strategies need to be developed.

1.1. Goal of This Research

The hypothesis of this research is that by providing new ART strategies to researchers and practitioners that offer appropriate techniques by considering organizations' circumstances, testing environments, and maintenance activities, the costs of regression testing can be reduced.

1.2. Approaches to Meet This Research Goal

To achieve this research goal, new ART strategies are proposed. In particular, this research investigated three ART strategies. One strategy utilized the fuzzy AHP method to address the issue of the results from the AHP method being subjective to the judgments made by the decision maker. A second strategy used a fuzzy expert system to obtain the benefits of a strategy which does not require pairwise comparisons. A third strategy utilized the Weighted Sum Model (WSM) to investigate the effectiveness of a simple, low-cost strategy for ART.

In addition to proposing these strategies, three empirical studies were conducted to investigate whether the strategies did indeed provide cost-savings. The first study investigated the fuzzy AHP approach. The second one studied a fuzzy expert system for ART (FESART). The third study evaluated the WSM and performed a statistical analysis of each of the strategies proposed in this work.

1.3. Impact of This Research

This research has significant implications for researchers and practitioners by providing strategies to help choose a regression testing technique considering important testing and environment factors. These strategies will help reduce the cost of regression testing by seeking to choose the most cost-effective technique for each regression testing session considering the organizations' circumstances, testing environments, and maintenance activities. Further, each of the strategies presented in this work are empirically studied and a statistical analysis was performed to give data to researchers and practitioners to use to choose the most appropriate strategy for their testing needs.

1.4. Organization of Dissertation

The remainder of this dissertation is divided into six chapters. Chapter 2 presents background and related work in the areas of regression testing techniques with the focus on test case prioritization, empirical studies evaluating regression testing techniques, models for evaluating regression testing techniques, Adaptive Regression Testing (ART) strategies, and decision making strategies. Chapter 3 describes each of the ART strategies in more detail. Chapter 4 describes the experiment conducted to investigate the new ART strategy utilizing fuzzy AHP and presents the results of the study. Chapter 5 describes the experiment conducted to investigate the new ART strategy utilizing a fuzzy expert system and presents the results of the study. Chapter 6 presents the results of a study investigating the WSM as an ART strategy and the results of a statistical analysis that was performed to evaluate the ART strategies presented in this work. Chapter 7 provides conclusions and possible future work which could be conducted in this area.

2. BACKGROUND

This chapter provides background information and related work relevant to regression testing techniques (focusing on test case prioritization techniques), empirical studies conducted to evaluate regression testing techniques, cost-benefit models used for evaluating regression testing techniques, Adaptive Regression Testing (ART), and decision making strategies. The discussion of decision making strategies is limited to the methods which are directly related to this work.

2.1. Regression Testing Techniques

Regression testing is the process of testing modified software systems to validate that changes to the system did not adversely affect previously validated code. Regression testing is an expensive activity, and to reduce the costs associated with regression testing many different test case selection, minimization, and prioritization techniques have been proposed and evaluated in the literature. This work is most closely related to test case prioritization techniques, so the discussion is limited to test case prioritization techniques here.

Test case prioritization techniques aim to find the ideal ordering of test cases according to a specific goal. For example, one commonly used goal is to achieve a high rate of fault detection. Algorithms which aim to achieve a high rate of fault detection attempt to reorder the test cases so that the test cases which find the most faults are executed first. This way if testing is halted early, the maximal amount of faults can be found in the shortened time frame. Consider an example of a simple software system which has five test cases (A, B, C, D, and E) that uncover ten faults. Table 1 displays a mapping of which test cases uncover which faults.

Now consider executing the test suite in two different orders, the original order (A-B-C-D-E) and a new order, E-D-B-C-A. Due to time constraints, testing had to be halted after the third test case. Figure 1 demonstrates the rate of fault detection

Table 1. Test Case Fault Mapping

	1	2	3	4	5	6	7	8	9	10
A	X									
B	X	X								
C							X			
D								X	X	X
E			X	X	X	X				

for both test case orders. After the third test case using the original order, only three of the ten faults (faults 1, 2, and 7) would have been found. The new order, however, was able to detect nine of the ten faults (leaving only fault 7 not found) in the same amount of testing time.

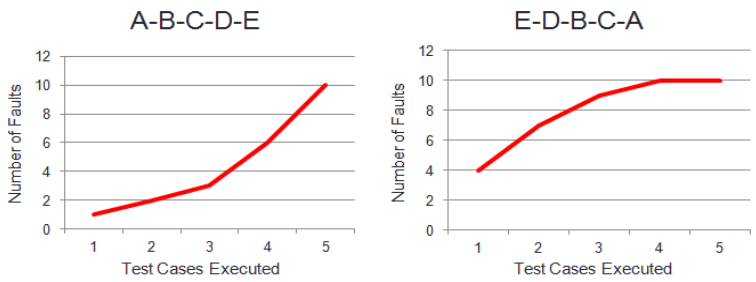


Figure 1. Rate of Fault Detection

This example shows how reordering test cases to find the maximal amount of faults early in the testing process has significant advantages. Many different techniques have been proposed to reorder test cases to achieve maximum benefit. Prioritization algorithms use various types of information, such as code coverage or code change information. For example, one technique, *total block coverage prioritization*, simply sorts the test cases by the order of the number of blocks they cover. One variation of this technique, *additional block coverage prioritization*, iteratively selects a test case that yields the greatest block coverage, adjusts the coverage information for the remaining test cases to indicate their coverage for the blocks not yet covered, and then repeats this process until all blocks are covered by at least one test case.

The idea of reordering test cases was first mentioned by Wong et al. [74]. In their work, the test cases reordered were already selected by a test case selection technique. The study noticed that even with a significant reduction in test size, different orders of test cases produced greater fault detection than larger test suites.

The concept of test case prioritization as its own regression testing technique was more formally defined by Rothermel et al. [53]. This study compared nine different test case prioritization techniques. Since its formal definition, many test case prioritization techniques have been created and empirically studied, and a recent survey by Yoo and Harman [77] provides an overview of these techniques. Since the survey, additional techniques continue to be proposed. For example, Zhang et al. [81] proposed a model to combine total and additional test case prioritization techniques that significantly outperformed the total and additional strategies in the study. Carlson et al. [10] implemented new prioritization techniques that incorporate a clustering approach using code coverage, code complexity, and history data on real faults, and Arafeen and Do [6] provide a test case prioritization technique using requirements-based clustering. In addition to being proposed and evaluated by researchers, prioritization techniques are also being used in practice by several software organizations [43, 64].

In order to investigate the effectiveness of prioritization techniques, empirical studies have been conducted to evaluate the proposed techniques. The next section of this chapter discusses empirical studies from the literature which evaluate the performance of prioritization techniques and investigate factors that affect the performance of techniques.

2.2. Empirical Studies

There have been many empirical studies performed to evaluate the cost and benefits of the proposed prioritization techniques (e.g. [16, 21, 22, 51, 53, 63]). These empirical studies showed promising results for the effectiveness of prioritizing test

cases. However, these studies used very simple metrics to evaluate the techniques. The most frequently used metric is the rate of fault detection. Evaluating techniques using simple metrics like the rate of fault detection ignore important costs related to the regression testing techniques themselves. Not accounting for those costs can lead to inaccurate evaluations of the technique's cost-effectiveness, and provide practitioners with incorrect data to use when choosing a regression testing technique.

To address this issue, recent research on test case prioritization has employed empirical studies to evaluate the cost-benefit trade-offs among techniques by considering various factors and testing contexts. Do et al. [14] studied the effects time constraints had on the cost-effectiveness of prioritization techniques. Results of another experiment [73] showed large trade-offs in the performance of regression testing techniques under fixed time periods. Elbaum et al. [21] studied the difference between techniques which operated at *fine granularity* (at the level of source code statements) and *coarser granularity* (at the function level). Qu et al. [52] investigated the impact configurable systems had on the effectiveness of prioritization techniques. Another study by Elbaum et al. [20] investigated the effects of varying levels of fault severity and test costs.

Each of these studies revealed important factors and testing contexts that impact the cost-effectiveness of regression testing techniques. These studies confirmed that prioritization techniques have potential for cost-savings, but the studies also showed that the cost-savings vary greatly across different software programs, and even across different versions of the same software system. These wide variances are attributed to factors involving the program under test, the test suites used to test them, the types of modifications made to the programs, and the testing processes. To properly evaluate prioritization techniques, costs associated with these factors should be considered. The next section discusses cost-benefit models which attempt to better evaluate regression testing techniques by incorporating costs associated with these factors.

2.3. Cost-Benefit Models

Evaluating regression testing techniques by simple metrics such as the rate of fault detection or the number of tests in the test suite ignore important factors which empirical studies discussed in the previous section have shown to impact the performance of the techniques. To address this issue, a few cost-benefit models have been proposed to evaluate regression testing techniques to date.

Leung and White [40] present a model which include the costs related to the testing time and time to execute the regression testing technique. This work was extended by Malishevksky et al. [44] to include benefits related to the omission of faults and rate of fault detection. This work was extended again by Do et al. [18] to incorporate additional cost factors related to software artifact analysis and technique execution time.

A more comprehensive cost-benefit model, the EVOMO model, was provided by Do and Rothermel [15] which accounts for additional context factors and considers costs and benefits across entire system lifetimes, rather than on single releases of those systems. In order to simplify the model, Do and Rothermel [17] performed a sensitivity analysis on the model. The results of their study showed the simplified model was able to assess relationships among the regression testing techniques in the same way as the full model. The simplified model is less expensive to utilize because it requires measuring fewer metrics. The simplified EVOMO model is the cost-benefit model used in this work to evaluate the prioritization techniques in each of the studies. In order to provide a more practical cost-benefit analysis of the ART strategies, in two of the studies, the EVOMO model was extended to include the cost of applying the ART strategy. The modification made to the EVOMO model to include this cost is discussed in more detail in Chapter 5.

The EVOMO model involves two equations: one that captures costs related to the salaries of the engineers who perform regression testing (to translate time

spent into monetary values) and one that captures revenue gains or losses related to changes in system release time (to translate time-to-release into monetary values). Significantly, the model accounts for costs and benefits across entire system lifetimes, rather than on snapshots (i.e. single releases) of those systems, through equations that calculate costs and benefits *across entire sequences of system releases*. The two equations that comprise EVOMO are as shown in Equation 1 and 2. A summary of the terms and coefficients used in the EVOMO model are summarized in Table 2. A more detailed description of the costs is described next.

$$Cost = PS * (\sum_{i=2}^{n}(CO_i(i) + CO_r(i) + c(i) * CF(i)) + K_1) \tag{1}$$

$$\begin{aligned} Benefit = REV * (\sum_{i=2}^{n}(ED(i) - (CO_i(i) + CO_r(i) + a_{tr}(i-1) * CA_{tr}(i-1) \\ +CR(i) + b(i) * CE(i) + CD(i))) - K_2) \end{aligned} \tag{2}$$

Cost of test setup (CS). CS includes the cost of activities required for preparing to run the tests. Some costs included are the cost of setting up the testing environment (both hardware and software) and arranging for the use of resources. This cost can vary based on the characteristics of the system under test.

Cost of identifying obsolete test cases (CO_i). This includes the costs of manual inspection of a version and its test cases, and determination, given modifications made to the system, which test cases are still applicable to the next version. This cost varies based on the type of test cases in the system and the experience of the test engineer.

Cost of repairing obsolete test cases (CO_r). Obsolete test cases can still be useful for subsequent versions of a system (for example, when a class interface is changed by one parameter) and therefore the test case may be repaired so it is no longer obsolete. This cost varies with the number of test cases needing repair and

Table 2. Coefficients and Terms for *EVOMO* Cost-Benefit Model

Term	Description
S	software system
i	index denoting a release S_i of S
n	the number of releases of the software system
$CS(i)$	time to setup for testing activities S_i
$CO_i(i)$	time to identify obsolete tests
$CO_r(i)$	time to repair obsolete tests
$CA_{in}(i)$	time to instrument all units in i
$CA_{tr}(i)$	time to collect traces for test cases in S_{i-1}
$CR(i)$	time to execute a technique itself on S_i
$CE(i)$	time to execute test cases on S_i
$CV_d(i)$	time to apply tools to check outputs of test cases run on S_i
$CV_i(i)$	time for inspecting the results of test cases
$CF(i)$	cost associated with missed faults after the delivery of S_i
$CD(i)$	cost associated with delayed fault detection feedback on S_i
REV	revenue in dollars per unit
PS	average hourly programmer's salary in dollars per unit
$ED(i)$	expected time-to-delivery for S_i when testing beings
$a_{in}(i)$	coefficient to capture reductions in costs of instrumentation for S_i due to the use of incremental analysis techniques
$a_{tr}(i)$	coefficient to capture reductions in costs of trace collection for S_i due to the use of incremental analysis techniques
$b(i)$	coefficient to capture reductions in costs of executing and validating test cases for S_i due to the use of incremental analysis techniques
$c(i)$	number of faults that are not detected by test suite on S_i
K_1	a fixed value used for CS and CV
K_2	a fixed value used for CS, CV, CA_{in}, and a_{in}

the complexity of the repairs.

Cost of supporting analysis (CA). This cost represents the costs of the analysis needed to support a regression testing technique. Some examples are the cost of instrumenting code, analyzing changes between old and new versions, and collecting test execution traces. This cost can vary greatly with the characteristics of the regression testing technique being used, the program being tested, and the tests in the test suite.

Cost of technique execution (CR). This is the time required to execute a regression testing technique itself. Like the cost of supporting analysis, this cost varies with the characteristics of the regression testing technique being used, the program being tested, and the tests in the test suite.

Cost of test execution (CE). This is the time required to execute the test cases. This cost can vary based on the test execution process. For example, if the execution process is manual, the cost is likely to be a lot higher than if it is automatic. The system under test and the particular test cases can also affect the cost.

Cost of test result validation (automatic via differencing) (CV_d). This is the time required to run a differencing tool on test outputs as test cases are executed.

Cost of test result validation (human via inspection) (CV_i). This is the time needed by engineers to inspect test output comparisons.

Missing faults (c **and** CF). For any regression testing technique that could miss faults, the number of faults missed, c, is measured. The cost of the missed faults is represented by CF.

Cost of delayed fault detection feedback (CD). CD captures the cost of delayed fault detection feedback. When faults are detected late in a regression testing cycle, efforts to correct them can delay product release. Faults detected early in a cycle can potentially be addressed prior to completion of the cycle.

2.4. Adaptive Regression Testing Strategies (ART)

The empirical studies discussed in this chapter that evaluate prioritization techniques revealed wide variances in performance across different software programs, and even across different versions of the same software system. These variances are attributed to factors involving the program under test, the test suites

used to test them, the types of modifications made to the programs, and the testing processes. Therefore, there is no single technique which is most cost-effective for each regression testing session.

Figure 2 presents an example of this situation. In this figure, there are four versions for a software system (V1, V2, V3, and V4), and four regression testing techniques being considered (Tech1, Tech2, Tech3, and Tech4). The arrows point, for each version, to the most cost-effective technique for that version. For V1, Tech2 is most cost-effective; for V2, Tech3 is most cost-effective; for V3, Tech1 is most cost-effective; for V4, Tech4 is most cost-effective. Since no single technique is most cost-effective, if one technique was used for all the versions it would be more costly than if the most cost-effective technique was identified and used for each version.

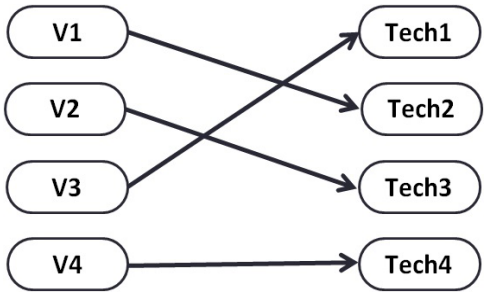


Figure 2. Mapping of Most Cost-Effective Technique for Each Software Version

This presents the problem, then, of how is the most-cost effective technique for a particular regression testing session identified? Very little research has been conducted on the problem of helping researchers and practitioners choose the most cost-effective technique for a particular software version. As an initial step towards solving this problem, one study by Arafeen and Do [5] proposed adaptive regression testing (ART) strategies to help identify the most cost-effective regression testing technique for each regression testing session.

ART strategies help researchers and practitioners consider important environment and testing factors in order to choose the most cost-effective technique for a particular regression testing session. In this study, one ART strategy utilizing the AHP method was developed. An experiment was conducted to evaluate whether the ART strategy was able to effectively choose the most cost-effective technique for each regression testing session. The results of the study were promising. When looking at the total of each of the cost-benefit calculations for all of the versions of a software system, the ART strategy was more cost-effective than the control strategies used in the experiment. However, when looking at each individual version, in several cases, the most cost-effective technique was not chosen. In order to capitalize on the cost-savings across a system's lifetime, the amount of versions which utilize the most cost-effective technique needs to be maximized. To develop strategies to do this, the weaknesses of the previous study and proposed ART strategy was analyzed in this work.

This research identifies several limitations of the study and the proposed strategy. To address these limitations, new ART strategies are presented. The decision making methods used in the strategies are discussed in the next sections of this chapter.

2.5. Multiple Criteria Decision Making Problems

A MCDM problem is a problem which has multiple conflicting criteria. Deciding which prioritization technique to use in a regression testing session has many different factors to consider which have trade-offs. These trade-offs are considered to be conflicting criteria. Therefore, methods which have been developed to solve MCDM problems would be appropriate to use to develop ART strategies. In fact, the previously proposed strategy utilizes the MCDM method, AHP.

Many MCDM methods have been proposed. The majority of the proposed methods involve numerical analysis of possible alternatives. Any MCDM method that involves numerical analysis of possible alternatives have three things in common. First, the methods require decision makers to determine relevant criteria and alternatives. Second, the methods assign numerical measures to the relative importance of the criteria and to the evaluation of alternatives on these criteria. Third, the numerical values are processed to determine a ranking for each alternative.

Of the proposed MCDM methods, the most widely used methods in the literature are WSM (Weighted Sum Model), AHP (Analytic Hierarchy Process), WPM (Weighted Product Model), ELECTRE (for Elimination and Choice Translating Reality), and the TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) [36]. The next three subsections discuss the MCDM methods used in this work: AHP, fuzzy AHP, and WSM.

2.5.1. AHP

The AHP method was developed by Saaty in 1980 [55]. The AHP method begins with the decision makers defining the goal (the problem they wish to solve). After the goal is established, the decision makers determine criteria that are important in achieving the goal, as well as alternatives they are considering utilizing to reach the goal. The goal, criteria, and alternatives are structured into an AHP hierarchy. An example of an AHP hierarchy is shown in Figure 3. The goal is placed at the top of the hierarchy. The next level of the hierarchy contains the criteria. In this figure there are five criteria the decision makers have determined to be important in reaching the goal. The last level of the hierarchy contains the alternatives. In this figure there are three alternatives considered in achieving the goal. The lines connecting each alternative to each criterion show how each alternative is evaluated according to each criterion. The evaluation is performed during the pairwise comparison process which is described next.

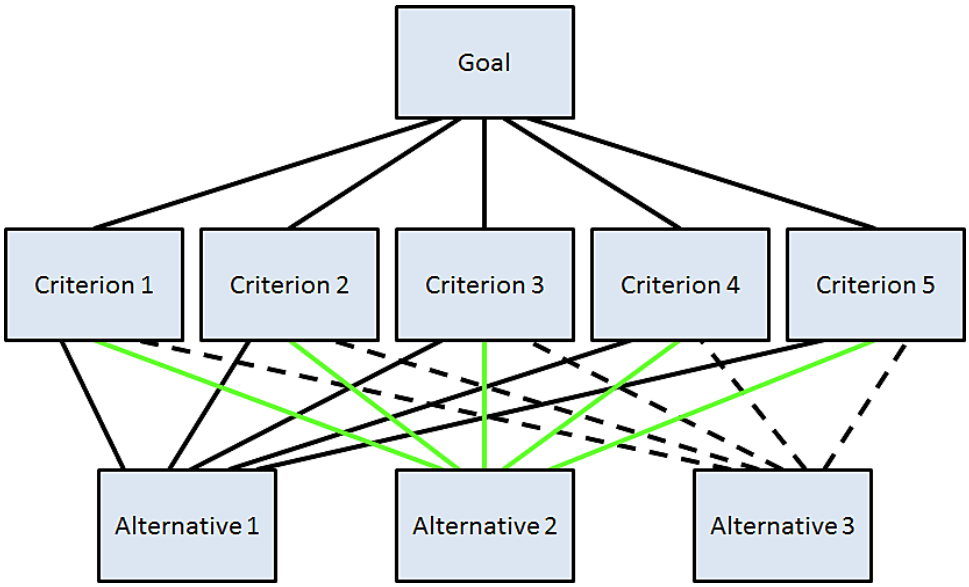


Figure 3. AHP Hierarchy

After the AHP hierarchy has been designed, a set of two pairwise comparisons are performed. The first set of pairwise comparison is between the pairs of criteria and the second set is between the pairs of alternatives. During the pairwise comparison process for the criteria, the person performing the comparisons assigns a relative importance weight to each criterion in the comparison. The importance weight is evaluated in terms of the criterion's importance in reaching the goal. Criteria with a large impact on achieving the goal should receive higher importance weights than those with less of an impact. A scale that is frequently used is the nine-point scale which is shown in Table 3.

Table 3. Scale of Weights

Weight	Definition of Weight
1	equally important
3	moderately important
5	strongly important
7	very strongly important
9	extremely important

After the pairwise comparisons are conducted for the criteria, the weights provided in the comparisons are structured into a matrix. Then, the local priority is calculated using the following equation:

$$LP_i = \frac{\sum_{j=1}^{N}(RW_{ij})}{\sum_{i=1}^{N}\sum_{j=1}^{N}(RW_{ij})} \tag{3}$$

where LP_i is a local priority of criterion i, RW_{ij} is the relative weight of criterion i over criterion j, and N is the number of criteria.

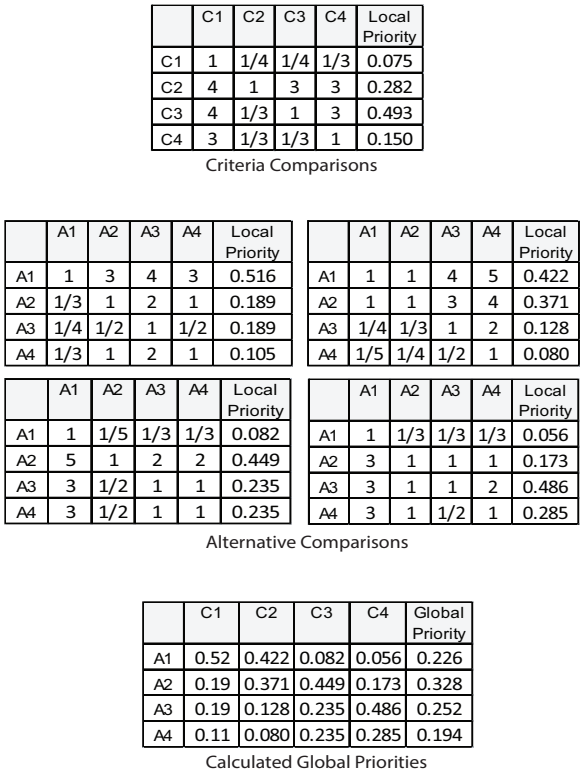
Pairwise comparisons are also completed to calculate the local priority for each alternative in respect to each criterion. The local priority for alternatives uses the same equation as the local priority for criteria. The alternative which more strongly meets the criterion in the comparison receives a higher weight. If an AHP hierarchy contains c criterion, there are c comparison matrices for the alternatives. An $M \times N$ matrix is constructed from the local priorities for criteria and alternatives, where M is the number of alternatives considered and N is the number of criteria. The global priority is then calculated with the following equation:

$$GP_K = \sum_{j=1}^{N}(LPA_{kj}) * (LP_j)) \tag{4}$$

where GP_k is the global priority for alternative k, N is the number of criteria, LPA_{kj} is a local priority of alternative k ($l \leq k \leq$ M) and criterion j, and LP_j is the local priority of criterion j. Using the global priority, the decision maker determines which alternative should be selected. The alternative with the highest global priority is the best alternative.

An example of the pairwise comparison and priority calculation processes is provided in Figure 4. In this example there are four criteria and four alternatives. The first matrix in the figure shows the pairwise comparisons made by the decision maker for the criteria. The decision maker ranks the criteria comparing the two

criterion in terms of how important they are in meeting the goal. For example, the decision maker assigned a ranking of 4 to C2 when compared to C1 (which means C1 is given a ranking of 1/4 when compared to C2). Table 3 shows a ranking of 4 means C2 is somewhere between moderately and strongly more important towards reaching the goal when compared to C1.



	C1	C2	C3	C4	Local Priority
C1	1	1/4	1/4	1/3	0.075
C2	4	1	3	3	0.282
C3	4	1/3	1	3	0.493
C4	3	1/3	1/3	1	0.150

Criteria Comparisons

	A1	A2	A3	A4	Local Priority
A1	1	3	4	3	0.516
A2	1/3	1	2	1	0.189
A3	1/4	1/2	1	1/2	0.189
A4	1/3	1	2	1	0.105

	A1	A2	A3	A4	Local Priority
A1	1	1	4	5	0.422
A2	1	1	3	4	0.371
A3	1/4	1/3	1	2	0.128
A4	1/5	1/4	1/2	1	0.080

	A1	A2	A3	A4	Local Priority
A1	1	1/5	1/3	1/3	0.082
A2	5	1	2	2	0.449
A3	3	1/2	1	1	0.235
A4	3	1/2	1	1	0.235

	A1	A2	A3	A4	Local Priority
A1	1	1/3	1/3	1/3	0.056
A2	3	1	1	1	0.173
A3	3	1	1	2	0.486
A4	3	1	1/2	1	0.285

Alternative Comparisons

	C1	C2	C3	C4	Global Priority
A1	0.52	0.422	0.082	0.056	0.226
A2	0.19	0.371	0.449	0.173	0.328
A3	0.19	0.128	0.235	0.486	0.252
A4	0.11	0.080	0.235	0.285	0.194

Calculated Global Priorities

Figure 4. AHP Comparisons and Priorities

The second set of matrices shows the comparisons for each alternative for each criterion. Each matrix represents the pairwise comparisons for each of the alternatives in regards to one criterion.

The bottom matrix takes each of the calculated local priorities for each alternative in regards to each criterion and places them into a final matrix to calculate the global priority. The alternative with the highest global priority is the recommended alternative. In this example, the second alternative, A2, has the highest global priority and should be chosen.

AHP has been used in many different areas. For instance, Kamal and Al-Harbi [32] use AHP in project management to determine the contractors' competence or ability to participate in the project bid. AHP has also been used to analyze and assess risks for a construction project [46], and to select the best maintenance strategy for an important oil refinery [8].

AHP has also been used in the area of software engineering. Ahmad and Laplante [4] use AHP to help select a software project management tool; Sadiq et al. [56] elicit and prioritize software requirements using AHP; and Zhang et al. [82] use AHP to aid in early effort estimation of the project. Karlsson et al. [33] and Perini et al. [49] compare AHP with other alternative methods in prioritizing software requirements. Yoo et al. [78] use AHP to improve test case prioritization techniques by employing expert knowledge and compare the proposed approach with the conventional coverage-based test case prioritization technique.

Although AHP has been shown to be useful in many different areas, there are several drawbacks of the method. First it is frequently criticized for being subjective to the judgements made by the decision makers [60, 67]. Second, its pairwise comparisons often result in inconsistent rankings. [7, 41]. Third, the pairwise comparisons required by the AHP method have been consistently regarded as being too time-consuming. Many empirical studies have been conducted in hopes of measuring important criteria for decision making processes, such as ease of use, time-consumption, and accuracy. These studies have frequently shown that decision makers preferred other methods when compared to AHP because of the time-consuming pairwise comparisons in the AHP method. For example, one study [3] compares five methods to prioritize software requirements. The results of the study showed AHP to be the worst of all of the techniques, with the main complaints being it was difficult to handle, not scalable, and slow. A similar study [27] found AHP to be the hardest to use and took the longest time to perform. Even more

studies confirm these results [34, 42, 50]. Fourth, the AHP method is not scalable. In AHP, the number of comparisons required to calculate priorities for a matrix of n elements is: $\frac{n^2-n}{2}$. The number of comparisons quadratically increases with the number of alternatives. At some point, AHP is no longer practical for large problems. Saaty suggests a limit of 7 ± 2 alternatives [54].

To address these limitations, this research proposes new ART strategies. One of the strategies, through the use of fuzzy AHP, addresses the issue of imprecision in the judgments made by the decision maker. The next section discusses the fuzzy AHP method and related work relevant to fuzzy AHP.

2.5.2. Fuzzy AHP

Fuzzy AHP has consistently been suggested as a way to handle imprecision by the judgements made by the decision makers in the AHP method [38, 60]. Fuzzy AHP methods use fuzzy logic in conjunction with the AHP method. The use of fuzzy logic is argued to handle possible imprecision in input provided by the decision maker.

There are many fuzzy AHP methods proposed by various researchers. Early work on fuzzy AHP was done by Van Laarhoven and Pedrycz [72], in which decision makers express their opinions in fuzzy numbers using triangular membership functions, and the mathematical model includes the logarithmic least squared method. Buckley [9] proposes a method using trapezoidal membership functions. Chang [11] introduces a new approach using triangular fuzzy numbers (TFN) and the extent analysis method. Cheng et al. [12] propose a new method based on linguistic variable weight. Pan [48] proposes a method that combines the use of triangular and trapezoidal fuzzy numbers. Of all of the methods proposed, Chang's extent analysis is, by far, the most commonly used and suggested method to handle the inaccuracies in the decision maker's judgments, and therefore is used in this work. Fuzzy set theory, fuzzy numbers, and the extent analysis method for fuzzy AHP are discussed next.

2.5.2.1 Fuzzy Set Theory and Fuzzy Numbers

To understand fuzzy AHP, some knowledge of fuzzy set theory is required. Fuzzy set theory was originally introduced by Zadeh [79] as a way to handle imprecise data. A fuzzy set is an extension of a conventional set. With conventional sets, elements are considered to either be a part of the set or not be a part of the set. The membership, μ_A, of an element, x, of a classical set, A, is defined by the equation below:

$$\mu_A(x) = \begin{cases} 1, & if \ \ x \in A \\ 0, & if \ \ x \notin A \end{cases} \tag{5}$$

Fuzzy sets allow partial membership. The degree of membership is calculated using a membership function which generates the degree of membership on the interval [0, 1]. Fuzzy sets can be formally defined by:

$$A = (x, \mu_A(x))|x \in X, \mu_A(x) : X \rightarrow [0, 1]) \tag{6}$$

where A is the fuzzy set, μ_A is the membership function, and X is the universe of discourse.

Calculating the degree of membership to a fuzzy set is performed by membership functions. There are different forms of membership functions. Three commonly used membership functions are triangular, trapezoidal, and gaussian. The triangular membership function is described using three values (a, b, c) where b is the modal value, a is the minimum boundary, and c is the maximum boundary. The trapezoidal membership function is described using four values (a, b, c, d), where a is the minimum value, b is the minimum support value, c is the maximum support value, and d is the maximum value. The gaussian membership function transforms the values into a normal distribution with the midpoint defining the ideal definition for the set. The midpoint is assigned a membership degree of 1.

To demonstrate how fuzzy set theory works, consider an example of a person's height. In this example, a person is classified as short if they are 40 inches or less, average if they are over 40 inches but less than 80 inches, and tall if they are over 80 inches. To reflect these classifications, the following traditional sets are defined:

$$A = \{x \mid \ x \leq 40\}$$

$$B = \{x \mid \ x > 40 \ \ and \ \ x < 80\}$$

$$C = \{x \mid \ x \ \ \geq 80\}$$

Membership for the traditional sets would then be:

$$\mu_A(x) = \begin{cases} 1, & if \ \ x \in A \\ 0, & if \ \ x \notin A \end{cases} \tag{7}$$

$$\mu_B(x) = \begin{cases} 1, & if \ \ x \in B \\ 0, & if \ \ x \notin B \end{cases} \tag{8}$$

$$\mu_C(x) = \begin{cases} 1, & if \ \ x \in C \\ 0, & if \ \ x \notin C \end{cases} \tag{9}$$

Now, imagine a person who has a height of 79 inches. In a traditional set, he would not be considered tall, he would be considered average height. If the input was off by 1 inch, he would be classified completely different. Now consider this in terms of fuzzy logic using fuzzy sets and triangular membership functions as an example. The appropriate fuzzy sets are defined in Table 4.

Table 4. Fuzzy Sets for Height Example

Linguistic Term	Triangular Fuzzy Number (a, b, c)
short	(0, 0, 40)
average	(0, 40, 80)
tall	(40, 80, 120)

Using these fuzzy sets and triangular membership functions, the degree of membership to a particular set can be calculated for given input. Traditionally, the triangular membership function is calculated using Equation 10.

$$\mu_A(x) = \begin{cases} 0, & x < a \\ 1 - \frac{|b-x|}{c-a}/2, & a < x < c \\ 0 & x > c \end{cases} \tag{10}$$

Using this equation a person with a height of 79 inches would have a membership degree of 0.975 to the fuzzy set *tall* and 0.025 to the fuzzy set *avg*. So this person would still be considered mostly *tall*, and only partially *average*, resulting in a much more accurate classification than being classified as only *average*.

2.5.2.2 Extent Analysis Method of Fuzzy AHP

The extent analysis uses triangular fuzzy numbers (TFNs). To understand some of the equations in the extent analysis, an understanding about some of the algebraic operations on TFNs is required. Consider the following TFNs: $A = (l_1, m_1, u_1)$ and $B = (l_2, m_2, u_2)$. Using those fuzzy numbers, the following algebraic operations are defined:

1. Addition:

$$A + B = (l_1 + l_2, m_1 + m_2, u_1 + u_2) \tag{11}$$

2. Multiplication:

$$AxB = (l_1 l_2, m_1 m_2, u_1 u_2) \tag{12}$$

3. Inverse:

$$A^{-1} \approx (\frac{1}{u_1}, \frac{1}{m_1}, \frac{1}{l_1}) \tag{13}$$

The extent analysis method begins with the same process used in the traditional AHP method. First, it creates the hierarchical structure, including the goal,

criteria, and alternatives. After the hierarchy is structured, the process continues by completing pairwise comparisons for the criteria and alternatives. The important difference between the traditional AHP and fuzzy AHP methods in this step, is that the crisp values for the importance weights given by the decision maker are converted into TFNs. The TFNs that correspond to the AHP weights used are shown in Table 5 [11].

Table 5. Fuzzy Number Scale

AHP Weight	TFN	Definition of Weight
1	(1, 1, 1)	equally important
3	(1, 3, 5)	moderately important
5	(3, 5, 7)	strongly important
7	(5, 7, 9)	very strongly important
9	(7, 9, 11)	extremely important

An overview of the fuzzy AHP process is provided in Figure 5. The decision maker makes the pairwise comparisons, which are converted into their corresponding TFNs. The TFN comparisons are then used in a four step process which consists of finding the synthetic extent value, computing the degree of possibility, normalizing the weight vector, and choosing the optimal alternative. Each of these steps are described in more detail next.

Step 1: Find the fuzzy synthetic extent with respect to the ith *object.*

To calculate the fuzzy synthetic extent value, let $C = \{C_1, C_2, ..., C_n\}$ be a set of n criteria, and $A = \{A_1, A_2, ..., A_m\}$ be a set of m alternatives, and M_C^j are TFNs for the ith criteria. The value of the fuzzy synthetic extent S_i with respect to the ith criteria is defined as follows:

$$S_i = \sum_{j=1}^{m}(M_C^j)[\sum_{i=1}^{n}(\sum_{j=1}^{m}(M_C^j)]^{-1} \tag{14}$$

To obtain $\sum_{j=1}^{m}(M_C^j)$, fuzzy addition of m extent analysis for a particular matrix is

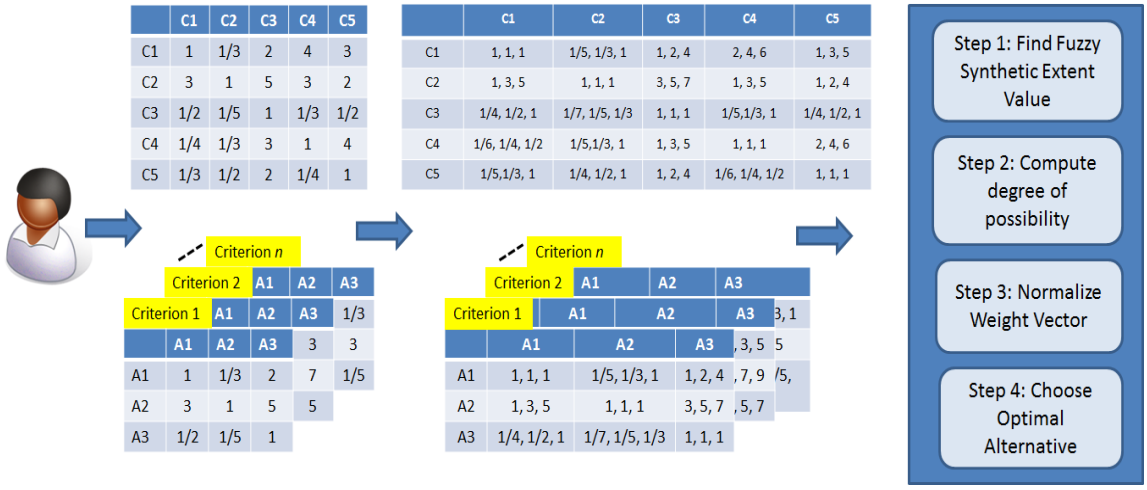


Figure 5. Overview of Fuzzy AHP Method

performed such that:

$$\sum_{j=1}^{m}(M_C^j) = \sum_{j=1}^{m}(l_j), \sum_{j=1}^{m}(m_j), \sum_{j=1}^{m}(u_j) \tag{15}$$

and to obtain $[\sum_{i=1}^{n}(\sum_{j=1}^{m}(M_C^j)]^{-1}$, perform fuzzy addition operations such that:

$$\sum_{i=1}^{n}(\sum_{j=1}^{m}(M_C^j) = \sum_{i=1}^{n}(l_i), \sum_{i=1}^{n}(m_i), \sum_{i=1}^{n}(u_i) \tag{16}$$

finally, compute the inverse by:

$$[\sum_{i=1}^{n}(\sum_{j=1}^{m}(M_C^j)]^{-1} = \frac{1}{\sum_{i=1}^{n}(u_i)}, \frac{1}{\sum_{i=1}^{n}(m_i)}, \frac{1}{\sum_{i=1}^{n}(l_i)} \tag{17}$$

Step 2: Compute the degree of possibility to get the non-fuzzy weight vector, V.

$$V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} min(S_1 \geq S_k) \\ min(S_2 \geq S_k) \\ \vdots \\ min(S_n \geq S_k) \end{bmatrix} \tag{18}$$

where, for element i, the subscript $k \in \{1, 2, ..., n\}$ and $k \neq i$. The degree of possibility of $S_2 = (l_2, m_2, u_2) \geq S_1 = (l_1, m_1, u_1)$ is obtained by:

$$V(S_2 \geq S_1) = \begin{cases} 1, & if \ \ m_2 \geq m_1 \\ 0, & if \ \ l_1 \geq u_2 \\ \frac{l_1 - u_2}{(m_2 - u_2) - (m_1 - l_1)}, & otherwise \end{cases} \tag{19}$$

Step 3: Normalize the weight vector

$$W = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} \frac{v_1}{\sum_{i=1}^{n}(v_i)} \\ \frac{v_2}{\sum_{i=1}^{n}(v_i)} \\ \vdots \\ \frac{v_n}{\sum_{i=1}^{n}(v_i)} \end{bmatrix} \tag{20}$$

Step 4: Choose the Optimal Alternative

The optimal alternative is the alternative with the highest global priority that is obtained from Step 3.

Although it is frequently suggested to use fuzzy AHP as a way to handle imprecision of the judgments made by the decision makers, there were no empirical studies found to support that fuzzy AHP is more effective than traditional AHP. All of the support for fuzzy AHP is in theory. This research includes an empirical study which compares the effectiveness of fuzzy AHP and traditional AHP in regards to

choosing the most cost-effective regression testing technique for a particular software version. The results of the study (discussed in Chapter 4) indicate that fuzzy AHP is more consistent than traditional AHP at choosing the most cost-effective technique for regression testing sessions.

The strategy utilizing fuzzy AHP addresses one limitation of the AHP method, but several limitations, such as inconsistent comparisons, the time required by the method, and thus the limited scalability of the method, still remain. To address these issues, additional ART strategies are proposed in this work. The decision making methods used (WSM and fuzzy expert systems) for the remaining ART strategies proposed in this research are discussed in the next sections.

2.5.3. Weighted Sum Model (WSM)

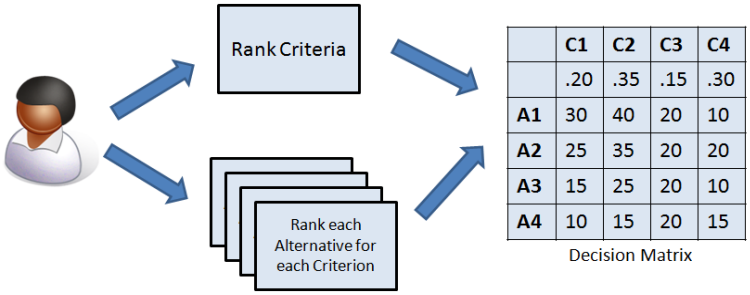
The Weighted Sum Model (WSM) is a simple method in which the decision makers provide the weights for criteria and alternatives, and the weighted sum is used to determine an alternative's preference. A general definition for the WSM with M alternatives and N criteria is as follows:

$$S = \sum_{j=1}^{N} (cw_j aw_{ij}) \tag{21}$$

where cw_j is the relative weight of importance of the criterion C_j and aw_j is the performance value of alternative A_j in terms of C_j.

An overview of the process for the WSM is provided in Figure 6. In this figure, there are four alternatives ($A1$, $A2$, $A3$, and $A4$) and four criteria ($C1$, $C2$, $C3$, $C4$). First, each criterion is given a relative importance weight. Then, each alternative is given a performance score in regards to each criterion. The relative importance weights for the criteria and the performance scores for the alternatives are placed in a decision matrix. In this example, the decision maker used a proportional weighting system (with a sum of 1) to weight the criteria. Using this method, criterion which are more important to reaching the goal receive a larger proportion. For example,

the decision maker in the figure felt the second criteria ($C2$) was the most important criteria in achieving this goal, so it was given the highest weight.



	C1	C2	C3	C4
	.20	.35	.15	.30
A1	30	40	20	10
A2	25	35	20	20
A3	15	25	20	10
A4	10	15	20	15

Figure 6. Overview of Weighted Sum Model

The WSM requires all criteria to have a consistent scale. In other words, if a higher value is deemed better for one criteria, each of the other criteria must have the same requirement. In problems where higher scores are better, the alternative with the highest weighted sum is chosen. In problems where lower scores are better, the alternative with the lowest weighted sum is chosen. In this example, for all of the criteria a higher ranking is better.

After the decision matrix is constructed, the weighted sum for each alternative is calculated. For example, the weighted sum for the first alternative ($A1$) in the figure can be calculated by:

$S_1 = 30 \text{ x } .20 + 40 \text{ x } .35 + 20 \text{ x } .15 + 10 \text{ x } .3 = 26$

To use the WSM to aid in choosing the best alternative, the weighted sum for each of the remaining alternatives would be calculated. After each of the weighted sums are calculated, the results can be compared to determine the best alternative. Since in this example the higher the rating the better, the alternative with the highest weighted sum is the preferred alternative.

The WSM [24] is one of the earliest and simplest MCDM methods developed, but is still widely used in many different areas. In fact, some argue [36, 70, 80] that it

is still one of the most popular and well known methods today. For example, just one area it has recently been used is in the medical field in public health assessments [66] and aiding with the scheduling of physicians [25]. More closely related to this work, the WSM has also been used recently in software engineering. A couple examples of how it has been used is to assess risks in software maintenance [1] and instantiate a variability model in requirements engineering [69].

The main reason why the weighted sum model is popular is because of its simplicity. If WSM is used in an ART strategy, its simplicity could provide a very low-cost strategy. For this reason, this research presents a new ART strategy utilizing the WSM to investigate the impact of using a simple, low-cost decision making method in an ART strategy has on the cost-benefit calculations for the strategy. The cost-benefit results of the ART strategy utilizing the WSM are compared with the results of the other ART strategies presented in this work in Chapter 6.

2.6. Fuzzy Expert Systems

AHP and fuzzy AHP use pairwise comparisons made by decision makers to provide their results. As discussed earlier, pairwise comparisons are frequently inconsistent, very time-consuming, and limited in scalability. A fuzzy expert system can address these issues by eliminating the need for pairwise comparisons, and also providing a system in which expert knowledge can be used to determine the best option. Fuzzy expert systems simulate the human decision making process, while accounting for the uncertainties of it through the use of fuzzy logic.

2.6.1. Fuzzy Expert Systems

A fuzzy expert system is an expert system comprised of fuzzy membership functions and rules. It contains three main parts: fuzzification, fuzzy inference, and defuzzification. A fuzzy expert system is represented in Figure 7. The process begins by the decision maker given crisp input to the fuzzy expert system. The

fuzzification process, using membership functions, provides a fuzzy input set to the fuzzy inference process. The fuzzy inference process uses fuzzy rules built from a knowledge base to provide a fuzzy output set to the defuzzification process. The defuzzification process takes that fuzzy output set and provides crisp output to the decision maker to use in the decision making process. Each of these processes is described in more detail in this section.

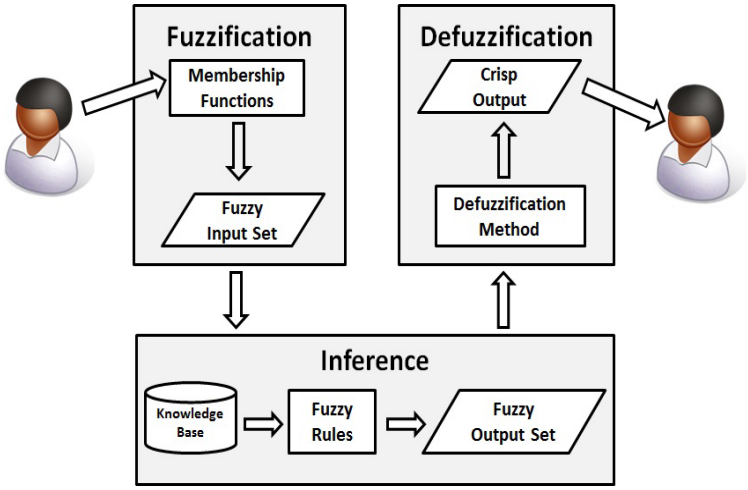


Figure 7. Fuzzy Expert System

2.6.1.1 Fuzzification

The fuzzification process takes input from the decision maker, and determines its degree of membership to the fuzzy sets using membership functions defined in the fuzzy expert system. Fuzzy sets and fuzzy set theory is described in more detail earlier in this chapter.

2.6.1.2 Fuzzy Inference

The fuzzy inference system takes the fuzzified input from the fuzzification process, and determines fuzzy output. The fuzzy inference process maps all inputs $x = [x_1, x_2, \ldots, x_n]$ to an output $f(x)$. The mapping is done using fuzzy rules. The antecedent of the fuzzy rule defines the fuzzy region of the input space, and the

consequent defines the fuzzy region of the output space. The fuzzy inference process is modeled in Figure 8. In this figure, the fuzzy inference process is shown in the area outlined by the dotted line. This particular inference system has three rules that are used to map the input x to an appropriate output set. $A1$, $A2$, and $A3$ are linguistic variables that categorize the input. Based on the categorized input, the rule determines the output (either $B1$, $B2$, or $B3$). For example, using Rule 1, if x is categorized as linguistic variable $A1$, then the output set is $B1$.

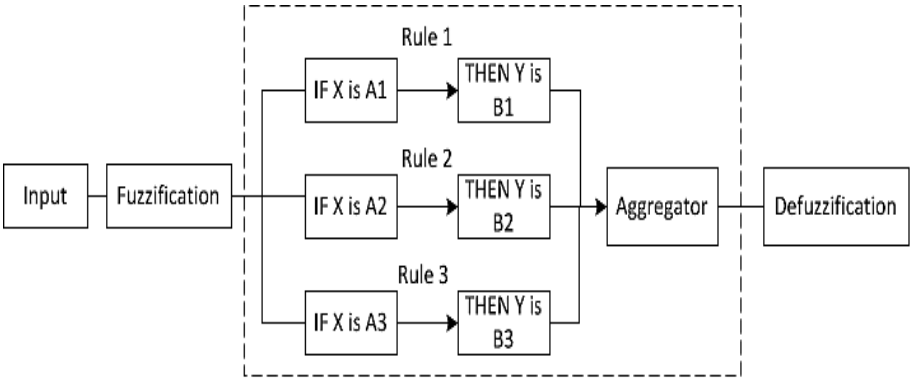


Figure 8. Fuzzy Inference Process

There are two popular inference systems: the Mamdani inference system [45] and the Takagi-Sugeno inference system [68]. The Mamdani inference system is the most commonly used system and is used in this research, so the discussion of inference systems is limited to the Mamdani inference system here.

The first step in a Mamdani fuzzy inference system is to match the input to the fuzzy rules which have some degree of truth in the antecedent forming the fuzzy conclusion set. Then the fuzzy rules in the fuzzy conclusion set are evaluated. The next step of the fuzzy inference system is the aggregation of the rule output. All the *then*-parts of the rules are combined into a final output set. The final output set is a fuzzy set which will require defuzzification for the final output.

2.6.1.3 Fuzzy Rules

A fuzzy rule is a conditional statement that uses linguistic variables. The fuzzy rules are used to determine output from fuzzy input. The knowledge needed to construct fuzzy rules in a fuzzy expert system comes from a combination of several difference sources. The most widely used sources are human knowledge and expertise, historical data analysis of a system, and engineering knowledge from existing literature. Fuzzy rules express knowledge about the relationship between input and output variables. A generic fuzzy rule assumes the following form:

If x is A then y is B

where A and B are linguistic values defined by fuzzy sets. The first part of the rule, the *if*-part, is called the antecedent and the *then*-part is called the consequent. Any rule that has some truth in the antecedent will be included in the fuzzy conclusion set. In the fuzzy conclusion set, if the antecedent is true to some degree of membership, then the consequent is also true to that same degree of membership. Some rules may contain more than one input in the antecedent, and the input variables may be combined using fuzzy set operators such as AND or OR. A generic fuzzy rule with two inputs, one using AND and one using OR is shown here:

If x is A AND y is B then z is C

If x is B OR y is B then z is C

where A, B, and C are linguistic values defined in the fuzzy set, x and y are the input variables, and z is the output variable. One of the most common ways for evaluating fuzzy rules with fuzzy operators is the Zadeh technique [79], which is also referred to as the min-max technique. The Zadeh technique for the fuzzy intersection takes the minimum degree of membership in the membership values of the antecedent. The technique is defined by:

$$\mu_{A \cap B}(x) = min[\mu_A(x), \mu_B(x)] \tag{22}$$

The Zadeh technique for fuzzy union takes the maximum degree of membership in the membership values of the antecedent. The technique is defined by:

$$\mu_{A \cup B}(x) = max[\mu_A(x), \mu_B(x)] \tag{23}$$

2.6.1.4 Defuzzification

Defuzzification is the way the fuzzy output from the inference process is converted to a crisp value. Many different defuzzification techniques have been proposed, but center of gravity is the most widely accepted and regarded as being accurate [61, 65]. The definition of the center of gravity is:

$$y^* = \frac{\int \mu_B(y) y dy}{\int \mu_B(y) dy} \tag{24}$$

where y^* is the defuzzified output, $\mu_B(y)$ is the aggregated membership function, and y is the output variable.

Fuzzy expert systems have been developed in many different areas to provide a simplified way to make complex decisions. For example, fuzzy expert systems have been developed in the medical field to diagnose heart disease [2] and back pain [31]. In economics, for choosing stock in the stock exchange [23], and in flight operations, to access risk in aviation [26].

Fuzzy expert systems have also been developed in the area of software engineering. They have been used frequently in software cost estimation [30, 35, 47]. There has been very little use in the area of software testing, however. Xu et. al developed a fuzzy expert system to build a new test selection technique [76]. This work focuses on test case prioritization techniques, and considers system lifetime and testing processes to help identify the most cost-effective technique for a specific regression testing session.

3. ADAPTIVE REGRESSION TESTING STRATEGIES

This chapter discusses the ART strategies used in this research. The first ART strategy discussed is the ART strategy utilizing the AHP method. This strategy was presented in prior work [5], but an overview of the strategy is provided here to better understand the new strategies presented in this work and their advantages over the AHP-based strategy. Then, three new strategies, which were developed in this research to address weaknesses of the previously proposed strategy, are presented and discussed. Each strategy discussed in this chapter is evaluated by empirical studies in this research (the results of the studies are presented in later chapters) in order to investigate their cost-effectiveness and to provide empirical data for researchers and practitioners to use when choosing strategies for their regression testing sessions.

3.1. ART using AHP

The AHP method was proposed as one potential ART strategy in previous work [5]. A high-level depiction of the process is provided in Figure 9. This figure shows how the decision maker utilizes data from previous empirical studies, history data from prior releases, and software metrics to assign weights to criteria and alternatives (in this figure the alternatives used were prioritization techniques and are depicted as techniques in the figure). The weights provided by the decision maker are entered into an AHP tool that calculates the global priorities for each alternative. The decision maker uses the global priority to choose the most appropriate technique (the alternative with the highest global priority is the preferred alternative).

To utilize the AHP method for ART, the following steps are performed:

1. Step 1: Set a goal

2. Step 2: Identify alternatives to reach the goal

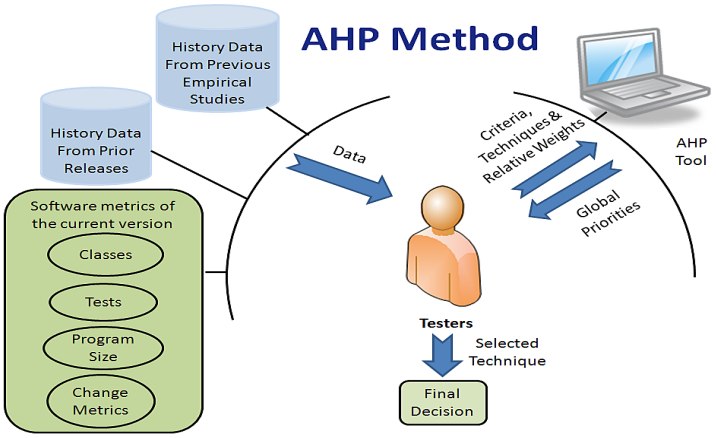


Figure 9. AHP Method

3. Step 3: Identify evaluation criteria for alternatives

4. Step 4: Complete pairwise comparisons between criteria and between alternatives for each criterion

5. Step 5: Obtain global priorities for each alternative

The next sections will describe each of these steps in more detail.

3.1.1. Step 1: Set a Goal

The overall goal for ART is to provide a strategy for cost-savings in regression testing. An example of a more specific goal that could be used in this strategy is to choose the most cost-effective regression testing technique for a particular software version.

3.1.2. Step 2: Identify Alternatives

To identify alternatives for ART, test engineers could consider possible regression testing techniques which have the potential to provide cost-savings for regression testing.

3.1.3. Step 3: Identify Evaluation Criteria

To choose evaluation criteria, test engineers need to consider criteria which affect the cost-effectiveness of regression testing techniques. In the AHP process, the goal, alternatives, and criteria (Steps 1 through 3) are placed in an AHP hierarchy. An example of a possible AHP hierarchy is shown in Figure 10. In this example, the test engineers identified the goal to be choosing the most cost-effective regression testing technique for a particular software version. They chose four possible prioritization techniques (Orig, Rand, Tcov, and Acov) as the alternatives, and four possible evaluation criteria (cost of applying prioritization technique, cost of software artifact analysis, cost of delayed fault detection, and cost of missed faults).

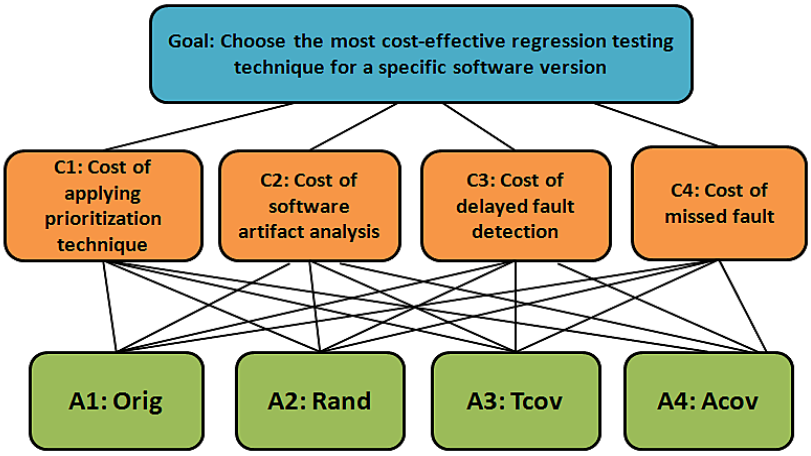


Figure 10. AHP Hierarchy for ART

3.1.4. Step 4: Pairwise Comparisons

After the hierarchy is created based on Steps 1 through 3, a set of pairwise comparisons are performed. In this step, test engineers can utilize knowledge from empirical studies, prior releases of a system, and their experience to evaluate the importance of each criterion in regards to achieving the goal by conducting

pairwise comparisons between each criterion. They will also perform a set of pairwise comparisons between the alternatives in regards to their performance in terms of each criterion.

3.1.5. Step 5: Obtain Global Priorities

Once the pairwise comparisons are complete, the global priority can be calculated (by using Equation 4 provided in Chapter 2). This can be done by an automated tool to save time. Using the global priority for each of the alternatives, the test engineers will be able to choose which regression testing technique they should use for that particular regression testing session.

Although this strategy showed some promising results in the previous study [5], there are several limitations with this strategy which have been discussed in Chapter 2 (such as imprecision in the judgments made by the decision maker and the time required by the pairwise comparisons). To address these limitations, this research presents new ART strategies. These strategies are discussed in the remaining sections of this chapter.

3.2. A Fuzzy AHP Approach to ART

A new ART strategy utilizing the fuzzy AHP method was developed in this research to address the issue of imprecision in the judgments made by the decision maker. Fuzzy AHP has been suggested to handle imprecision in the AHP process, but empirical studies have not been conducted to support this claim. In this work, a new ART strategy is developed using fuzzy AHP to investigate whether using fuzzy logic in conjunction with AHP can more effectively choose the most cost-effective regression testing technique. Later in this work (Chapter 4), a study comparing the cost-benefit results for the fuzzy AHP strategy are compared with the cost-benefit results for the traditional AHP strategy to provide an empirical study lacking in the literature which compares the two methods.

The process for the ART strategy utilizing the fuzzy AHP method is as follows. The fuzzy AHP strategy begins the same way as the traditional AHP strategy, by determining the AHP hierarchy. To create a hierarchy, the test engineer needs to define the goal, alternatives, and criteria. The steps for determining the goal, alternatives, and criteria for an AHP hierarchy for ART were explained previously, and an example of a possible AHP hierarchy for ART is shown in Figure 10.

The fuzzy AHP process continues by the decision makers completing the sets of pairwise comparisons for the criteria and each of the alternatives for every criterion. To perform the comparisons, the decision maker utilizes the commonly used scale for the AHP method (shown in Table 3 in Chapter 2) and assigns a value of 1 through 9. In the pairwise comparisons for the criteria, if the decision maker assigns a high number to one criterion, it is because he or she thought that criterion was more important towards reaching the goal than the other criterion being considered in the comparison. For example, for ART, when comparing the criteria, the decision maker could consider two costs: the cost of a missed fault and the cost of applying the prioritization technique. If the decision maker felt the cost of a missed fault was much more important to reaching the goal (of determining a cost-effective regression testing technique) than the cost of applying the prioritization technique, he or she would rank the cost of a missed fault closer to, or maybe even at, the value of 9 when compared to the cost of applying the prioritization technique.

After all the comparisons are made (for both the criteria and the alternatives in regards to each criterion), the values from all of the pairwise comparisons are converted to their matching TFN according to Table 5 in Chapter 2. Then, the four steps of the extent analysis method are performed on the fuzzy pairwise comparisons.

An example of the extent analysis being applied to ART is illustrated next. The example uses one of the comparisons made by one of the decision makers from the study in Chapter 4. The decision maker in this example was performing

the pairwise comparisons between four alternatives (here, test case prioritization techniques Orig, Tcov, Acov, and Rand) in terms of one specific criterion (in this example, cost of missed faults). The comparisons are shown in Table 6. These comparisons have already been converted to their corresponding TFNs.

Table 6. Fuzzy Pairwise Comparisons

	Orig	Tcov	Acov	Rand
Orig	(1, 1, 1)	(1/6, 1/4, 1/2)	(1/6, 1/4, 1/2)	(1/5, 1/3, 1)
Tcov	(2, 4, 6)	(1, 1, 1)	(1, 3, 5)	(1, 3, 5)
Acov	(2, 4, 6)	(1/5, 1/3, 1)	(1, 1, 1)	(1, 3, 5)
Rand	(1, 3, 5)	(1/5, 1/3, 1)	(1/5, 1/3, 1)	(1, 1, 1)

Once the comparisons matrix is filled with the appropriate TFN's, a four step calculation process is performed to determine the global priority. Each of the equations were presented in Chapter 2, and an example of these calculations using the comparisons from Table 6 is shown here.

Step 1: Find the fuzzy synthetic extent with respect to the ith *object.*

The equation for calculating the fuzzy extent matrix S_i is shown in Equation 14 in Chapter 2. Equation 14 is broken down into equations 15, 16, and 17 (each of these are also shown in Chapter 2).

Using Equation 15 and the pairwise comparisons in Table 6 regarding alternative *Orig*, the following calculation is used:

(1 + 1/6 + 1/6 + 1/5), (1 + 1/4 + 1/4 + 1/3), (1 + 1/2 + 1/2 + 1) = (1.533, 1.833, 3.000)

This calculation is performed on each alternative, resulting in the following matrix:

$$\sum_{j=1}^{m}(M_C^j) = \begin{bmatrix} 1.533 & 1.833 & 3.00 \\ 5 & 11 & 17 \\ 4.2 & 8.33 & 13.00 \\ 2.4 & 4.67 & 8 \end{bmatrix} \tag{25}$$

The next step is to apply Equation 16 to the matrix that was just calculated. An example of these calculations is shown here:

(1.533 + 5 + 4.2 + 2.4, 1.833 + 11 + 8.33 + 4.67, 3.00 + 17 + 13 + 8) = (13.133, 25.833, 41)

Then, compute the inverse using Equation 17:

(1.533/41, 1.833/25.833, 3/13.133) = (.0374, .0710, .2284)

The inverse is calculated for each row, resulting in the fuzzy extent value matrix shown below:

$$S = \begin{bmatrix} .0374 & .0710 & .2284 \\ .1220 & .4258 & 1.2944 \\ .1024 & .3226 & .9898 \\ .0585 & .1806 & .6091 \end{bmatrix} \tag{26}$$

Steps 2 and 3 consist of *computing the degree of possibility to get the non-fuzzy weight vector, V*, and *normalizing the weight vector*. The equations (Equations 19 and 20) for each of these steps are given in Chapter 2. Using those equations on the example presented here, the resulting normalized weight vector is shown in Table 7.

Table 7. Normalized Weight Vector

Orig	Tcov	Acov	Rand
0.0827	0.3585	0.3204	0.2385

Step 4: Choose the Optimal Alternative

The optimal alternative is the alternative with the highest global priority that is obtained from Step 3. In the example, Tcov has the highest global priority and should be chosen as the prioritization technique for that particular regression testing session.

3.3. FESART

The previous two strategies presented in this chapter require pairwise comparisons. Pairwise comparisons are very time consuming, can often result in inconsistent comparisons, and are not scalable. To address these problems, a fuzzy expert system for ART, called FESART, was developed. This section describes FESART, and how each of the main parts of a fuzzy expert system (fuzzification, fuzzy inference using fuzzy rules, and defuzzification) can be applied to ART.

3.3.1. Fuzzification

The fuzzification process takes input from the decision maker and determines its degree of membership to fuzzy sets defined in FESART using the membership functions defined in FESART. The input provided by the decision maker contains information which aids in the decision making process. For example, for ART, costs that impact the cost-effectiveness of regression testing techniques (such as cost of missed faults, cost of delayed fault detection, etc) could be considered. Considering these costs, the decision maker provides some knowledge about how a particular regression testing technique performs according to that cost criterion. For example, if the cost of missed faults was one criterion being considered, the decision maker would provide some judgment about how high (or low) they felt the cost would be in terms of the regression testing technique being considered. The input provided by the decision maker is then fuzzified according to its degree of membership to the membership functions provided in FESART.

The membership functions should appropriately categorize the input criteria so it can be useful for determining appropriate output. For example, consider a FESART system that utilizes three triangular membership functions for each criterion being considered. Triangular membership functions are defined by three values (a, b, c) where b is the modal value, a is the minimum boundary, and c is the maximum boundary. These membership functions are shown in Table 8. The resulting fuzzy input set from the fuzzification process is used as input for the fuzzy inference process.

Table 8. Membership Function for Input Variables

Linguistic Value	Triangular Fuzzy Numbers(*a, b, c*)
Low (L)	(-3, 1, 5)
Average (A)	(1, 5, 9)
High (H)	(5, 9, 13)

3.3.2. Fuzzy Inference

The fuzzy inference process takes the fuzzified input from the fuzzification process and determines the fuzzy output set. Consider a fuzzy output set for FESART with eight triangular membership functions. The output is rated on a scale from 1 to 9, with the membership functions being evenly distributed across these values. The membership functions are shown in Table 9. The output set was built to categorize the overall cost for the regression testing technique and are categorized from low to high. $L1$, $L2$, and $L3$ are considered low costs, with $L1$ being the lowest. Then, $A1$ and $A2$ are categorized as average cost, with $A1$ being lower than $A2$. $H1$, $H2$, and $H3$ are all high costs, with $H3$ being the highest cost.

The fuzzy output set is determined by using fuzzy rules. In a fuzzy expert system, the fuzzy rules bring expert knowledge into the system to aid in the decision making process. The knowledge needed to construct fuzzy rules in a fuzzy expert system comes from a combination of several different sources. The most widely used sources are human knowledge and expertise, historical data analysis of a system,

Table 9. Membership Function for Output Variable

Linguistic Value	Triangular Fuzzy Numbers (*a, b, c*)
L1	(-.14, 1, 2.14)
L2	(1, 2.14, 3.29)
L3	(2.14, 3.29, 4.43)
A1	(3.29, 4.43, 5.57)
A2	(4.43, 5.57, 6.71)
H1	(5.57, 6.71, 7.86)
H2	(6.71, 7.86, 9)
H3	(7.86, 9, 10.14)

and engineering knowledge from existing literature. To develop rules for FESART, knowledge about the factors that influence cost-benefits for regression testing techniques is needed. To gain this knowledge, each of the previously mentioned methods can be used.

If FESART considered four cost criteria (cost of applying prioritization technique, cost of missed faults, cost of delayed fault detection, and cost of software artifact analysis), each criterion could be considered and evaluated through information gained from the methods listed above. Using this knowledge, the criteria could be ordered by their impact on cost-benefit trade-offs. An example order could be the cost of missed faults (CF), cost of delayed fault detection (CD), cost of applying the prioritization techniques (CR), and costs of software artifact analysis (CA), with the cost of missed faults having the strongest impact and the cost of software artifact analysis having the least impact. Fuzzy rules could then be developed so that the final cost is calculated according to the importance ordered here.

3.3.3. Defuzzification

The last step in FESART is to use the defuzzification process to provide decision makers with crisp output to use in their decision making process. Many different defuzzification techniques have been proposed in the literature and are described more in Chapter 2.

3.4. Weighted Sum Model (WSM)

The ART strategy using the WSM begins by the decision makers weighting the criteria. The decision makers do not use pairwise comparisons like the AHP and fuzzy AHP strategy, they just provide a direct weight for each criterion. Like the other strategies for ART, the decision makers would need to determine criteria important to choosing cost-effective regression testing techniques. Then, they determine a performance score for each alternative (regression testing technique) in regards to each criterion. Then the weighted sum is used to determine the best alternative.

The WSM has received many criticisms [37, 59], but is still a widely used decision making method [36, 70]. The WSM is popular because of its simplicity and scalability. For these reasons, this research studies WSM as an ART strategy to investigate how effectively this simple, low-cost approach can identify the most cost-effective regression testing technique for a particular regression testing session.

3.5. Evaluating Cost Criteria for ART

Each of the ART strategies presented in this chapter consider different factors that affect the cost-effectiveness of regression testing (such as costs related to organizations' circumstances, testing environments, and maintenance activities) to choose the most cost-effective technique for a particular regression testing session. This section describes how decision makers can provide the necessary input for each of the strategies regarding these types of cost criteria.

As an example, the strategies could consider the cost of applying regression testing techniques, the cost of code analysis, the cost of fixing missed faults, and so on. In order to consider these costs in each of the strategies to choose the most cost-effective technique for a particular regression testing session, the decision maker can utilize knowledge from previous empirical studies, history data from previous versions of the system, and different software metrics (such as number of classes,

number of tests, program size, and change characteristics). For example, Elbaum et al. [19] reports results of a multiple case study investigating the modifications made in the evolution of four software systems. The goal of their study was to determine how size, distribution, and location of the modifications made to a software system during maintenance impact the cost-effectiveness of regression testing techniques. The results of their study provide helpful trade-offs and constraints that affect the success of regression testing techniques. For example, they found that the distribution of changes greatly impacted the difference in performance between the *additional coverage* (Acov) and *total coverage* (Tcov) prioritization techniques. They found that when the changes were highly distributed, it greatly benefited the Acov technique, but often hurt the performance of Tcov.

Another series of empirical studies performed by Elbaum et al. [21] revealed useful information regarding the effectiveness of different techniques. In general, their studies provide information regarding the trade-off between the benefits of early fault detection versus the cost of applying the regression testing technique itself. If the cost of performing the technique costs more than the savings generated by a higher rate of fault detection, then the technique is not worth employing. A technique is only superior to another technique if the gains achieved by the first technique with respect to the second technique are greater than the additional costs (if any) of using the first technique.

The knowledge gained from these studies and additional studies (e.g. [15, 14, 22]), along with knowledge of the systems under test, knowledge of results of previous versions of the system under test, and the decision maker's experience with regression testing can provide adequate knowledge for decision makers to use in each of the ART strategies presented in this chapter.

4. EMPIRICAL STUDY 1: EVALUATING THE FUZZY AHP APPROACH

This chapter discusses an empirical study [57] conducted to evaluate the new fuzzy AHP-based ART strategy presented in Chapter 3.2. This strategy was developed to address the inaccuracies introduced by the decision maker in the pairwise comparison process. Fuzzy AHP has frequently been suggested in the literature as a way to handle imprecision by the decision makers in the AHP method, but no empirical studies have supported this claim. This empirical study investigates whether the fuzzy AHP method is more effective than the AHP method in terms of ART by studying the following research question:

RQ: Is the fuzzy AHP method more effective than the AHP method for selecting appropriate test case prioritization techniques across the system lifetime?

This experimental design replicates that of the earlier study [5] with an additional test case prioritization technique application mapping strategy, fuzzy AHP. The following subsections present, for this experiment, the objects of analysis, independent variables, dependent variables and measures, and experimental setup and design.

4.1. Objects of Analysis

In this study, five Java programs were obtained from the SIR infrastructure [13]. The programs used were *ant*, *xml-security*, *jmeter*, *nanoxml*, and *galileo*. *Ant* is a Java-based tool similar to the Unix tool *make* where extensions are implemented as Java classes instead of shell-based commands. *Jmeter* is a load-testing tool. *Xml-security* is a component library that implements XML signature and encryption standards. *Nanoxml* is a small XML parser for Java, and *galileo* is a Java bytecode analyzer. Several sequential versions of each of these programs are available. The first three programs are provided with JUnit test suites, and the last two are provided with TSL (Test Specification Language) test suites.

Table 10 lists, for each object of analysis, data on its associated "Versions" (the number of versions of the object program), "Classes" (the number of class files in the latest version of that program), "Size (KLOCs)" (the number of lines of code in the latest version of the program), and "Test Cases" (the number of test cases available for the latest version of the program). To study the research question, fault data is required. To obtain the fault data, mutation faults provided with the programs [16] were used. The rightmost column, "Mutation Faults", is the total number of mutation faults for the program (summed across all versions).

Table 10. Experiment Objects and Associated Data

Objects	Versions	Classes	Size (KLOCs)	Test Cases	Mutation Faults
ant	9	914	61.7	877	412
jmeter	6	434	42.2	78	386
xml-sec.	4	145	15.9	83	246
nanoxml	6	64	3.1	216	204
galileo	16	68	14.5	912	2494

4.2. Variables and Measures

In this study, one independent variable and one dependent variable were manipulated. These variables are discussed in the next two sections.

4.2.1. Independent Variable

To investigate the research question, one independent variable: *test case prioritization technique application mapping strategy*, which assigns, to a specific sequence of versions $S_i, S_{i+1}, \ldots S_j$ for system S, specific test case prioritization techniques is manipulated. As test case prioritization techniques the following techniques were utilized: original order (Orig: the order in which test cases are executed in the original testing scripts provided with the object programs), random order (Rand: in this experiment, averages of 30 runs of random order), and two test case prioritization heuristics (total block coverage (Tcov) and additional block coverage (Acov). Each of these techniques are explained in Chapter 2.

Six mapping strategies are considered in this research as follows:

- Orig-all: Uses the original technique across versions
- Tcov-all: Uses the total block coverage technique across versions
- Acov-all: Uses the additional coverage technique across versions
- Rand-all: Uses the random technique across versions
- AHP: Selects the best technique among four prioritization techniques (Tcov, Acov, Rand, and Orig) using the AHP method described in Chapter 3.1.
- Fuzzy AHP: Selects the best technique among four prioritization techniques (Tcov, Acov, Rand, and Orig) using the fuzzy AHP method described in Chapter 3.2.

4.2.2. Dependent Variable and Measures

The dependent variable in the study is the *relative cost-benefit value* calculated using the EVOMO economic model [17] (described in Chapter 2), and the equation below (Equation 27). The cost and benefit components are measured in dollars. To determine the *relative cost-benefit* of prioritization technique T with respect to baseline technique *base*, the following equation is used:

$$(\text{Benefit}_T - \text{Cost}_T) - (\text{Benefit}_{base} - \text{Cost}_{base}) \tag{27}$$

When this equation is applied, positive values indicate that T is beneficial compared to the *base*, and negative values indicate otherwise. The original technique was used as a baseline in this experiment (meaning that Orig-all is the baseline strategy).

4.3. Experiment Setup and Procedure

In order to measure costs such as delayed fault detection, the object programs needed to contain some faults. Thus, artifacts equipped with mutation faults and

mutant groups were used. The mutants were created by the *ByteME* (Bytecode Mutation Engine) tool from the SIR repository [16]. Each mutant group contained, at most, 10 mutants that were randomly selected per version.

As described previously, both the AHP and fuzzy AHP methods begin by establishing a goal, criteria, and alternatives. The goal of ART is to determine the most cost-effective regression testing technique for a specific software version. Each of the strategies consider the following four criteria:

- Cost of applying test case prioritization technique: the time required to run a test case prioritization algorithm
- Cost of software artifact analysis: the costs of instrumenting programs and collecting test execution traces
- Cost of delayed fault detection: the waiting time for each fault to be exposed while executing test cases under a test case prioritization technique
- Cost of missed fault: the time required to correct missed faults

For alternatives, four test case prioritization techniques are considered (Orig, Tcov, Acov, and Rand). The AHP hierarchy was constructed using the criteria listed above and the test case prioritization techniques as alternatives.

Then, two different human testers, who have over seven years of industry experience, independently performed the set of pairwise comparisons for the criteria and the alternatives in regards to each criterion using the common scale developed for AHP. The scale is described in Chapter 2 and shown in Table 3. The decision makers utilized empirical studies (e.g. [15, 14, 22]), history data from prior releases, and software metrics to assign relative weights in each of the pairwise comparisons (a more detailed description of the how the decision makers utilized this knowledge to assign relative weights in the comparisons is provided in Chapter 3).

These comparisons were entered into an AHP tool to calculate the global priorities for the traditional AHP process. Then, to calculate the values for the fuzzy AHP process, the weights given in the comparisons were converted into their corresponding TFNs according to the scale in Table 5 (from Chapter 3). Then, code was written in MATLAB to calculate the global priorities for the fuzzy AHP method using the extent analysis (the equations for the extent analysis are provided in Chapter 2). The TFNs were run through the calculations in the MATLAB code, and the global priorities for the fuzzy AHP method were recorded. Using each of the global priorities, the prioritization technique recommended by each strategy for each version of every software system was recorded.

Another important factor considered in this experiment is time constraints. Software companies often face strict deadlines with product releases, and due to deadline and budgetary constraints not all of the planned testing can be completed. It is common for software companies to cut back on testing activities in order to ensure a timely release of their product. Because the AHP method was investigated under this situation in the previous study [5], the same regression testing process assumption used in that study is applied to this experiment.

The degree of time constraints during the regression testing phase can vary by the types of maintenance activities for a particular software release or a company's circumstances (e.g., different amount or complexity of feature update, technical personnel loss, etc.). Because of these varying time constraints, this experiment considers different time constraints for each version when the regression testing strategies are applied. To do so, for each of the test case prioritization technique, a random level of time constraints (25%, 50%, or 75%) is assigned for each version. These time constraint levels represent situations where time constraints shorten the testing process by 25%, 50%, and 75%.

To implement time constraint levels, the test execution process was shortened for each version by the assigned time constraint level, and each of the costs from the EVOMO model were measured for each time constraint. Further, four sets of random assignments across all versions for each program as shown in Figure 4.3 were run. For instance, for run 1, each version was given a randomly assigned time constraint: 50% for V1, 25% for V2, 75% for V3, and 50% for V4. This random assignment was repeated four times and defined as "Run n" (n = 1, 2, 3, and 4). Finally, the cost-benefit results for the recommended techniques for each strategy were recorded.

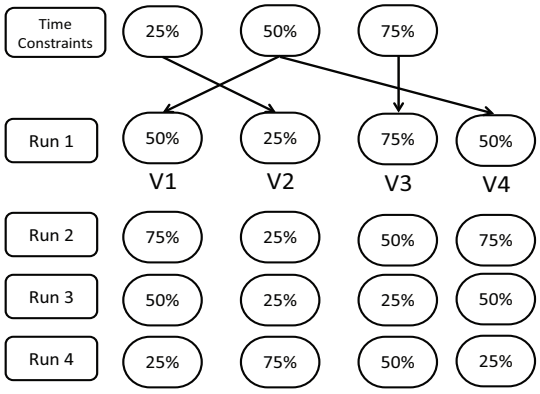


Figure 11. Random Assignment of Time Constraint Levels

4.4. Threats to Validity

This section discusses the construct, internal, and external threats to the validity of this study.

4.4.1. Construct Validity

In this study, four evaluation criteria were used for the AHP hierarchy. These criteria could be a threat to construct validity because other criteria relevant to the testing process could be considered. A second issue is the subjectivity of the decision makers in the pairwise comparisons. As mentioned in prior sections, the subjectivity in pairwise comparisons is a well-known problem with the traditional

AHP method. This issue was addressed by using the fuzzy AHP method, but even with the fuzzy AHP method, some subjectivity is still present.

4.4.2. Internal Validity

The internal validity of this experiment could be threatened by human mistakes. The experiment required collecting pairwise comparisons from two different decision makers. These comparisons had to be entered into an AHP tool for the traditional AHP method and then converted into TFNs and entered into MATLAB for fuzzy AHP calculations. It is possible that data entry mistakes could have happened in this procedure. To control this threat, the AHP tool used in the experiment had algorithms to check against inconsistencies in the pairwise comparisons.

4.4.3. External Validity

There are a few issues that limit the external validity and prevent generalization of the results of the study. First, only one fuzzy AHP method was used. There are other fuzzy AHP methods, and by only using one we cannot generalize if all fuzzy AHP methods would perform better than the traditional AHP method. Furthermore, the results are limited to generalize ART in terms of MCDM methods. This study only considers two MCDM methods. There are many more MCDM methods which have not been considered. Also, four test case prioritization techniques were considered. There are more prioritization techniques that could be studied, such as requirements-based prioritization, risk-based prioritization, or techniques using different algorithms.

4.5. Data and Analysis

In this section, the results of the study are presented. The results for each program are shown in Tables 11 - 17. Results for *ant* are broken into two tables (Tables 11 and 12). Results for *jmeter*, *xml-security*, and *nanoxml* are shown in Tables 13, 14, and 15, and results for galileo are shown in Tables 16 and 17. The table (or tables) for each program shows the cost-benefit results of four runs of random

assignments (run 1 through run 4 in the table) for three time constraint levels (25%, 50%, and 75%) for each version of the five programs. The Orig-all strategy was used as the baseline in the cost-benefit calculations, so it is not displayed in the table.

The data in the table shows the relative cost-benefit value in dollars with respect to the baseline technique (Orig) as defined in Section 4.2.2. The results for decision maker 1 (DM1) are shown under the headings AHP-1 and Fuzzy AHP-1, and decision maker 2 (DM2) is shown under AHP-2 and Fuzzy AHP-2. Positive cost-benefit values indicate a greater cost-benefit than the baseline technique. Negative cost-benefit values indicate less cost-benefits than the baseline technique.

Tables 11 and 12 show the results for *ant*. When looking at the totals for *ant*,

Table 11. Experiment 1: Relative Cost-Benefit Results for *ant* (Runs 1 and 2)

ant							
Run 1							
	Tcov -all	Acov -all	Rand -all	AHP-1	Fuzzy AHP-1	AHP-2	Fuzzy AHP-2
v1	135	77	-40	77	135	77	77
v2	205	209	139	209	209	209	209
v3	-58	-62	48	-62	48	-62	48
v4	-66	14	0	14	14	14	0
v5	-99	-133	26	26	26	26	26
v6	-142	-180	7	7	7	7	7
v7	-160	-201	32	32	32	32	32
v8	-107	-248	146	146	146	146	146
Total	**-292**	**-524**	**358**	**449**	**617**	**449**	**545**
Run 2							
v1	367	232	102	367	367	232	232
v2	205	209	139	209	209	209	209
v3	-151	92	49	92	49	92	92
v4	-155	-72	-58	-72	-58	-58	-58
v5	-157	-191	18	18	18	18	18
v6	-142	-180	7	7	7	7	7
v7	275	234	324	234	324	324	324
v8	-107	-248	146	146	146	146	146
Total	**135**	**76**	**727**	**1002**	**1063**	**970**	**970**

Table 12. Experiment 1: Relative Cost-Benefit Results for *ant* (Runs 3 and 4)

ant							
Run 3							
	Tcov -all	Acov -all	Rand -all	AHP-1	Fuzzy AHP-1	AHP-2	Fuzzy AHP-2
v1	-19	-70	-98	-70	-70	-98	-98
v2	207	209	79	209	207	209	209
v3	-58	-62	48	-62	48	-62	48
v4	12	13	42	13	42	13	42
v5	-99	-133	26	26	26	26	26
v6	-37	-113	87	87	87	87	87
v7	-142	-183	48	48	48	48	48
v8	143	116	292	116	292	116	292
Total	**7**	**-223**	**524**	**367**	**680**	**339**	**654**
Run 4							
v1	135	77	-40	77	135	77	77
v2	55	326	161	326	326	326	326
v3	-59	91	46	91	91	91	46
v4	-66	14	0	14	14	14	0
v5	-145	-179	32	32	32	32	32
v6	337	407	560	560	560	560	560
v7	-160	-201	32	32	32	32	32
v8	-128	115	215	115	215	115	215
Total	**-31**	**650**	**1006**	**1247**	**1405**	**1247**	**1288**

the AHP strategy performed better (meaning the prioritization technique chosen by the AHP strategy (both AHP-1 and AHP-2) was more cost-effective) than the other control strategies except for one case. In run 3, the Rand-all strategy performed better than the AHP strategy for both decision makers. When the results for the fuzzy AHP strategy are compared with those of the first four control strategies (Orig, Rand, Acov, and Tcov), the fuzzy AHP strategy (for both Fuzzy AHP-1 and Fuzzy AHP-2) outperformed the control strategies for all cases. Further, Fuzzy AHP-1 outperformed its corresponding AHP strategy (AHP-1) for all cases, and Fuzzy AHP-2 outperformed AHP-2 for all but one case (in run 3, two strategies produced the same values.). For some versions the cost-benefit values are the same for both

Table 13. Experiment 1: Relative Cost-Benefit Results for *jmeter*

Run 1							
	Tcov -all	Acov -all	Rand -all	AHP-1	Fuzzy AHP-1	AHP-2	Fuzzy AHP-2
v1	15	17	50	15	50	17	17
v2	-51	153	93	153	153	153	153
v3	130	266	277	266	277	266	266
v4	121	31	5	121	121	31	31
v5	-196	-196	-135	-196	-135	-196	-196
Total	**19**	**271**	**290**	**359**	**466**	**271**	**271**
Run 2							
v1	47	135	180	135	135	135	180
v2	-85	-85	-6	-6	-6	-6	-6
v3	130	266	277	266	277	266	266
v4	-64	-65	-142	-65	-65	-64	-64
v5	-174	-144	-136	-144	-136	-144	-136
Total	**-146**	**107**	**172**	**186**	**205**	**187**	**240**
Run 3							
v1	15	17	50	15	50	17	17
v2	-51	153	93	153	153	153	153
v3	-66	22	-36	22	22	22	22
v4	35	274	5	274	274	274	274
v5	-174	-144	-136	-144	-136	-144	-136
Total	**-241**	**322**	**-24**	**320**	**363**	**322**	**330**
Run 4							
v1	-73	116	97	116	116	116	116
v2	-51	153	93	153	153	153	153
v3	-66	22	-36	22	22	22	22
v4	35	274	5	274	274	274	274
v5	-196	-196	-135	-196	-135	-196	-196
Total	**-351**	**369**	**24**	**369**	**430**	**369**	**369**

decision makers. In these instances, the decision maker's had similar rankings, and the strategy then chose the same prioritization technique. In these situations the modifications and testing circumstances for that run made one strategy's advantages (or disadvantages) so apparent that both decision makers ranked them in a similar fashion.

The results for *jmeter* are shown in Table 13. Looking at the results for *jmeter* in regards to the totals for each run, AHP-1 was more cost-effective than the other control strategies for all but one case (in run 3, Acov-all was better than AHP-1). However, AHP-2 was a little less beneficial, only being more cost-effective than the control strategies for one case (run 2), having the same cost-benefits as Acov-all for two cases (runs 3 and 4), and being less cost-effective than Rand-all in run 1. When the fuzzy AHP strategy is compared with the first three control strategies, unlike the result for *ant*, there are some differences between Fuzzy AHP-1 and Fuzzy AHP-2. Fuzzy AHP-1 was more cost-effective than all the control strategies for all cases, but Fuzzy AHP-2 was more cost-effective than those control for only two cases (runs 2 and 3). Compared to the AHP strategy, Fuzzy AHP-1 outperformed AHP-1 for all cases, and Fuzzy AHP-2 outperformed AHP-2 for two cases (runs 2 and 3) and tied for two cases (runs 1 and 4).

Unlike the results for *ant* and *jmeter*, the results for *xml-security* (shown in Table 14) were very different. For both the AHP and fuzzy AHP approaches, the Acov strategy was chosen for all cases. Acov was the most desirable technique for *xml-security* because the changes between subsequent versions were relatively small. Therefore, the decision makers ranked Acov higher in terms of delayed fault detection. Delayed fault detection was also ranked higher than other criteria in the criteria comparisons, so it had a large impact on the final global priorities.

For *nanoxml* (results shown in Table 15), AHP-1 outperformed all control strategies for all cases except the Acov strategy (AHP-1 and Acov produced the same results), and AHP-2 outperformed all control strategies including Acov. Both fuzzy AHP strategies produced better results compared to the control strategies, and the AHP strategy, except for one case (AHP-2 and Fuzzy AHP-2 were tied in run 4).

Table 14. Experiment 1: Relative Cost-Benefit Results for *xml-security*

Run 1							
	Tcov -all	Acov -all	Rand -all	AHP-1	Fuzzy AHP-1	AHP-2	Fuzzy AHP-2
v1	177	274	88	274	274	274	274
v2	26	117	-44	117	117	117	117
v3	170	170	71	170	170	170	170
Total	**373**	**561**	**115**	**561**	**561**	**561**	**561**
Run 2							
v1	37	38	6	38	38	38	38
v2	26	117	-44	117	117	117	117
v3	499	546	315	546	546	546	546
Total	**562**	**701**	**115**	**701**	**701**	**701**	**701**
Run 3							
v1	268	331	203	331	331	331	331
v2	-48	14	-190	14	14	14	14
v3	170	170	71	170	170	170	170
Total	**390**	**515**	**84**	**515**	**515**	**515**	**515**
Run 4							
v1	37	38	6	38	38	38	38
v2	26	117	-44	117	117	117	117
v3	499	546	315	546	546	546	546
Total	**562**	**701**	**277**	**701**	**701**	**701**	**701**

The results for *galileo* (shown in Tables 16 and 17) show both the AHP and fuzzy AHP strategies were more cost-effective than the control strategies for all cases. Furthermore, the fuzzy AHP strategy was more cost-effective than its corresponding AHP strategy (for both decision makers) except for one case (Fuzzy AHP-2 was not better than AHP-2 in run 3).

The total cost-benefit calculation for all versions can be a helpful way to compare the strategies, but one version's cost-benefit calculation could skew the results. To account for this, the data is examined in another way: by the total number of versions that produced the best results.

Figure 12 shows the average (across all runs) for the total number of versions that produced the best results for every strategy for each object program (in this

Table 15. Experiment 1: Relative Cost-Benefit Results for *nanoxml*

	Tcov -all	Acov -all	Rand -all	AHP-1	Fuzzy AHP-1	AHP-2	Fuzzy AHP-2
Run 1							
v1	931	966	975	966	966	975	975
v2	468	778	596	778	778	778	778
v3	-43	40	-27	40	40	40	40
v4	-48	-50	1	-50	1	-50	1
v5	-27	39	-40	39	39	39	39
Total	**1281**	**1773**	**1505**	**1773**	**1824**	**1782**	**1833**
Run 2							
v1	928	962	860	962	962	962	928
v2	565	790	683	790	790	790	790
v3	-43	40	-27	40	40	40	40
v4	-48	-50	1	-50	1	-50	1
v5	451	541	453	541	541	541	541
Total	**1853**	**2282**	**1970**	**2282**	**2334**	**2283**	**2300**
Run 3							
v1	-59	-23	-15	-23	-23	-23	-15
v2	565	790	683	790	790	790	790
v3	163	657	525	657	657	657	657
v4	-48	-50	1	-50	1	-50	1
v5	-27	39	-40	39	39	39	39
Total	**594**	**1413**	**1154**	**1413**	**1464**	**1414**	**1473**
Run 4							
v1	931	966	975	966	966	975	975
v2	468	778	596	778	778	778	778
v3	509	563	482	563	563	563	563
v4	-48	-50	1	-50	1	1	1
v5	451	541	453	541	541	541	541
Total	**2311**	**2798**	**2507**	**2798**	**2849**	**2858**	**2858**

figure, the Orig-all strategies were included). For example, for *ant*, Acov performed best for two versions on average.

To simplify the comparison, when the figure was constructed, results for AHP-1 and AHP-2 were averaged. The results are presented as AHP-avg in the figure. The same process was applied for Fuzzy AHP-1 and Fuzzy AHP-2, and is denoted as Fuzzy AHP-avg in the figure. Examining the average values for AHP and Fuzzy AHP makes sense in practice because when organizations make decisions based

Table 16. Experiment 1: Relative Cost-Benefit Results for *galileo* (Runs 1 and 2)

Run 1							
	Tcov -all	Acov -all	Rand -all	AHP-1	Fuzzy AHP-1	AHP-2	Fuzzy AHP-2
v1	172	691	580	691	691	691	691
v2	-115	366	297	366	366	366	366
v3	235	526	381	526	526	526	526
v4	168	309	380	309	309	309	380
v5	-3	56	9	56	56	56	56
v6	-115	344	283	344	344	344	344
v7	-186	216	130	216	216	216	216
v8	-75	379	289	379	379	379	379
v9	-311	204	118	204	204	204	204
v10	-77	456	151	456	456	456	456
v11	-4	575	528	575	528	575	528
v12	-105	148	154	148	154	148	154
v13	-72	112	250	112	250	112	250
v14	-86	-251	-249	-249	-249	-251	-249
v15	-80	211	293	293	293	293	293
Total	**-654**	**4342**	**3594**	**4426**	**4523**	**4424**	**4594**
Run 2							
v1	-51	461	401	461	461	461	461
v2	-114	358	251	368	368	368	368
v3	51	242	187	242	242	242	242
v4	168	309	380	309	380	309	380
v5	667	700	565	700	700	700	700
v6	-115	344	283	344	344	344	344
v7	-98	224	170	224	224	224	224
v8	-126	364	246	364	364	364	364
v9	-311	204	118	204	204	204	204
v10	-77	456	151	456	456	456	456
v11	-174	579	462	579	462	462	579
v12	-3	136	177	136	177	177	177
v13	-72	112	250	112	250	112	250
v14	-83	-216	-124	-124	-124	-216	-124
v15	-80	211	293	211	293	293	293
Total	**-418**	**4494**	**3810**	**4586**	**4801**	**4500**	**4918**

on the experts' opinions, typically they average the values produced by multiple experts. Some strategies are not present in the figure for some programs because

Table 17. Experiment 1: Relative Cost-Benefit Results for *galileo* (Runs 3 and 4)

Run 3							
	Tcov -all	Acov -all	Rand -all	AHP-1	Fuzzy AHP-1	AHP-2	Fuzzy AHP-2
v1	-125	715	515	715	715	715	715
v2	-114	368	251	368	368	368	368
v3	-75	452	318	452	452	452	452
v4	5	56	13	56	56	56	56
v5	667	700	565	700	700	700	700
v6	-49	228	246	228	228	228	228
v7	-76	285	250	285	285	285	285
v8	-126	364	246	364	364	364	364
v9	-76	469	374	469	469	469	469
v10	-74	405	256	405	405	405	405
v11	-174	579	462	579	462	579	462
v12	-84	228	232	228	232	228	232
v13	-85	-57	120	-57	120	-57	120
v14	-122	273	188	188	188	273	188
v15	-111	-125	64	64	64	64	64
Total	**-619**	**4940**	**4100**	**5044**	**5108**	**5129**	**5108**
Run 4							
v1	-51	461	401	461	461	461	461
v2	-115	366	297	366	366	366	366
v3	51	242	187	242	242	242	242
v4	5	56	13	56	56	56	56
v5	-3	56	13	56	56	56	56
v6	-40	272	262	272	272	272	272
v7	-98	224	170	224	224	224	224
v8	-75	379	289	379	379	379	379
v9	-311	204	118	204	204	204	204
v10	-74	405	256	405	405	405	405
v11	-174	579	462	579	462	462	579
v12	-3	136	177	136	177	177	177
v13	-85	-57	120	-57	120	-57	120
v14	-83	-216	-124	-124	-124	-216	-124
v15	-111	-125	64	64	64	64	64
Total	**-1167**	**2982**	**2705**	**3263**	**3364**	**3095**	**3481**

those strategies did not produce best results for any versions across all runs (e.g., Tcov-all in *nanoxml* and *galileo*).

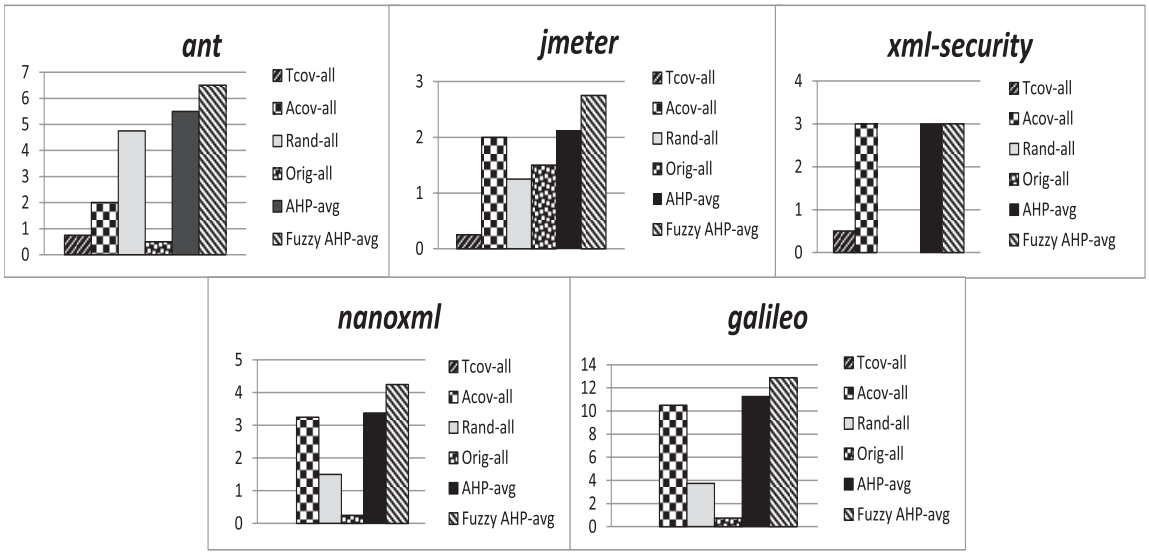


Figure 12. Experiment 1: Total Number of Versions that Were Most Cost-Effective

Overall, Fuzzy AHP-avg produced the best results across all programs except for one program: *xml-security*. For *xml-security*, Fuzzy AHP-avg tied with two other strategies (Acov-all and AHP-avg).

Examining the control techniques overall, AHP-avg outperformed the control strategies for all cases expert for *xml-security* in which case AHP-avg tied with Acov-all. Between Acov-all, Tcov-all, Rand-all, and orig-all, Acov-all's performance was frequently better than the other three strategies, but not in all cases (Rand-all performed better than Acov-all for *ant*) In the case of Tcov-all, its performance was worse than Rand-all and even for Orig-all overall.

4.6. Discussion

The results of this study indicates that the prioritization techniques chosen by the fuzzy AHP process are consistently more cost-effective than the control strategies and the AHP strategy with only a few exceptions. Thus, it can be said that by using fuzzy set theory, a more cost-effective ART strategy is provided.

This statement is supported when examining the data in a couple ways. First, when examining the data by the total cost-benefit calculations, the total cost-

benefits for fuzzy AHP were higher than the other strategies. Second, when the average for the total number of versions where each strategy was most cost-effective is examined, the fuzzy AHP strategy is the highest for all five programs (with the AHP and Acov-all strategies being equal for the *ant* program). Even when the number of the most cost-effective versions is averaged across all runs, the fuzzy AHP strategy is consistently more cost-effective than all other strategies. This result implies that fuzzy AHP's performance is more stable across all programs for all runs than the each of the control strategies and the AHP strategy.

The findings of this study provide significant implications for practitioners and researchers in software engineering. The results show that there is potential for cost-savings in regression testing by choosing a prioritization technique based on criteria related to a specific software version. Furthermore, this study provides a comparison of two different MCDM strategies and provides practitioners with empirical data to show which MCDM strategy performs better in the context of ART.

In addition, fuzzy AHP has frequently been suggested in the literature and used to handle the imprecision of judgments made by the decision maker, but there has not been any empirical study to validate the claims that the fuzzy AHP process is more effective than the AHP process at choosing the best alternative. This study finally provides empirical evidence that, in the area of regression testing, the fuzzy AHP process is more effective than the AHP process by choosing the most cost-effective regression testing technique more frequently than the traditional AHP method.

5. EMPIRICAL STUDY 2: EVALUATING A FUZZY EXPERT SYSTEM FOR ART

This chapter discusses an empirical study [58] conducted to evaluate a fuzzy expert system for ART. A fuzzy expert system can address the time and scalability issues of the fuzzy AHP and AHP strategies because it does not require pairwise comparisons. To investigate whether a fuzzy expert system can provide a cost-effective ART strategy the following research question was studied:

RQ: Is a fuzzy expert system that considers testing environments and contexts more cost-effective than the other ART strategies presented to date at choosing the most cost-effective regression testing techniques across a system lifetime?

To investigate this research question, a fuzzy expert system for ART, called FESART, was developed. Then an empirical study was conducted to investigate the cost-effectiveness of FESART, and compare the cost-benefit results with the cost-benefit results of the AHP and fuzzy AHP strategies. This chapter the experiment's objects, variables, setup and procedure, and results in more detail.

5.1. Objects of Analysis

This experiment utilizes the same object programs as the first empirical study in this work. For more information on the object programs refer to Chapter 4.

5.2. Variables and Measures

This experiment utilizes one independent variable and one dependent variable described in the next sections.

5.2.1. Independent Variable

The independent variable is the *test case prioritization technique application mapping strategy* which assigns, to a specific sequence of versions, $S_i, S_{i+1}, \ldots S_j$, for system S, specific test case prioritization techniques. There are three strategies used

in this study. Each strategy chooses one of four prioritization techniques (total block coverage, additional block coverage, random order, and original order). Total block coverage sorts test cases by the order of the number of blocks they cover. Additional block coverage selects a test case that yields the greatest block coverage, adjusts the coverage information for the remaining test cases to indicate their coverage for the blocks not yet covered, and then repeats this process until all blocks are covered by at least one test case. Random order is the average of a number of runs (in this experiment 30 runs) with random ordering of test cases. Original order executes the test cases in the order given in the test script provided with the object programs. The three strategies used are as follows:

- AHP: Uses the ART strategy utilizing the AHP method across all versions. This strategy is used as the baseline strategy.
- Fuzzy AHP: Uses the ART strategy utilizing the fuzzy AHP method across all versions.
- FESART: A new ART strategy that utilizes a fuzzy expert system to select the best technique across all versions.

5.2.2. Dependent Variable and Measures

The dependent variable in the study is the *relative cost-benefit value*. This value is calculated using the EVOMO economic model [17] (described in Chapter 2). The costs considered in the EVOMO model account for the costs related to the regression testing techniques, but they do not consider the cost related to applying the ART strategy. In the previous studies, this cost was not considered in the cost-benefit calculations. However, to evaluate the approaches properly, the cost of applying the strategy should be considered. In this research, this cost was added to the cost-benefit calculations. The EVOMO model was modified to include one

additional cost: C_{ART} (the cost of applying the ART strategy). C_{ART} is a cost related to human effort, so it is applied in the equations in the same way as other costs related to human effort (by capturing the cost related to the salary of the engineer who performed the activity).

The cost and benefit calculations for the EVOMO model are measured in dollars. To determine the *relative cost-benefit* of the ART strategy, S, with respect to baseline strategy, *base* (in this experiment the strategy utilizing the AHP method is used for the base), the following equation is used:

$$(\text{Benefit}_S - \text{Cost}_S) - (\text{Benefit}_{base} - \text{Cost}_{base}) \tag{28}$$

When this equation is applied, positive values indicate that S is beneficial compared to the *base*, and negative values indicate otherwise.

5.3. Experiment Setup and Procedure

The experimental setup is similar to that of the setup described in Chapter 4.3. In order to measure costs such as delayed fault detection, the object programs needed to contain some faults. This study used mutation faults and mutant groups created by the *ByteME* (Bytecode Mutation Engine) tool from the SIR Repository [16]. Each mutant group contained, at most, 10 mutants that were randomly selected per version.

To evaluate the cost-effectiveness of FESART, this study implemented FESART using the same four cost criteria as the last study (the cost criteria are described in Chapter 4.3). Two decision makers, each having seven years of industry experience in software development, rated the input criteria for each version of every object program. The decision makers rated each criterion for each prioritization technique in terms of their cost on a scale from 1 to 9, with 9 being considered a very high cost.

The fuzzy inference engine in the FESART system implemented in this study utilized 67 rules. The process for building these rules is as follows. All of the possible combinations of membership functions for each of the criteria were considered. There are four input variables, and three membership functions were used for each one, so there were 81 unique combinations. Each combination was evaluated and assigned an appropriate output set. Then, the rules were studied to see if any of them could be combined or eliminated. The rule set was reduced to 67 rules. The following example demonstrates how some of the rules were eliminated. In the original rule set, the following three rules existed:

IF CF is H and CD is H and CR is H and CA is H then Cost is $H3$.
IF CF is H and CD is H and CR is H and CA is A then Cost is $H3$.
IF CF is H and CD is H and CR is H and CA is L then Cost is $H3$.

In these rules the values for CF, CD, and CR were high and they have stronger impact than CA, so it ended up that for each possible value for CA, the output set was still $H3$. The value of CA did not have any impact on these particular rules, so those three rules can be reduced into the following rule:

IF CF is H and CD is H and CR is H then Cost is $H3$

A complete rule set for FESART is shown in Table 18. To understand further how the rules were developed, consider a subset of the rules: Rules 1 through 7. Because CF and CD were determined to have the most impact on the cost-benefit calculations and are classified as a high cost in these rules, the final cost is categorized in the different high cost output sets ($H1$, $H2$, and $H3$). When CR and CA have a higher cost, then the final cost is determined on the higher end of the high output sets ($H2$ or $H3$) and when CR and CA have a low cost, the cost is on the lower end of the high cost output sets ($H1$).

The fuzzy output set is defuzzified to provide crisp output to the decision maker to utilize in the decision maker process. This is done through the defuzzification

Table 18. Fuzzy Rules for FESART

1. IF CF is H and CD is H and CR is H then Cost is H3	2. IF CF is H and CD is H and CR is A and CA is H then Cost is H3
3. IF CF is H and CD is H and CR is A and CA is A then Cost is H2	4. IF CF is H and CD is H and CR is A and CA is L then Cost is H2
5. IF CF is H and CD is H and CR is L and CA is H then Cost is H2	6. IF CF is H and CD is H and CR is L and CA is A then Cost is H1
7. IF CF is H and CD is H and CR is L and CA is L then Cost is H1	8. IF CF is H and CD is A and CR is H and CA is H then Cost is H2
9. IF CF is H and CD is A and CR is H and CA is A then Cost is H2	10. IF CF is H and CD is A and CR is H and CA is L then Cost is H1
11. IF CF is H and CD is A and CR is A then Cost is H1	12. IF CF is H and CD is A and CR is L and CA is H then Cost is H1
13. IF CF is H and CD is A and CR is L and CA is A then Cost is A2	14. IF CF is H and CD is A and CR is L and CA is L then Cost is A2
15. IF CF is H and CD is L and CR is H and CA is H then Cost is H1	16. IF CF is H and CD is L and CR is H and CA is A then Cost is H1
17. IF CF is H and CD is L and CR is H and CA is L then Cost is A2	18. IF CF is H and CD is L and CR is A and CA is H then Cost is A2
19. IF CF is H and CD is L and CR is A and CA is A then Cost is A2	20. IF CF is H and CD is L and CR is A and CA is L then Cost is A1
21. IF CF is H and CD is L and CR is L then Cost is A1	22. IF CF is A and CD is H and CR is H and CA is H then Cost is H2
23. IF CF is A and CD is H and CR is H and CA is A then Cost is H1	24. IF CF is A and CD is H and CR is H and CA is L then Cost is H1
25. IF CF is A and CD is H and CR is A and CA is H then Cost is H1	26. IF CF is A and CD is H and CR is A and CA is A then Cost is H1
27. IF CF is A and CD is H and CR is A and CA is L then Cost is A2	28. IF CF is A and CD is H and CR is L and CA is H then Cost is A2
29. IF CF is A and CD is H and CR is L and CA is A then Cost is A2	30. IF CF is A and CD is H and CR is L and CA is L then Cost is A1
31. IF CF is A and CD is A and CR is H and CA is H then Cost is H1	32. IF CF is A and CD is A and CR is H and CA is A then Cost is A2
33. IF CF is A and CD is A and CR is H and CA is L then Cost is A2	34. IF CF is A and CD is A and CR is A and CA is H then Cost is A2
35. IF CF is A and CD is A and CR is A and CA is A then Cost is A1	36. IF CF is A and CD is A and CR is A and CA is L then Cost is A1
37. IF CF is A and CD is A and CR is L and CA is H then Cost is A1	38. IF CF is A and CD is A and CR is L and CA is A then Cost is A1
39. IF CF is A and CD is A and CR is L and CA is L then Cost is L3	40. IF CF is A and CD is L and CR is H then Cost is A1
41. IF CF is A and CD is L and CR is A and CA is H then Cost is A1	42. IF CF is A and CD is L and CR is A and CA is A then Cost is L3
43. IF CF is A and CD is L and CR is A and CA is L then Cost is L3	44. IF CF is A and CD is L and CR is L and CA is H then Cost is L3
45. IF CF is A and CD is L and CR is L and CA is A then Cost is L2	46. IF CF is A and CD is L and CR is L and CA is L then Cost is L2
47. IF CF is L and CD is H and CR is H and CA is H then Cost is A2	48. IF CF is L and CD is H and CR is H and CA is A then Cost is A2
49. IF CF is L and CD is H and CR is H and CA is L then Cost is A1	50. IF CF is L and CD is H and CR is A then Cost is A1
51. IF CF is L and CD is H and CR is L and CA is H then Cost is A1	52. IF CF is L and CD is H and CR is L and CA is A then Cost is L3
53. IF CF is L and CD is H and CR is L and CA is L then Cost is L3	54. IF CF is L and CD is A and CR is H and CA is H then Cost is A1
55. IF CF is L and CD is A and CR is H and CA is A then Cost is A1	56. IF CF is L and CD is A and CR is H and CA is L then Cost is L3
57. IF CF is L and CD is A and CR is A and CA is H then Cost is L3	58. IF CF is L and CD is A and CR is A and CA is A then Cost is L3
59. IF CF is L and CD is A and CR is A and CA is L then Cost is L2	60. IF CF is L and CD is A and CR is L then Cost is L2
61. IF CF is L and CD is L and CR is H and CA is H then Cost is L3	62. IF CF is L and CD is L and CR is H and CA is A then Cost is L2
63. IF CF is L and CD is L and CR is H and CA is L then Cost is L2	64. IF CF is L and CD is L and CR is A and CA is H then Cost is L2
65. IF CF is L and CD is L and CR is A and CA is A then Cost is L2	66. IF CF is L and CD is L and CR is A and CA is L then Cost is L1
67. IF CF is L and CD is L and CR is L then Cost is L1	

process. Many different defuzzification techniques have been proposed. This experiment utilized the center of gravity method because it is the most widely accepted and is regarded as being accurate [61, 65].

The output provided by FESART, helped the decision maker determined which technique should be used for each version of every program. Output from FESART with lower numbers represent a lower cost for using that technique for that particular software version.

This study accounts for time constraints in the same way the study in Chapter 4 does (by assigning random time constraints from 25%, 50%, and 75% to a set of four runs). A more detailed description of this process is provided in Chapter 4.3. When the decision makers evaluated the input criteria, they took the time constraints into consideration when providing their input. Also, the cost-benefit calculations were calculated by measuring the appropriate costs for each technique when the testing process was shortened by the assigned time constraint.

5.4. Threats to Validity

This section discusses the construct, internal, and external threats to the validity of our study.

5.4.1. Construct Validity

The construct validity could be threatened by the number of criteria considered in this experiment. Four criteria were considered, but additional criteria could be considered which could change the results. Also, FESART was developed utilizing 67 rules. A fuzzy expert system with fewer or more rules could be developed and potentially change the results.

5.4.2. Internal Validity

The ratings from the decision maker were entered into the fuzzy expert system built in MATLAB. Each of the produced outputs from the expert system were double-checked, but the possibility of small marginal human errors still exists due to the ratings being hand entered into MATLAB.

5.4.3. External Validity

The external validity of this experiment could be limited in a couple ways. First, three triangular membership functions for the input set and eight triangular membership functions for the output set were used. Many different numbers of membership functions could be considered, as well as different types (e.g. gaussian, trapezoidal, etc.). The results cannot be generalized because the type and number of membership functions used are not representative of those functions. Also, two decision makers were used in this study. The backgrounds and experience levels for the decision makers could differ from those of professional programmers, so we cannot generalize the findings of this study. To reduce this risk decision makers who have several years of industry experience were selected.

5.5. Data and Analysis

This section presents the results of the experiment. Each version for every program is assigned a random time constraint (25%, 50%, or 75%). This procedure is performed four times giving four runs of random time constraint levels for every version for all programs. The cost-benefit results for the four runs for each program are shown in Tables 19 - 25. The AHP-based ART strategy is used as the baseline strategy in the relative cost-benefit calculation, so it is not displayed in the tables.

Table 19. Experiment 2: Relative Cost-Benefit Results for *ant* (Runs 1 and 2)

Run 1				
	Fuzzy AHP-1	Fuzzy AHP-2	FESART-1	FESART-2
v1	52	-5	88	37
v2	-4	-2	29	39
v3	104	108	140	149
v4	-3	-15	31	25
v5	-3	-5	30	39
v6	-3	0	29	36
v7	-3	-3	30	35
v8	-3	2	29	36
Total	**137**	**80**	**406**	**396**
Run 2				
v1	-7	-3	30	173
v2	-4	-2	29	39
v3	-47	-2	-13	38
v4	11	-2	46	39
v5	-3	-5	30	38
v6	-3	0	29	36
v7	86	-3	71	36
v8	-3	2	29	36
Total	**30**	**-15**	**251**	**435**

The cost-benefit values for each programs are displayed in a separate table (*galileo* and *ant* are split into two tables because of they have a larger number of versions than the other programs). The data in each table show the cost-benefit values, in dollars, with respect to the AHP-based ART strategy (baseline) defined in Chapter 5.2.2 for that program. Positive cost-benefit values indicate greater cost-

Table 20. Experiment 2: Relative Cost-Benefit Results for *ant* (Runs 3 and 4)

Run 3				
	Fuzzy AHP-1	Fuzzy AHP-2	FESART-1	FESART-2
v1	-6	-5	81	36
v2	-6	-2	31	39
v3	104	108	140	149
v4	26	28	60	67
v5	-3	-5	30	39
v6	-3	0	30	37
v7	-3	-3	29	34
v8	173	176	206	211
Total	**282**	**297**	**607**	**612**
Run 4				
v1	52	-5	88	37
v2	-4	-2	34	40
v3	-5	-47	29	38
v4	-3	-15	31	25
v5	-3	-5	29	38
v6	-3	0	30	38
v7	-3	-3	30	35
v8	97	100	130	136
Total	**128**	**23**	**401**	**387**

benefits than the baseline strategy, and negative values indicate fewer cost-benefits than the baseline strategy. Two decision makers were used in this study. Results for the first decision maker for the ART strategy utilizing fuzzy AHP are labeled Fuzzy AHP-1, and Fuzzy AHP-2 is used for the second decision maker. Similarly, the results for the first decision maker for FESART are labeled FESART-1, and FESART-2 is used for the second decision maker.

When examining the total cost-benefit values for all of the versions of a program, FESART is more cost-effective than the other ART strategies for all four runs of all five programs. One of the biggest reasons FESART was consistently more cost-effective than the other two strategies was because the cost of applying the ART strategy was much lower. If the strategies picked the same prioritization technique, FESART would be more cost-effective because the cost of applying the strategy was

Table 21. Experiment 2: Relative Cost-Benefit Results for *jmeter*

Run 1				
	Fuzzy AHP-1	Fuzzy AHP-2	FESART-1	FESART-2
v1	29	-3	65	32
v2	-3	-2	34	36
v3	6	-2	44	34
v4	-3	-2	36	130
v5	58	59	93	95
Total	**87**	**50**	**272**	**327**
Run 2				
v1	-7	42	76	78
v2	-3	-2	33	36
v3	6	-2	44	34
v4	-3	-2	35	39
v5	5	7	32	43
Total	**-2**	**43**	**220**	**230**
Run 3				
v1	29	-3	65	32
v2	-3	-2	34	36
v3	-5	-2	32	32
v4	-3	-2	36	40
v5	5	7	32	43
Total	**23**	**-2**	**199**	**183**
Run 4				
v1	-7	-3	31	34
v2	-3	-2	34	36
v3	-5	-2	32	32
v4	-3	-2	36	40
v5	58	60	93	95
Total	**40**	**51**	**226**	**237**

lower than that of the AHP and fuzzy AHP methods. For example, for version 2 of run 1 for *jmeter*, all strategies chose the additional block coverage technique as the most cost-effective technique. However, the costs of applying the AHP and fuzzy AHP strategies were higher, so FESART produced a better result. FESART took, on average, half the time of the other two strategies. Another reason FESART was more cost effective when looking at the total cost-benefit calculation was because it chose the most cost-effective technique more frequently than the other strategies. A

Table 22. Experiment 2: Relative Cost-Benefit Results for *xml-security*

Run 1				
	Fuzzy AHP-1	Fuzzy AHP-2	FESART-1	FESART-2
v1	-5	-3	33	36
v2	-3	-2	36	40
v3	-2	0	34	36
Total	**-10**	**-5**	**103**	**112**
Run 2				
v1	-5	-3	31	34
v2	-3	-2	36	35
v3	-2	0	35	37
Total	**-10**	**-5**	**102**	**106**
Run 3				
v1	-5	-3	33	35
v2	-3	-2	37	41
v3	-2	0	34	36
Total	**-10**	**-5**	**104**	**112**
Run 4				
v1	-5	-3	31	34
v2	-3	-2	36	40
v3	-2	0	35	37
Total	**-10**	**-5**	**102**	**111**

probable reason for this is because some of the expert knowledge needed to choose the most cost-effective technique is placed in the fuzzy expert system, unlike the other strategies, where all the expert knowledge is required by the decision maker. When looking at the total cost-benefit values between AHP and fuzzy AHP, fuzzy AHP was frequently more cost-effective than AHP.

To summarize the results and show them visually, a series of bar graphs is presented by averaging the total cost-benefit values for the four runs in Figure 13. The figure shows the average totals for FESART being largely more cost-effective than the other strategies. In particular, in the case of *galileo*, the differences between FESART and others are more outstanding than other programs. Also, the totals show fuzzy AHP being more cost-effective than AHP for four of the five programs. For *xml-security*, the average results are negative, which means the strategy is less

Table 23. Experiment 2: Relative Cost-Benefit Results for *nanoxml*

Run 1				
	Fuzzy AHP-1	Fuzzy AHP-2	FESART-1	FESART-2
v1	-7	4	42	33
v2	-3	-2	35	35
v3	-5	-2	32	34
v4	47	49	87	91
v5	-3	-3	33	34
Total	**29**	**46**	**229**	**227**
Run 2				
v1	-7	-39	33	34
v2	-3	-2	34	35
v3	-5	-2	32	34
v4	47	49	87	91
v5	-3	-3	34	35
Total	**29**	**3**	**220**	**229**
Run 3				
v1	-6	3	40	40
v2	-3	-2	34	35
v3	-5	-2	33	35
v4	47	49	87	91
v5	-3	-3	33	34
Total	**30**	**45**	**227**	**235**
Run 4				
v1	-7	4	42	33
v2	-3	-2	35	35
v3	-5	-2	33	35
v4	47	49	87	91
v5	-3	-3	34	35
Total	**29**	**46**	**230**	**229**

cost-effective than the baseline. The results for fuzzy AHP are negative for *xml-security* because the techniques chosen were often the same as the techniques chosen by the AHP strategy, and the cost of applying the fuzzy AHP strategy was slightly higher, making it less cost-effective for those cases.

The total cost-benefit values provide a general trend about the data, but one version's cost-benefit value can skew the results. Thus, the results for the individual versions are examined. Examining the results of each version provides more insight

Table 24. Experiment 2: Relative Cost-Benefit Results for *galileo* (Runs 1 and 2)

Run 1				
	Fuzzy AHP-1	Fuzzy AHP-2	FESART-1	FESART-2
v1	-7	-5	30	37
v2	-4	-2	32	39
v3	-5	-2	30	38
v4	-3	69	103	110
v5	-3	-3	29	-9
v6	-3	0	30	38
v7	-3	-3	30	35
v8	-3	-2	29	36
v9	-4	-2	33	39
v10	-5	-2	30	39
v11	-50	-49	31	38
v12	3	3	36	44
v13	135	139	169	176
v14	-3	-1	193	200
v15	-3	-2	30	38
Total	**42**	**138**	**835**	**898**
Run 2				
v1	-6	-5	-31	36
v2	-4	-2	33	39
v3	-5	-2	29	38
v4	-3	69	103	110
v5	-3	-3	30	38
v6	-3	0	30	38
v7	-3	-3	29	34
v8	-3	-2	30	37
v9	-4	-2	32	39
v10	-5	-2	30	39
v11	-120	115	31	38
v12	38	-3	71	38
v13	135	139	169	176
v14	-3	90	70	127
v15	-3	-2	30	38
Total	**8**	**387**	**686**	**865**

about the ART strategies. First, examining it this way shows that FESART is consistently more cost-effective than the other two strategies across all versions for all programs except for a few cases (e.g., version 7 of run 2 for the first decision

Table 25. Experiment 2: Relative Cost-Benefit Results for *galileo* (Runs 3 and 4)

Run 3				
	Fuzzy AHP-1	Fuzzy AHP-2	FESART-1	FESART-2
v1	-7	-5	30	38
v2	-4	-2	33	39
v3	-5	-2	30	39
v4	-3	-2	31	38
v5	-3	-3	30	38
v6	-3	0	47	54
v7	-3	-3	30	36
v8	-3	-2	30	37
v9	-4	-2	34	-55
v10	-5	-2	29	38
v11	-120	115	31	38
v12	1	1	30	43
v13	174	177	206	213
v14	-3	-89	116	36
v15	-3	-2	29	36
Total	**9**	**179**	**736**	**668**
Run 4				
v1	-6	-5	-31	36
v2	-4	-2	-35	40
v3	-5	-2	29	38
v4	-3	-2	-12	38
v5	-3	-3	-17	39
v6	-3	0	19	37
v7	-3	-3	29	34
v8	-3	-2	30	38
v9	-4	-2	32	39
v10	-5	-2	30	38
v11	-120	115	32	155
v12	38	-3	71	38
v13	174	177	207	214
v14	-3	90	70	127
v15	-3	-2	30	37
Total	**47**	**354**	**484**	**948**

maker for *ant*). Second, comparing AHP and fuzzy AHP for individual versions, it is found that the overall trend is different from what is observed with the total value comparison. Although the fuzzy AHP is frequently more cost-effective than AHP

in regards to the total cost-benefit values for programs, there are many versions which are less cost-effective than AHP because the cost of applying the fuzzy AHP strategy is slightly higher than the AHP strategy. A tool was used for the AHP calculations which made them go quicker, but no such tool was available for fuzzy AHP. Instead, code had to be written in MATLAB to do the calculations and then the pairwise comparisons had to be manually entered into MATLAB to calculate the results. This process is slightly more time consuming than the tool used for AHP, so when the two strategies choose the same technique, the fuzzy AHP strategy is slightly less cost-effective.

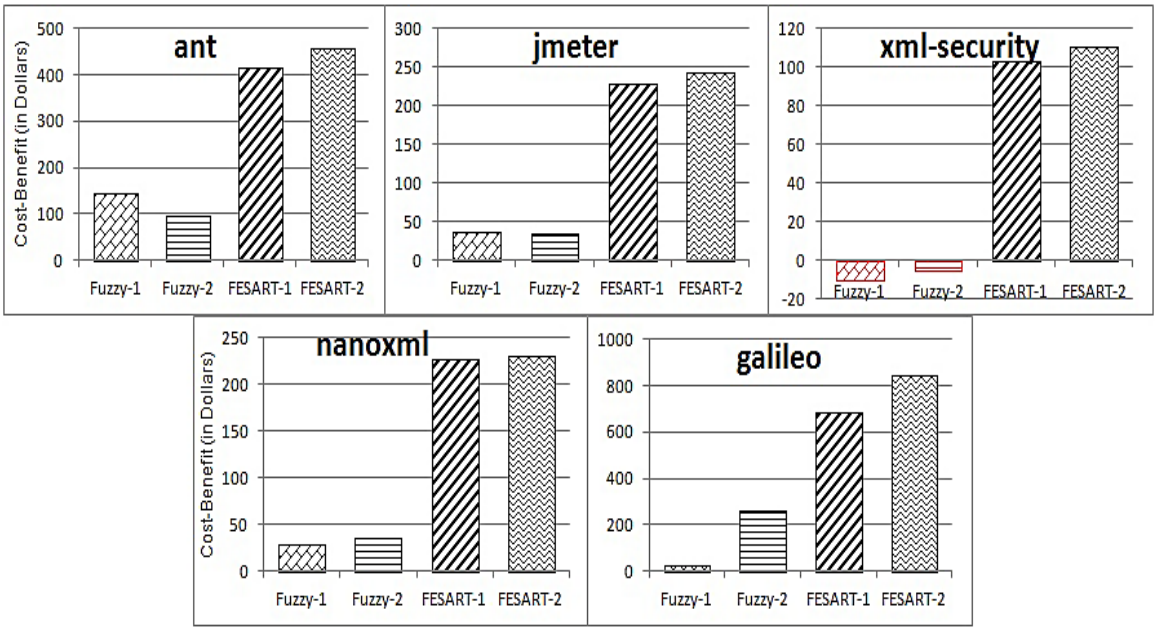


Figure 13. Experiment 2: Average Cost-Benefit Totals

5.6. Discussion and Implications

FESART was developed to address the limitations of the previously proposed ART strategies utilizing the AHP and fuzzy AHP methods. This section discusses how FESART effectively addresses these limitations as well as the implications of the experiment's results for researchers and practitioners.

5.6.1. FESART Strategy Results

Developing a strategy that does not require pairwise comparisons eliminates some of the problems with the previous ART strategies. First, the issue of inconsistent comparisons is eliminated. By not requiring the decision maker to rank the alternatives compared to other alternatives, the risk of inconsistency in the rankings is eliminated. Second, FESART is less time consuming than a strategy requiring pairwise comparisons. The number of weights needed by the decision maker in FESART is reduced from the number of weights required by pairwise comparisons, making it less time-consuming for the decision maker. Third, the decreased input required by the decision maker helps address the issue of scalability. Fewer weights required by the decision maker makes FESART more scalable than the other strategies.

By addressing these limitations, the results indicate that FESART is more cost-effective than the previously proposed ART strategies. One of the biggest contributors to FESART being more cost-effective is the reduction in the amount of time it takes to apply the strategy. Because the time required by the FESART strategy was less than the other two strategies, if the strategies chose the same technique, FESART was more cost-effective. In addition, in some situations, FESART chose a more cost-effective technique than the other strategies, making the total cost-savings even greater. One possible explanation for FESART choosing a more cost-effective technique than the other strategies is that some of the expert knowledge is placed in the rule base in the fuzzy expert system, so the amount of knowledge required by the decision maker is not as high as it is with the previous strategies.

5.6.2. Understanding the Implications of the Results

The findings of this experiment provide practical implications for practitioners and researchers in software engineering. These results show that FESART, that considers cost criteria related to testing environments and contexts, improves the

cost-effectiveness of that regression testing session.

Savings of hundreds of dollars presented in this study may be unimportant. In practice, however, regression testing could take days or even weeks, so if results such as those presented in this study scale up, savings of the dollar amount may be substantial. For instance, in this study, small/medium sized programs were used, but typical industrial applications have millions of lines of code (e.g., a popular accounting software, Quickbooks, has over 80,000 files and ten million lines of code). Thus, if they were to apply the FESART strategy, the savings would be far greater than those presented in this study.

Further, the costs associated with the defects escaped into the released system could impact the results greatly. This study considered ordinary defects (not severe defects). A survey by Shull et al. [62] suggests that the effort to find and fix severe defects is far more expensive than non-severe defects. Thus, if different types of defects are taken into account, the FESART approach could have an even greater impact on cost savings related to early fault detection.

6. EMPIRICAL STUDY 3: A COMPARATIVE ART STUDY

The previous experiments outlined in Chapters 4 and 5 have shown cost-savings in regression testing by utilizing the ART strategies presented in this work. The fuzzy AHP approach to ART showed greater cost-savings than the traditional AHP method, but still required pairwise comparisons. The FESART strategy was consistently more cost-effective than the AHP and fuzzy AHP strategies. One major contribution to that was because of its low cost. FESART does not require the time-consuming pairwise comparisons. This raises the question of whether the success of FESART can be attributed to the time saved by not requiring pairwise comparisons. Could another less time-consuming approach produce the same cost-savings? Further, with the different strategies presented in this research, a comparative study of each of the strategies presented in this work would help researchers and practitioners to understand the trade-offs and cost-effectiveness of each of the strategies. This study investigates the following research question:

RQ: How do each of the ART strategies perform when the cost of applying the strategy is considered?

In addition to this research question, this study investigates whether a low cost strategy is always more cost-effective than a strategy with a higher cost. To investigate the research question, this chapter presents an empirical study. First, another low cost approach to ART utilizing the Weighted Sum Model (WSM) is developed to investigate the differences between two low cost strategies (with FESART being the second low cost approach). Then, a statistical analysis of each of the proposed strategies is performed to evaluate if there is a statistical significance between the cost-benefit calculations of the ART strategies studied in

this research. This chapter discusses the objects of analysis, variables and measures, results, analysis and discussion of the study.

6.1. Objects of Analysis

This experiment utilizes the same object programs as the previous studies presented in Chapters 4 and 5. For more information on the object programs refer to Chapter 4.

6.2. Variables and Measures

One independent variable and one dependent variable are utilized in this study. These variables are described in the next sections.

6.2.1. Independent Variable

The independent variable is the *test case prioritization technique application mapping strategy* which assigns, to a specific sequence of versions, $S_i, S_{i+1}, \ldots S_j$, for system S, specific test case prioritization techniques. There are four strategies used in this study. Each strategy chooses one of four prioritization techniques (total block coverage, additional block coverage, random order, and original order. Each of these techniques have been described previously in this work). The four mapping strategies used in this study are as follows:

- AHP: Uses the ART strategy utilizing the AHP method across all versions. This strategy is used as the baseline strategy.
- Fuzzy AHP: Uses the ART strategy utilizing the fuzzy AHP method across all versions.
- FESART: Uses the ART strategy that utilizes a fuzzy expert system to select the best technique across all versions.
- WSM: A new ART strategy that utilizes the Weighted Sum Model (WSM) to select the best technique across all versions.

6.2.2. Dependent Variable and Measures

The dependent variable in the study is the *relative cost-benefit value*. This value is calculated using the EVOMO economic model [17] (described in Chapter 2).

The costs considered in the EVOMO model account for the costs related to the regression testing techniques, but they do not consider the cost related to the ART strategy. In the second empirical study in this research (presented in Chapter 5), the EVOMO model was extended to consider this cost. This study uses that extended version of the EVOMO model to incorporate the cost of applying the ART strategy into the study.

The cost and benefit calculations for the EVOMO model are measured in dollars. To determine the *relative cost-benefit* of the ART strategy, S, with respect to baseline strategy, *base* (in this experiment the strategy utilizing the AHP method is used for the base), the following equation is used:

$$(\text{Benefit}_S - \text{Cost}_S) - (\text{Benefit}_{base} - \text{Cost}_{base}) \tag{29}$$

When this equation is applied, positive values indicate that S is beneficial compared to the *base*, and negative values indicate otherwise.

6.3. Experiment Setup and Procedure

The experimental setup is similar to that of the setup of the previous two empirical studies in this research. The study required the ability to measure costs such as delayed fault detection, so the object programs needed to contain some faults. To obtain faults in the object programs, mutation faults and mutant groups were created by the *ByteME* (Bytecode Mutation Engine) tool from the SIR Repository [16]. Each mutant group contained, at most, 10 mutants that were randomly selected per version.

To identify the most cost-effective technique for a specific software version, each ART strategy considered four cost criteria. These criteria are discussed in Chapter 3. The criteria were evaluated for all versions of each object program for each of the strategies by two decision makers. The decision makers each have seven years of industry experience in software development. The process for evaluating the criteria for the fuzzy AHP and FESART strategies are provided in Chapters 4 and 5 respectively. For the new strategy considered in this experiment, the WSM, the decision makers each weighted the criteria first. For this strategy, they each proportionally weighted the criteria to total up to 1. Then, each of the alternatives were weighted in terms of the criteria on a scale from 1 to 9. A scale from 1 to 9 was chosen for this strategy to keep it consistent with the other strategies to limit the threats to validity when comparing the results for each strategy. For this strategy, it was decided that a higher rating meant it performed better (or had a lower cost). So the higher the weighted sum, the better the technique was in the evaluation.

This study accounts for time constraints as the previous studies in Chapter 4 and 5 did. However, this study differs from the previous studies, because in order to perform a statistical analysis on the data, more runs were needed. The process of assigning the time constraints was the same. For each run, a random time constraint of 25%, 50%, and 75% was assigned. In the previous experiments, four runs were used. In this experiment twenty different runs were used in order to be able to perform a statistical analysis.

6.4. Threats to Validity

This section discusses the construct, internal, and external threats to the validity of our study.

6.4.1. Construct Validity

The construct validity could be threatened by the number of criteria and alternatives considered in each of the strategies. Four criteria and four alternatives

were considered, but additional criteria and alternatives could be considered which could change the results.

6.4.2. Internal Validity

Each of the strategies required collecting input from the decision makers and entering them into a tool and/or performing calculations on the input. A small margin of error could exist during these processes for each of the strategies.

6.4.3. External Validity

This study compares the cost-benefit results of the three new strategies presented in this research. Although the study reveals some trade-offs between the strategy, there are many more strategies which could be created preventing the results to be generalized.

6.5. Data and Analysis

This section presents the results of the experiment. Each version for every program is assigned a random time constraint (25%, 50%, or 75%). This procedure is performed twenty times giving twenty runs of random time constraint levels for every version for all programs. These twenty runs are used to determine if there is a statistical significance between the strategies studied in the experiment. Because the results showed some differences between programs and decision makers, the statistical analysis was conducted separately for each program for each decision maker.

6.5.1. Statistical Analysis Procedure

The statistical analysis was performed using the Statistical Analysis System (SAS) [1]. To perform the statistical analysis, this research begins with the Kruskal-Wallis test. This test was chosen because the data did not meet the assumptions required for the ANOVA procedure. The ANOVA procedure assumes that the data

[1] http://www.sas.com

is distributed normally with no severe outliers. The data from this experiment did not meet this assumption. When assumptions for ANOVA are not successfully met, the Kruskal-Wallis test is a commonly used method.

The Kruskal-Wallis test begins by ranking the data in terms of its rank to the overall data set. The smallest value gets a rank of 1, the second-smallest gets a rank of 2, etc. If there are data that are the same, the tied observations get average ranks. For example, if there were four identical values occupying the second, third, fourth, and fifth smallest places, these rankings would get averaged, and each would receive a ranking of 3.5.

Then, the sum of the ranks is calculated for each group and a test statistic which considers the variance of the ranks among the groups is calculated. This test statistic is approximately chi-square distributed, which means that the probability of getting a particular value by chance is the p-value corresponding to the chi-square.

The results of the Kruskal-Wallis test include the chi-square and the p-value. The p-value represents the probability that the differences of the data could have occurred by chance. Traditionally, an accepted boundary for the p-value is .05. When the p-value is less than or equal to .05 it is accepted that the differences found by the test are too large to have occurred by chance. In other words, it can be said that the differences found by the test are definitely due to the differences in the data being studied, and not due to chance.

The results for the Kruskal-Wallis test performed on the cost-benefit calculation for twenty runs for each version of every program are presented in Table 26.

For each of the programs, the p-values are less than .05, so the results indicate there is a statistical significance between the groups. The results of the Kruskal-Wallis test do not reveal *which* groups (in this case, which strategy) have a statistical difference, only that one does exist between at least one group. In order to draw

conclusions on each strategy, further testing needs to be performed to investigate which strategy is statistically different from the other strategies.

To examine which strategy (or strategies) are statistically different from the others, a multiple comparison method is required. This work uses the Bonferroni method. The Bonferroni method was applied to each program and the results for each program are presented in the following subsections.

6.5.2. Results for *ant*

The box plots for cost-benefit calculations for twenty runs are shown in Figure 14. In this box plot (and each of the remaining box plots for the other programs), FESART-1 represents the cost-benefit results for FESART for the first decision maker, and FESART-2 represents the cost-benefit results for the second decision maker. Fuzzy AHP-1 is for the results for the fuzzy AHP strategy for the first decision maker, and Fuzzy AHP-2 shows the results for the fuzzy AHP strategy for the second decision maker. WSM-1 represents the cost-benefit results for the first decision maker, and WSM-2 shows the results for the second decision maker. The cost-benefit results for the traditional AHP method are used as the baseline strategy, and so they are not shown in the box plots.

The box plots for *ant* show FESART (for both decision makers) is noticeably more cost-effective than the other two strategies for both decision makers. Between the decision makers, the second decision maker (which will now be referred to as

Table 26. Kruskal-Wallis Results

Program	DM1		DM2	
	Chi-Square	p-value	Chi-Square	p-value
ant	34.36	$< .0001$	39.41	$< .0001$
jmeter	41.05	$< .0001$	39.38	$< .0001$
xml-security	20.25	$< .0001$	39.90	$< .0001$
nanoxml	47.02	$< .0001$	46.50	$< .0001$
galileo	39.92	$< .0001$	45.12	$< .0001$

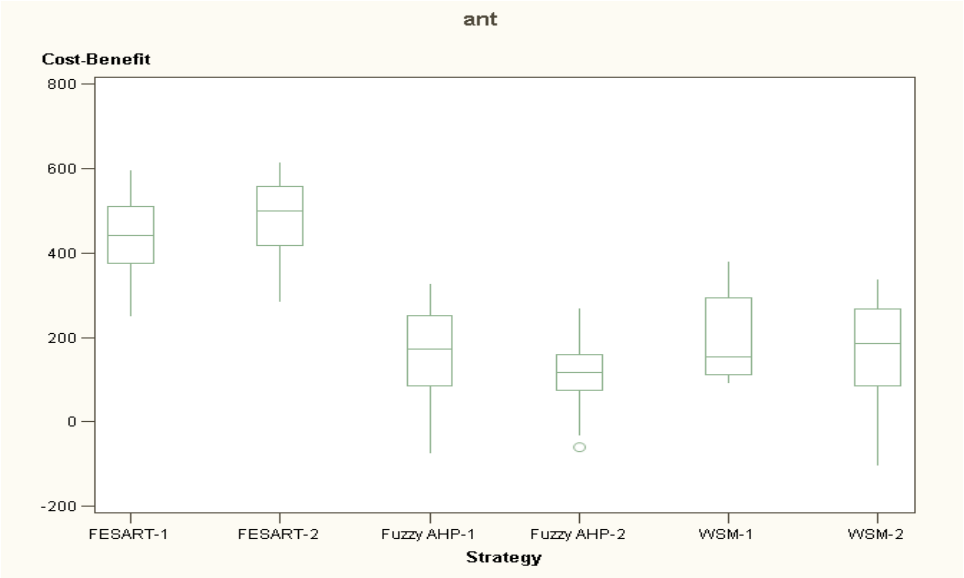


Figure 14. Box Plots for *ant*

DM2) shows greater cost-benefits than the first decision maker (DM1).

Unlike the results for FESART, the results are not as straight-forward when comparing the fuzzy AHP and WSM strategies. The results for Fuzzy AHP-1 and WSM-2 appear to be pretty normally distributed, but the results for Fuzzy AHP-2 has a severe outlier, and the upper half of the values for WSM-1 are much more spread out than the lower half.

The WSM for DM1 appears to be just slightly more cost-effective than the fuzzy AHP strategy for DM1 and the WSM for DM2. There is not a noticeable difference between these three. However, the results for fuzzy AHP-2 is noticeably different from the other three strategies (fuzzy AHP-1, WSM-1, and WSM-2). The cost-benefit results for fuzzy AHP-2 has a severe outlier (represented by the circle below the lower quartile) and the results for WSM-1 show the upper fifty percent having greater variability than the lower fifty percent. When comparing the median scores, the WSM-2 performed the best (among the four) with fuzzy AHP-1 closely following. The median scores for WSM-1 and fuzzy AHP-2 are fairly close together,

with the scores for WSM-1 being slightly higher. From the boxplot it appears no significant conclusions can be made between fuzzy AHP and WSM.

To investigate the observations made by looking at the box plots, the Bonferroni method was applied to the results for *ant*. The results of the Bonferroni method are shown in Table 27. Each group (in this case ART strategy) is given a group letter. Strategies with the same group letter indicate they are not statistically different.

For *ant* both decision makers have the same groupings. The results show that the FESART strategy is grouped differently than fuzzy AHP and WSM strategies, so there is a statistical difference between FESART and the other two strategies. The fuzzy AHP and WSM strategies share the same group, so there is no statistical difference between those two strategies.

Table 27. Bonferroni Results for *ant*

	ant			
	DM1		DM2	
Strategy	Mean	Group	Mean	Group
FESART	438.06	A	487.48	A
Fuzzy AHP	197.12	B	163.04	B
WSM	163.11	B	115.14	B

6.5.3. Results for *jmeter*

The box plots for *jmeter* are shown in Figure 15. For *jmeter* the results differ between the decision makers and will be discussed separately. Like *ant*, the box plots show FESART as the most cost-effective strategy. Unlike *ant* there is also a clear cost-savings for the WSM for DM1 when compared to the fuzzy AHP strategy (for both decision makers) and to the WSM for DM2. There appears to be very little difference between the cost-benefit calculations for fuzzy AHP-1, fuzzy AHP-2, and WSM-2. Also, the results for WSM-2 show some severe outliers.

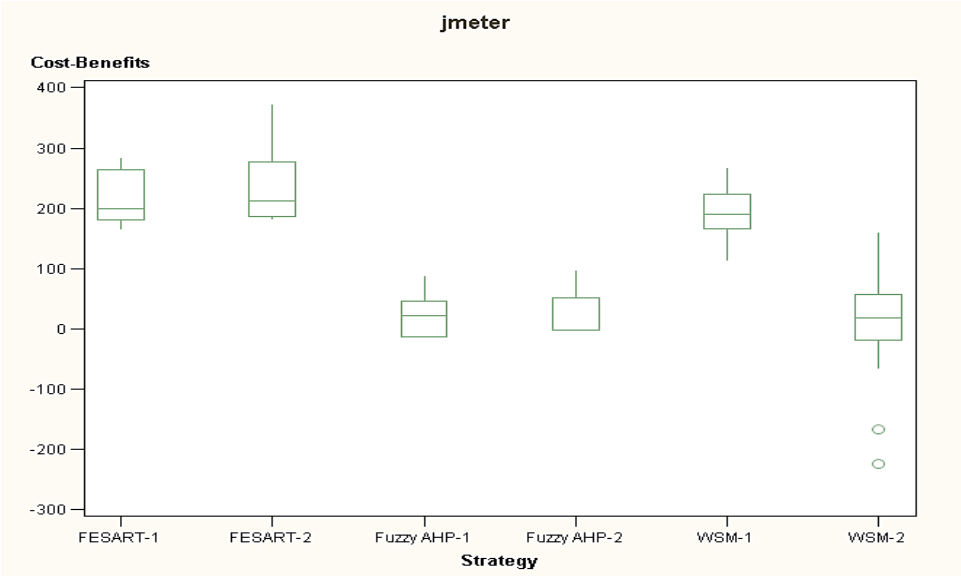


Figure 15. Box Plots for *jmeter*

The Bonforroni results are shown in Table 28. For DM1, there is a statistical difference between the FESART and fuzzy AHP strategies, but not between FESART and WSM. These two strategies (FESART-1 and WSM-1) are statistically more cost-effective than fuzzy AHP-1. For DM2, there is a statistical difference between FESART and the other two strategies with FESART being more cost-effective than the other two strategies, and the other two strategies being placed in the same group.

One interesting thing to note about the results for *jmeter* is that there is quite a bit of variance between the two decision makers for the WSM strategy. For DM1, there was not enough difference in the cost-benefits between FESART and WSM-1 to even be statistically significant. However, for DM2 there was quite a bit of difference, and it *was* statistically significant. The WSM has been known to be a somewhat volatile decision making strategy, with the results being strongly dependent on the decision maker, so results like these would make sense for the WSM.

Table 28. Bonferroni Results for *jmeter*

	jmeter			
	DM1		DM2	
Strategy	Mean	Group	Mean	Group
FESART	215.88	A	231.13	A
Fuzzy AHP	23.97	B	26.28	B
WSM	188.60	A	6.59	B

6.5.4. Results for *xml-security*

The box plots for *xml-security* are shown in Figure 16. Like *jmeter* the WSM shows quite a bit of variance. This is especially true when looking at the results for DM1. Some values in the boxplot for WSM-1 are actually higher than the values for both decision makers for FESART. The median value for WSM-1, however, is lower than the median value for each of the other strategies. With the exception of a few of the extreme values for WSM-1, like *ant* and *jmeter*, FESART is the most cost-effective strategy.

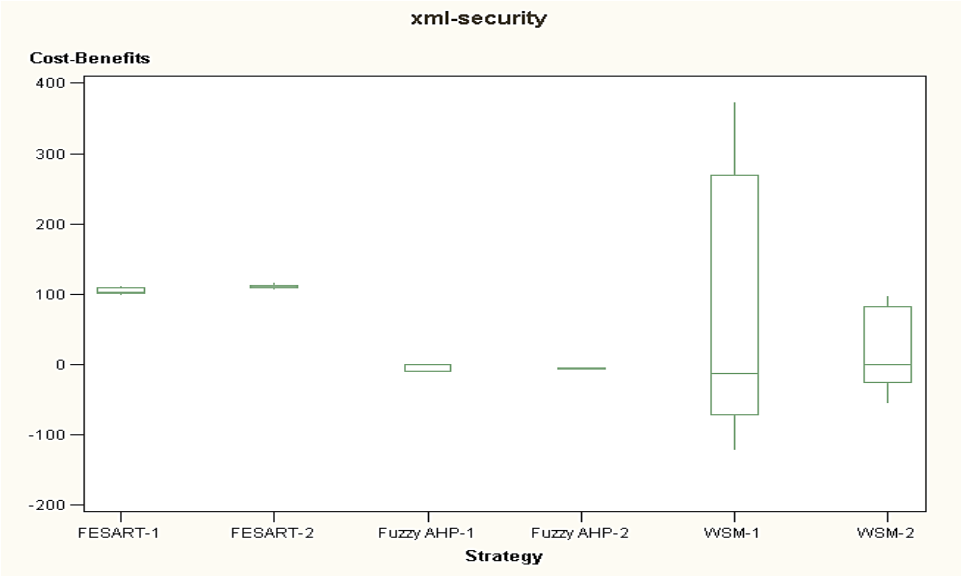


Figure 16. Box Plots for *xml-security*

The Bonferroni results for *xml-security* are shown in Table 29. For DM1 there is a statistical significance between FESART and fuzzy AHP, but there is not enough difference between FESART and WSM or WSM and fuzzy AHP to say there is a statistical difference. For DM2, there is a statistical significance between FESART and the other two strategies, but not between Fuzzy AHP and WSM.

Table 29. Bonferroni Results for *xml-security*

	xml-security			
	DM1		DM2	
Strategy	Mean	Group	Mean	Group
FESART	104.5	A	111.15	A
Fuzzy AHP	-6.99	B	-5.38	B
WSM	62.48	B A	13.45	B

6.5.5. Results for *nanoxml*

Box plots for *nanoxml* are shown in Figure 17. The box plots for *nanoxml* show there are quite a few outliers. Only FESART-2 and WSM-2 do not include any outlier. From the box plots, the cost-benefits for FESART is greater than any of the other strategies for both decision makers. And like *jmeter* and *xml-security*, the values for WSM are pretty inconsistent. WSM-1 shows a pretty low outlier, and there is quite a difference between the values for WSM-1 and WSM-2.

The Bonferroni results for *nanoxml* are shown in Table 30. Once again the groups the strategies were placed in is different between the two decision makers. Unlike any of the previous programs, however, there is a statistical difference between each of the strategies for both decision makers. FESART-1 and FESART-2 were placed in Group A, being the most cost-effective strategy. For DM1, fuzzy AHP is placed in the next group, Group B, while the WSM strategy is placed in Group B for DM2. One interesting thing to note is because of the wide variance in values for WSM for DM1, it was actually group different than WSM-2 although the median value for both was very close.

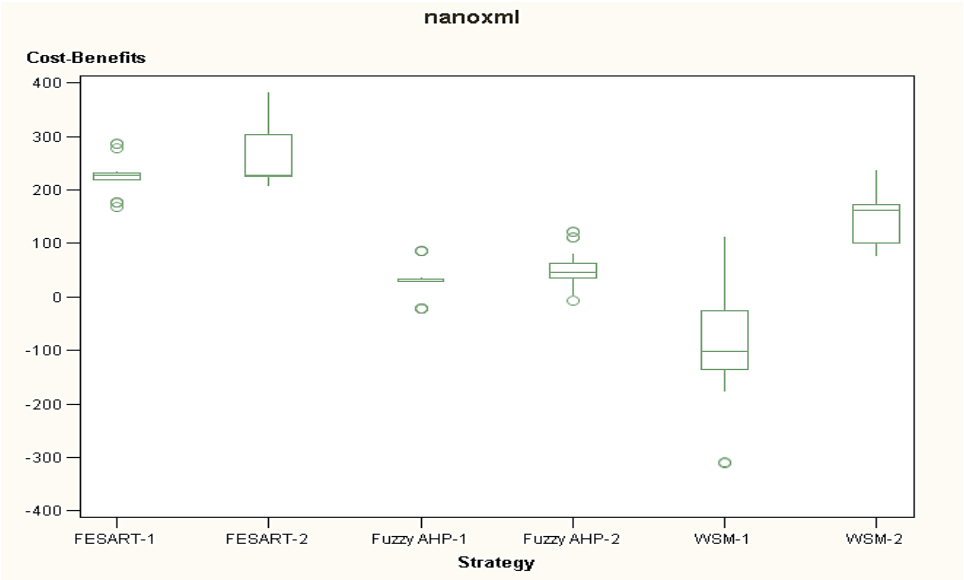


Figure 17. Box Plots for *nanoxml*

6.5.6. Results for *galileo*

Box plots for *galileo* are shown in Figure 18. The values for FESART show to be the most cost-effective again. The values here are higher than values of the other programs because *galileo* contains more versions, which shows that as the life of a program gets longer, the cost savings of utilizing the ART strategies is greater. The box plots for the WSM and fuzzy AHP strategies show the values for the WSM are higher than the fuzzy AHP strategy for DM2, but shows little difference between the values for DM1.

Table 30. Bonferroni Results for *nanoxml*

	nanoxml			
	DM1		DM2	
Strategy	Mean	Group	Mean	Group
FESART	225.06	A	262.32	A
Fuzzy AHP	28.57	B	51.43	C
WSM	-94.54	C	149.86	B

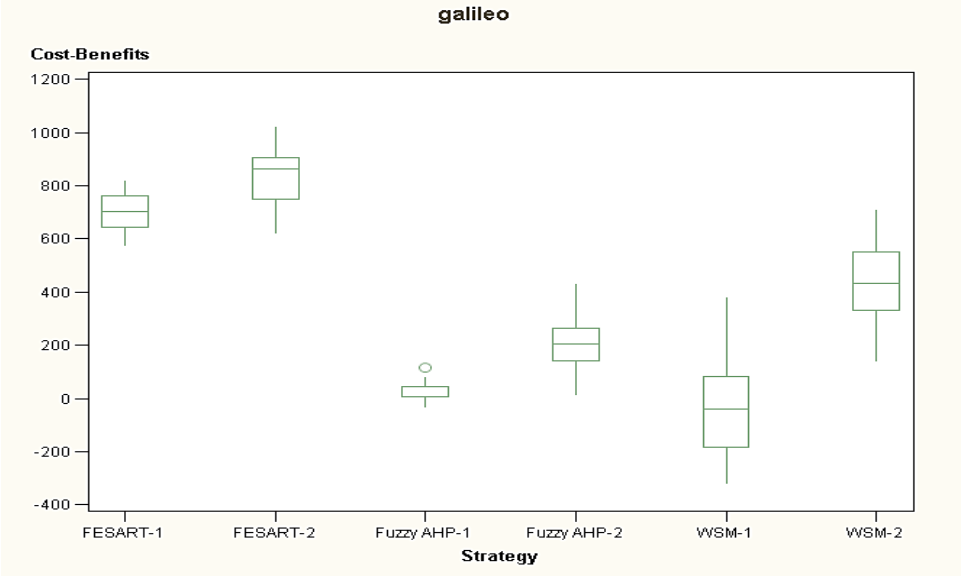


Figure 18. Box Plots for *galileo*

The Bonferroni results are shown in Table 31. The results show there is a statistical difference between FESART and the other strategies for both decision makers. FESART is statistically the most cost-effective strategy. The results for the remaining two strategies are different between the two decision makers. For DM1, there is no statistical difference between the fuzzy AHP and WSM strategies, but there is a statistical difference between these two strategies for DM2.

Table 31. Bonferroni Results for *galileo*

	galileo			
	DM1		DM2	
Strategy	Mean	Group	Mean	Group
FESART	703.84	A	827.04	A
Fuzzy AHP	31.79	B	219.9	C
WSM	-41.61	B	434.87	B

6.5.7. General Results for All Programs

Although the results differ between the programs and decision makers, to attempt to gain a general trend of the data, Figure 19 represents the average between the decision makers of the mean value of twenty runs for each program.

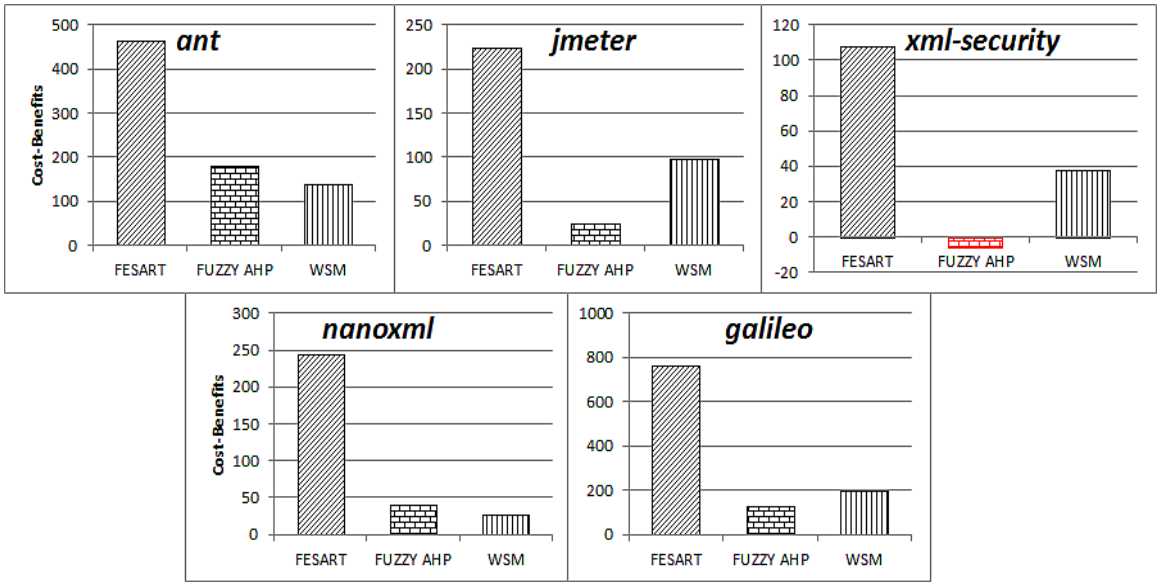


Figure 19. Average Means for All Programs

For each of the programs, FESART was the most cost-effective strategy. The WSM strategy came in second, being more cost-effective than the fuzzy AHP strategy for three of the five programs (*jmeter*, *xml-security*, and *galileo*). The fuzzy AHP strategy was more cost-effective than the WSM for two of the five programs (*ant* and *nanoxml*).

6.6. Discussion

The goal of this study was to investigate the effectiveness of each of the proposed strategies when the cost of applying the strategy is considered. FESART was developed and evaluated in Chapter 5, and the results indicated FESART was more cost-effective than any of the other proposed ART strategies. One of the big reasons FESART was more cost-effective than the other strategies was because the cost of applying the strategy was lower than the other strategies. This research investigates, then, whether any low cost strategy would be as effective as FESART. To study this question, a new ART strategy utilizing the WSM was developed and evaluated.

The results of the study indicated that the low-cost strategy was, at times, more cost-effective than the fuzzy AHP strategy, and consistently more cost-effective than the AHP strategy. However, the results also showed that FESART was consistently more cost-effective than the WSM strategy. These results indicate that the cost-effectiveness of FESART is not solely related to its low cost. Other possible factors that contribute to FESART's increased performance is the expert knowledge-base used to generate the fuzzy rule set and the use of fuzzy logic to handle imprecision in the input provided by the decision makers. The variability in the cost-benefit results for the WSM strategy between decision makers show that the WSM is more sensitive to the input provided by the decision maker. For each program, the results for FESART only had minimal variation, but the variation was large for the WSM.

The further investigate the research question, of how each of the strategies perform with the cost of applying the strategy is considered, a statistical analysis was performed for each of the ART strategies discussed in this research. The results of the statistical analysis had a few different trends. One trend that was consistent among all programs and each decision maker was that FESART was statistically different from the other strategies, being the most cost-effective strategy. Other trends that were shown in the study was the volatility of the WSM strategy. The results for the WSM differed, not only among the different programs, but also between the two decision makers. Another trend, which is somewhat related to the volatility of the WSM, was that there was not a consistent statistical difference between the WSM and fuzzy AHP strategies. For example, the results for *ant* showed no statistical difference between the WSM and fuzzy AHP strategies for either decision maker. The results for *jmeter* showed a statistical difference between the results for the WSM for DM1, but not for DM2. The results for *nanoxml* showed a statistical difference for both decision makers, but for DM1 fuzzy AHP

was grouped as a more cost-effective strategy and for DM2 the WSM was grouped as the more cost-effective strategy. Because of the varying performance between the fuzzy AHP and WSM strategies, no sound conclusions can be made as to which strategy is more cost-effective between these two strategies.

When compared to the traditional AHP strategy, the results show that each of the strategies presented in this work are consistently more cost-effective. In this study, the AHP strategy was used as the baseline strategy, so the cost-benefit calculation for all twenty runs for the AHP strategy would be zero, and therefore is not shown on the box plots. The box plots show that the majority of the values for each of the strategies are above zero, meaning they are more cost-effective than the AHP (baseline) strategy.

The implications of this study have significant impact for researchers and practitioners. This study investigates the cost-benefit performance of each of the strategies while accounting for the cost of applying the strategies. This study also provides statistical data supporting that the strategies proposed in this work are more cost-effective than the previously proposed ART strategy utilizing the AHP method. Researchers and practitioners will be able to use this data, and the strategies proposed in this work, to perform their regression testing sessions.

7. CONCLUSION AND FUTURE WORK

This dissertation has presented new ART strategies which address the limitations of the previous strategy and evaluated them through a series of empirical studies. In particular, three strategies were developed. One strategy utilized the fuzzy AHP method to address the issue of the results from the AHP method being subjective to the judgments made by the decision maker. A second strategy used a fuzzy expert system to obtain the benefits of a strategy which does not require pairwise comparisons. A third strategy utilized the Weighted Sum Model (WSM) to investigate the effectiveness of a simple, low-cost strategy for ART.

Each of the empirical studies evaluated the strategies in terms of their cost-benefit results. The results of the studies indicated that the new strategies presented in this work provide for greater cost-savings for regression testing than the previously proposed strategy. The studies revealed some helpful trends, such as the FESART strategy, overall, appears to be the most cost-effective strategy of each of the strategies presented in this research. One major contribution to that is because of its low cost. However, the results of the studies also show that any low cost strategy would not produce the same results (when the WSM was evaluated, there was a wide variance in its cost-benefit results, and often it was the least cost-effective of each of the strategies presented in this work).

7.1. Merits and Impact of This Research

The results found in this research provides important practical implications for both researchers and practitioners. Since the cost of regression testing is very high, strategies to reduce the cost are very important. This work provides multiple strategies that can help reduce the cost of regression testing. The dollar amounts shown in this work may seem insignificant to some, but if the strategies were utilized in practice, the savings could be substantial. For instance, only small and medium

sized programs were used in the studies in this work. Industrial applications are very large, many of them containing millions of lines of code (the programs used in this study were only in the thousands to tens of thousands). If ART strategies were used on these large applications, the cost savings could be much larger than those presented in these studies. Also, these studies only considered ordinary faults. Studies have shown that the costs associated with severe defects are much more costly than ordinary defects. Considering severe defects could greatly increase the cost-benefit calculations.

Additional contributions of this work include the empirical studies provided evaluating the performance of different decision making strategies, some of which were severely lacking in the literature (i.e. fuzzy AHP vs traditional AHP). The empirical studies performed in this work will provide researchers with data demonstrating the success (or lack of success) of the varying methods used in the context of regression testing. Further, the empirical studies provide data for researchers and practitioners to use when considering adopting ART strategies. The data will help them see how ART strategies may be beneficial and help them choose an appropriate strategy to meet their regression testing needs.

7.2. Future Work

Although this research has provided some important contributions, there are some areas in which this work could be studied in the future. First, these studies only considered four test case prioritization techniques. Additional prioritization techniques could be considered that may have a greater cost-savings than the ones considered in this research. Also, other regression testing techniques, such as test case selection and test case minimization techniques, could be considered.

Another important area for future work is to investigate the scalability of the strategies by incorporating larger programs. Each of the experiments in this work used five small to medium sized Java programs. Even with small and medium sized

programs there were cost-savings shown in the experiments. If the ART strategies scale with larger programs the cost-savings would be even greater.

This work considered four cost criteria for each strategy. Future work could consider additional cost criteria. Further, additional empirical studies which evaluate other factors that could contribute to the cost-effectiveness of regression testing techniques could be conducted which would give more data to use when deciding on an appropriate regression testing technique for a particular regression testing session. This additional knowledge could then be used by the decision maker or even integrated into the ART strategies.

With the information provided in this work and any future work performed in the areas just mentioned, there is strong potential for large cost-savings in regression testing through the use of ART strategies.

REFERENCES

[1] W. Abdelmoez, M. Ibrahim, M.A. Omar, and H.H. Ammar, *Sensitivity analysis of maintainability-based risk factors for software architectures*, 8th International Conference on Informatics and Systems (INFOS), IEEE, 2012, pp. SE–29.

[2] A. Adeli and M. Neshat, *A fuzzy expert system for heart disease diagnosis*, Proceedings of International Multi Conference of Engineers and Computer Scientists, Hong Kong, vol. 1, 2010.

[3] V. Ahl, *An experimental comparison of five prioritization methods*, Master's Thesis, School of Engineering, Blekinge Institute of Technology, Ronneby, Sweden (2005).

[4] N. Ahmad and P.A. Laplante, *Software project management tools: Making a practical decision using AHP*, Software Engineering Workshop, 2006. SEW'06. 30th Annual IEEE/NASA, IEEE, 2006, pp. 76–84.

[5] M.J. Arafeen and H. Do, *Adaptive regression testing strategy: An empirical study*, 22nd International Symposium on Software Reliability Engineering (ISSRE), 2011, pp. 130–139.

[6] ———, *Test case prioritization using requirements-based clustering*, Proceedings of the 6th International Conference on Software Testing, Verification, and Validation, IEEE, 2013.

[7] J. Benítez, X. Delgado-Galván, J.A. Gutiérrez, and J. Izquierdo, *Balancing consistency and expert judgment in ahp*, Mathematical and Computer Modelling (2011), vol. 54, no. 7, pp. 1785–1790.

[8] M. Bevilacqua and M. Braglia, *The analytic hierarchy process applied to maintenance strategy selection*, Reliability Engineering & System Safety (2000), vol. 70, no. 1, pp. 71–83.

[9] J. Buckley, *Fuzzy hierarchical analysis*, Fuzzy sets and systems (1985), vol. 17, no. 3, pp. 233–247.

[10] R. Carlson, H. Do, and A. Denton, *A clustering approach to improving test case prioritization: An industrial case study*, Proceedings of the 27th International Conference on Software Maintenance, IEEE, 2011, pp. 382–391.

[11] D. Chang, *Applications of the extent analysis method on fuzzy AHP*, European journal of operational research (1996), vol. 95, no. 3, pp. 649–655.

[12] C. Cheng, K. Yang, and C. Hwang, *Evaluating attack helicopters by AHP based on linguistic variable weight*, European Journal of Operational Research (1999), vol. 116, no. 2, pp. 423–435.

[13] H. Do, S. Elbaum, and G. Rothermel, *Supporting controlled experimentation with testing techniques: An infrastructure and its potential impact*, International Journal on Empirical Software Engineering (2005), vol. 10, no. 4, pp. 405–435.

[14] H. Do, S. Mirarab, L. Tahvildari, and G. Rothermel, *The effects of time constraints on test case prioritization: A series of controlled experiments*, IEEE Transactions on Software Engineering (2010), vol. 26, no. 5, pp. 593–617.

[15] H. Do and G. Rothermel, *An empirical study of regression testing techniques incorporating context and lifecycle factors and improved cost-benefit models*, Proceedings of the ACM SIGSOFT Symposium on Foundations of Software Engineering, November 2006, pp. 141–151.

[16] ———, *On the use of mutation faults in empirical assessments of test case prioritization techniques*, IEEE Transactions on Software Engineering (2006), vol. 32, no. 9, pp. 733–752.

[17] ———, *Using sensitivity analysis to create simplified economic models for regression testing*, Proceedings of the International Conference on Software Testing and Analysis, July 2008, pp. 51–62.

[18] H. Do, G. Rothermel, and A. Kinneer, *Prioritizing junit test cases: An empirical assessment and cost-benefits analysis*, Empirical Software Engineering (2006), vol. 11, no. 1, pp. 33–70.

[19] S. Elbaum, P. Kallakuri, A. Malishevsky, G. Rothermel, and S. Kanduri, *Understanding the effects of changes on the cost-effectiveness of regression testing techniques*, Software testing, verification and reliability (2003), vol. 13, no. 2, pp. 65–83.

[20] S. Elbaum, A. Malishevsky, and G. Rothermel, *Incorporating varying test costs and fault severities into test case prioritization*, Proceedings of the 23rd International Conference on Software Engineering, IEEE Computer Society, 2001, pp. 329–338.

[21] ———, *Test case prioritization: A family of empirical studies*, IEEE Transactions on Software Engineering (2002), vol. 28, no. 2, pp. 159–182.

[22] S. Elbaum, G. Rothermel, S. Kanduri, and A. Malishevsky, *Selecting a cost-effective test case prioritization technique*, Software Quality Journal (2004), vol. 12, no. 3, pp. 185–210.

[23] M. Fasanghari and G. Montazer, *Design and implementation of fuzzy expert system for tehran stock exchange portfolio recommendation*, Expert Systems with Applications (2010), vol. 37, no. 9, pp. 6138–6147.

[24] P. Fishburn, *Letter to the editoradditive utilities with incomplete product sets: Application to priorities and assignments*, Operations Research (1967), vol. 15, no. 3, pp. 537–542.

[25] A. Gunawan and H. Lau, *Master physician scheduling problem*, Journal of the Operational Research Society (2012), vol. 64, no. 3, pp. 410–425.

[26] M. Hadjimichael, *A fuzzy expert system for aviation risk assessment*, Expert Systems with Applications (2009), vol. 36, no. 3, pp. 6512–6519.

[27] S. Hatton, *Early prioritisation of goals book series*, Advances in Conceptual Modeling Foundations and Applications (2007), vol. 4802, no. 235–244, pp. 517–526.

[28] H. Hemmati and L. Briand, *An industrial investigation of similarity measures for model-based test case selection*, IEEE 21st International Symposium on Software Reliability Engineering (ISSRE), IEEE, 2010, pp. 141–150.

[29] H. Hsu and A. Orso, *Mints: A general framework and tool for supporting test-suite minimization*, 31st International Conference on Software Engineering (ICSE), IEEE, 2009, pp. 419–429.

[30] S. Kad and V. Chopra, *Fuzzy logic based framework for software development effort estimation*, Research Cell: An International Journal of Engineering Sciences (2011), vol. 1, pp. 330–342.

[31] A. Kadhim, A. Alam, and H. Kaur, *Design and implementation of fuzzy expert system for back pain diagnosis*, International Journal of Innovative Technology & Creative Engineering, IJITCE (2011), no. 9, pp. 16–22.

[32] M. Kamal and A. Al-Harbi, *Application of the AHP in project management*, International Journal of Project Management (2001), vol. 19, pp. 19–27.

[33] J. Karlsson, C. Wohlin, and B. Regnell, *An evaluation of methods for prioritizing softawre requirements*, Information and Software Technology (1998), vol. 39, pp. 939–947.

[34] L. Karlsson, T. Thelin, B. Regnell, P. Berander, and C. Wohlin, *Pair-wise comparisons versus planning game partitioningexperiments on requirements prioritisation techniques*, Empirical Software Engineering (2007), vol. 12, no. 1, pp. 3–33.

[35] M. Kazemifard, A. Zaeri, N. Ghasem-Aghaee, M.A. Nematbakhsh, and F. Mardukhi, *Fuzzy emotional cocomo ii software cost estimation (fecsce) using multi-agent systems*, Applied Soft Computing (2011), vol. 11, no. 2, pp. 2260–2270.

[36] G. Khorasani, M. Yadollahi, and A. Tatari, *Implementation of mcdm methods in road safety managment*, Proceedings of the International Conferencce on Transport, Civil, Architecture and Environment engineering (ICTCAEE), 2012, pp. 30–35.

[37] Y. Kim and O.L. De Weck, *Adaptive weighted-sum method for bi-objective optimization: Pareto front generation*, Structural and multidisciplinary optimization (2005), vol. 29, no. 2, pp. 149–158.

[38] S. Lee, *Using fuzzy AHP to develop intellectual capital evaluation model for assessing their performance contribution in a university*, Expert Systems with Applications (2010), vol. 37, no. 7, pp. 4941–4947.

[39] A. Leitner, M. Oriol, A. Zeller, I. Ciupa, and B. Meyer, *Efficient unit test case minimization*, Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering, ACM, 2007, pp. 417–420.

[40] H. Leung and L. White, *A cost model to compare regression test strategies*, Proceedings. Conference on Software Maintenance, IEEE, 1991, pp. 201–208.

[41] L. Lin and C. Wang, *On consistency and ranking of alternatives in uncertain ahp*, Natural Science (2012), vol. 4, no. 5, pp. 340–348.

[42] O. López-Ortega and M. Rosales, *An agent-oriented decision support system combining fuzzy clustering and the ahp*, Expert Systems with Applications (2011), vol. 38, no. 7, pp. 8275–8284.

[43] M. J. Harrold and A. Orso, *Retesting software during development and maintenance*, Proceedings of the International Conference on Software Maintenance: Frontiers of Software Maintenance, September 2008, pp. 88–108.

[44] A. Malishevsky, G. Rothermel, and S. Elbaum, *Modeling the cost-benefits tradeoffs for regression testing techniques*, Proceedings of International Conference on Software Maintenance, IEEE, 2002, pp. 204–213.

[45] E. Mamdani and S. Assilian, *An experiment in linguistic synthesis with a fuzzy logic controller*, International journal of man-machine studies (1975), vol. 7, no. 1, pp. 1–13.

[46] M.A. Mustafa and J.F. Al-Bahar, *Project risk assessment using the analytic hierarchy process*, IEEE Transactions on Engineering Management (1991), vol. 38, no. 1, pp. 46–52.

[47] A. Nassif, L. Capretz, and D. Ho, *Estimating software effort based on use case point model using sugeno fuzzy inference system*, 23rd International Conference on Tools with Artificial Intelligence (ICTAI), IEEE, 2011, pp. 393–398.

[48] N.F. Pan, *Fuzzy AHP approach for selecting the suitable bridge construction method*, Automation in construction (2008), vol. 17, no. 8, pp. 958–965.

[49] A. Perini, F. Ricca, and A. Susi, *Tool-supported requirements prioritization: Comparing the AHP and CBRank methods*, Information and Software Technology (2009), vol. 51, pp. 1021–1032.

[50] A. Perini, F. Ricca, and A. Susi, *Tool-supported requirements prioritization: Comparing the ahp and cbrank methods*, Information and Software Technology (2009), vol. 51, no. 6, pp. 1021–1032.

[51] B. Qu, C. Nie, B. Xu, and X. Zhang, *Test case prioritization for black box testing*, Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International, vol. 1, IEEE, 2007, pp. 465–474.

[52] X. Qu, M. Cohen, and Rothermel G., *Configuration-aware regression testing: An empirical study of sampling and prioritization*, Proceedings of the International Conference on Software Testing and Analysis, July 2008, pp. 75–86.

[53] G. Rothermel, R.H. Untch, C. Chu, and M. J. Harrold, *Test case prioritization: An empirical study*, Proceedings of the Conference on Software Maintenance, August 1999, pp. 179–188.

[54] T. Saaty and M. Ozdemir, *Why the magic number seven plus or minus two*, Mathematical and Computer Modelling (2003), vol. 38, no. 3, pp. 233–244.

[55] T. L. Saaty, *The analytic hierarchy process*, McGraw-Hill, 1980.

[56] M. Sadiq, S. Ghafir, and M. Shahid, *An approach for eliciting software requirements and its prioritization using analytic hierarchy process*, International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom), IEEE, 2009, pp. 790–795.

[57] A. Schwartz and H. Do, *A fuzzy ahp approach to improve adaptive regression testing strategies*, Tech. Report NDSU-CS-TR-13-001, North Dakota State University, 2013.

[58] ———, *A fuzzy expert system for cost-effective regression testing strategies*, Proceedings of the twenty-ninth IEEE international conference on Software Maintenance, September 2013.

[59] M. Scott and E. Antonsson, *Compensation and weights for trade-offs in engineering design: beyond the weighted sum*, Journal of Mechanical Design (2005), vol. 127, p. 1045.

[60] C. Shen, M.J. Cheng, C.W. Chen, F.M. Tsai, and Y.C. Cheng, *A fuzzy AHP-based fault diagnosis for semiconductor lithography process*, International Journal of Innovative Computing, Information and Control (2011), vol. 7, no. 2.

[61] P. Shill, K. Pal, F. Amin, and K. Murase, *Genetic algorithm based fully automated and adaptive fuzzy logic controller*, International Conference on Fuzzy Systems (FUZZ), IEEE, 2011, pp. 1572–1579.

[62] F. Shull, *What we have learned about fighting defects*, Int'l. Softw. Metrics Symp., 2002.

[63] H. Srikanth, L. Williams, and J. Osborne, *System test case prioritization of new and regression test cases*, International Symposium on Empirical Software Engineering, IEEE, 2005.

[64] A. Srivastava and J. Thiagarajan, *Effectively prioritizing tests in development environment*, Proceedings of the International Symposium on Software Testing and Analysis, July 2002, pp. 97–106.

[65] P. Srivastava, S. Kumar, A.P. Singh, and G. Raghurama, *Software testing effort: An assessment through fuzzy criteria approach*, Journal of Uncertain Systems (2011), vol. 5, no. 3, pp. 183–201.

[66] M. Su, C. Chen, W. Lu, G. Liu, Z. Yang, and B. Chen, *Urban public health assessment and pattern analysis: comparison of four cities in different countries*, Frontiers of Earth Science (2013), pp. 1–8.

[67] C. Sun, *A performance evaluation model by integrating fuzzy ahp and fuzzy topsis methods*, Expert systems with applications (2010), vol. 37, no. 12, pp. 7745–7754.

[68] T. Takagi and M. Sugeno, *Fuzzy identification of systems and its applications to modeling and control*, IEEE Transactions on Systems, Man and Cybernetics (1985), no. 1, pp. 116–132.

[69] A.K. Thurimella and S. Ramaswamy, *On adopting multi-criteria decision-making approaches for variability management in software product lines*, Proceedings of the 16th International Software Product Line Conference-Volume 2, ACM, 2012, pp. 32–35.

[70] E. Triantaphyllou, *Multi-criteria decision making methods*, Springer, 2000.

[71] T. Tuglular and G. Gercek, *Mutation-based evaluation of weighted test case selection for firewall testing*, Fifth International Conference on Secure Software Integration and Reliability Improvement (SSIRI), IEEE, 2011, pp. 157–164.

[72] P. Van Laarhoven and W. Pedrycz, *A fuzzy extension of Saaty's priority theory*, Fuzzy Sets and Systems (1983), vol. 11, no. 1, pp. 199–227.

[73] A. Walcott, M. L. Soffa, G. M. Kapfhammer, and R. S. Roos, *Time-aware test suite prioritization*, Proceedings of the International Conference on Software Testing and Analysis, July 2006, pp. 1–12.

[74] W. E. Wong, J. R. Horgan, S. London, and A. P. Mathur, *Effect of test set minimization on fault detection effectiveness*, Proceedings of the International Conference on Software Engineering, April 1995, pp. 41–50.

[75] K. Wu, Z. Xu, and C. Duan, *Research and implementation of evaluation system model for grid investment based on improved fuzzy-ahp method*, Tenth International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES), IEEE, 2011, pp. 350–353.

[76] Z. Xu, K. Gao, and T. Khoshgoftaar, *Application of fuzzy expert system in test case selection for system regression test*, International Conference on Information Reuse and Integration, IEEE, 2005, pp. 120–125.

[77] S. Yoo and M. Harman, *Regression testing minimisation, selection and prioritisation : A survey*, Software Testing, Verification, and Reliability (2010).

[78] S. Yoo, M. Harman, P. Tonella, and A. Susi, *Clustering test cases to achieve effective and scalable prioritisation incorporating expert knowledge*, Proceedings of the International Conference on Software Testing and Analysis, July 2009, pp. 201–212.

[79] LA Zadeh, *Fuzzy sets*, Information and Control (1965), vol. 8, pp. 338–353.

[80] E.K. Zavadskas, Z. Turskis, J. Antucheviciene, and A. Zakarevicius, *Optimization of weighted aggregated sum product assessment*, Electronics and Electrical Engineering (2012), vol. 122, no. 6, pp. 3–6.

[81] L. Zhang, D. Hao, L. Zhang, G. Rothermel, and H. Mei, *Bridging the gap between the total and additional test-case prioritization strategies*, Proceedings of the 35th International Conference on Software Engineering, ICSE, 2013, pp. 192–201.

[82] Y. Zhang, X. Zhang, X. Zhao, and T. Zhang, *Early effort estimation by ahp: A case study of project metrics in small organizations*, 2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE), vol. 1, IEEE, 2012, pp. 452–456.