

Test case prioritization for components using FANP

¹Shah Nazir, ¹Sara Shahzad, ¹Neelam Mukhtar, ²Humaira Khan, ¹Islam Zada, ¹Muhammad Nazir and ³Rohul Amin

¹Department of Computer Science, University of Peshawar; ²Institute of Management Sciences, Peshawar

³Department of Mathematics, University of Peshawar. {snshahnzr, humairamzai, mnazirsajid}@gmail.com, {sara, raminmath}@upesh.edu.pk, {sameen_gul, islamzada28}@yahoo.com

Abstract: Testing of software components is a critical activity in Component Based Software Engineering. The available candidate off-the-shelf components are considered to be well-tested by their designers. Still their integration within a new system may raise several problems including requirements trade-off in the context of the integrating system. Multiple test cases are generated to test the selected software components from all different aspects. Prioritization of component test case is aimed at deciding the priorities of components test case based on the required test criteria. In the proposed methodology Fuzzy Analytic Network Process (FANP) approach is used for component test case (CTC) prioritization. The method combines fuzzy and ANP process for prioritization of test cases for software components.

[Shah Nazir, Sara Shahzad, Neelam Mukhtar, Humaira Khan, Islam Zada, Muhammad Nazir and Rohul Amin. **Test case prioritization for components using FANP.** *Life Sci J* 2014;11(6s):504-511] (ISSN:1097-8135). <http://www.lifesciencesite.com>. 106

Keywords: Component based software engineering (CBSE), test case prioritization, component testing, Fuzzy ANP

1. Introduction

CBSE is aimed at reusing self-designed and commercial off the shelf (COTS) components for it software system development. When software components are designed for reuse, they are passed through extensive testing, without assuming the context of any integrating system. Testability of software components is also a big issue, faced during component engineering and CBSE. Maintaining testability of software components has been a hot issue of research in the CBSE community [1]. A number of techniques have been identified and devised. The success of components based systems depends upon reliable components and their economical testing with the integrating system [2]. Component testing is performed to evaluate the conformity of component to its planned specifications and is also used to assess the reliability of component. An optimal component testing is the one which gives high probability of finding more errors based on the component execution and integration. With a large set of test cases identified and generated for a thorough testing of the component, priorities of test cases are to be set. This is important to evaluate the cost effectiveness of the test cases and also as the sequence in which the tests are executed also matters [3]. This situation asks for test case prioritization to make proper decisions, so that it becomes a case for making decisions about priorities of test cases. Prioritization is aimed at organizing test cases for implementation [4]. ANP is the extended version of Analytic Hierarchy Process (AHP) used for multi criteria decision problem. It structures the elements of clusters into network. When some attributes exists and weights of these attributes are

derived from expert opinions then ANP is an efficient method in such phenomena. ANP method is used for evaluation security of software components using attributes of ISO/IEC 27002 standard [5]. As AHP and ANP is a suitable multi-criteria decision method, these conventional methods of AHP and ANP however seems to be unproductive in dealing with the complexity and uncertainty during the pair wise comparison of elements. In real life, decision making process stockholder may be unsure about the decision, due to deficient information. So to avoid such situation of deficient information, FANP is used to prioritize component test case prioritization under fuzzy condition. FANP measure inconsistency of uncertain human preferences by suitable consistency index [6]. Fuzzy linguistic are also used for decision analysis of web services [7].

The rest of the paper is organized as follows. Related work is presented in section 2. A brief description to the proposed methodology is presented in section 3. Discussion part of the paper is presented in section 4. The paper is concluded in section 5.

2. Related work

Several diverse methodologies related to software component testing and prioritization has been developed by different researchers. J. Gao et al. [8] tests component based on API test model. The method has been implemented on COMPTest tool which automatically spot out component based API modifications and reusable test cases. D. Giannakopoulou et al. [9] introduce assume guarantee test which test component before it is assembled, as later on the cost of integrated system test is increasing. The authors also explained predictive testing, that uses

component assumption and guarantees to test integrated system. The methodology is explained with the help of two NASA case studies. O. A. L. Lemos et al. [10] presents the comparison of Brazilian Symposium on Software Engineering (SBES) to International Conference in Software Engineering (ICSE) for evaluation studies in software testing research. The survey includes papers of 25 years of SBES. The similar survey analysis has been done for ICSE. Survey results shows that the rate of paper in SBES has significantly increased only 20 percent of the paper, when it is compared to ICSE, it is low. It presents 40 percent of papers. C. Mao and Y. Lu [11] presented an improved version of regression testing which is based on changes in the components to form as a whole. It is a joint method of component developer and user. The method is based on experiment which showed that the proposed regression testing is efficient. K. H. S. Hla et al. used practical swarm optimization algorithm for the prioritization of test cases for embedded real time system retesting [12]. B. Korel et al. Present model base prioritization for a class of modification in which only the source code is modified not the model [13].

Weiqun Zheng and Gary Bundell [14] introduced a new concept of contract for testability and developed a set of important concepts including test contract, effectual contract scope, internal and external test contract. Karambir and R. Rani [15] presented test case prioritization including coverage based and cost oriented test case prioritization technique and measure the effectiveness of prioritization techniques. S. Mohanty et al. [16] performed regression testing for component based software system (CBSS). The states and transitions in component based system are represented by UML state chart diagram. Later on these state chart are converted into component integration graph (CIG) to show their inter communication links. The algorithm takes this CIG as input and produce a prioritize test suit. S. Elbaum, et al. [17] addresses the solution towards the problems which raised from the previous literature. Some of these problems include: a) would the prioritization technique will be effective when intended to modified version, b) what tradeoffs exists between prioritization technique of granularity and course granularity and c) would the integration of measure of error freeness into prioritization progress the usefulness. W. Zheng and G. Bundell [18] introduced a UML software component based testing (SCT), model based software component testing, designed under technique of SCT. The activities in the proposed method is divided into two phases, in the first phase a set of UML based test model is build and in the second phase generate test cases from the set of UML test model. C. C. Michael [19] quantify the situation if components in one

environment has been executed without failure and these component are also executed in the new environment with failure. The author developed bounds on components probability of failure based on its previous behavior in the new environment. S. Nazir et al. [20] worked on the selection of software components which is based upon certain component attributes. T. P. Jacob and T. Ravi proposed a method for prioritization of test cases which firstly check the modified lines. The method have used genetic algorithm for test case prioritization [21].

W.G. Alghabban and M.R.J. Qureshi proposed a component selection framework based on pliability metric for software quality. The method is validated by a sample of online questionnaire [22]. G. Rothermel et al. [23] elaborated different techniques to regression test case prioritization. The technique includes a) test case based on coverage of code component, b) test case based on coverage code component which is not previously covered and c) test case based on estimated ability to expose error in code. Analysis of the experiments shows that the rate of error detection is improved. The method spot out different cost benefit tradeoffs. A. M. P'erez and S. Kaiser proposed an approach for matching delay for multi level test cases [24]. I. Yoon et al. [25] observed that each time testing of new version of component from start is a complex task. The author proposed to perform incremental prioritized compatibility testing. The algorithm in the method computes the difference between new and previous version of release. The method has been evaluated on using five years data of scientific middleware component.

3. Methodology

The proposed methodology uses the concept of Fuzzy analytic network process [26]. The following sections explain the different concepts associated with the methodology and then demonstrate the concept for component test case prioritization is described below.

3.1 Fuzzy sets

A Fuzzy set is an extended form of traditional sets, introduced by Zadeh in 1965 [27]. Fuzzy sets are used in system design to handle uncertainty and imprecision. The degree of membership function of fuzzy set is within range of [0-1] interval. The following diagram shows triangular fuzzy membership function used in the proposed methodology for test case prioritization. In the diagram the X-axis shows the level of degree of relationship that is (low, medium and high), while the Y-axis shows the value of these degree that is between the range of [0, 1]. In Matlab these membership functions are designed for plotting some particular problem.

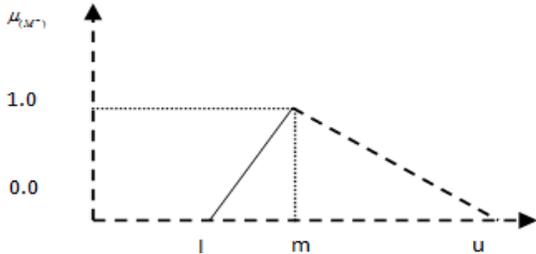


Figure 1. Triangular fuzzy number

3.2 Analytic network process (ANP)

ANP is the generalization of Analytic Hierarchy Process (AHP) developed by Saaty [28]. It is designed as a network of different elements, where each element facilitates measuring of dependencies among different factors of concern for making strategic decisions, and feedback [29]. In an ANP, the elements in the network are arranged according to their relationship as goal, criteria, and alternatives. Elements are structured according to the given criteria and alternatives which lead toward the required goal. Further details of ANP method can be found in Saaty [28], however, the following diagram visually shows the general structure of ANP. The detailed version of this diagram for our proposed methodology is given in Figure 4.

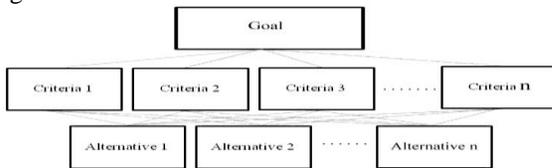


Figure 2. Graphical representation of analytic network process (ANP) [28]

3.3 Fuzzy ANP (FANP)

Traditional AHP/ANP techniques do not clearly help in making decisions in uncertain and complex situations. In a situation when the decision maker or stakeholder goes to uncertainty about the level of weights which they provide. The incomplete information is provided for the prioritization of component test cases. So to overcome such problem, FANP is used under fuzzy condition for the decision about component test case prioritization. FANP is an efficient method for multi criteria decision making problems related to uncertainty.

3.4 FUZZY ANP for component test case prioritization

It is evident that FANP is very helpful when there are interdependencies among the various elements of the network. With the help of triangular fuzzy number Pair wise comparison matrices among various elements are formed. The proposed method for test case prioritization of components incorporates attributes [30, 31].

- Key:
 Controllability (C)
 Observe ability (O)
 Isolatability (I)
 Separation of concern (S)
 Automatibility (A) and
 Heterogeneity (H).

The following table shows some of the test case prioritization techniques (Table 1).

Table 1. Component Test case prioritization techniques

Author	Technique	Focus
J. Gao et al. [8]	Systematic regression testing and tool for software components	Systematic re test method
D. Giannakopoulou et al. [9]	Assume-guarantee testing for software components	Assume guarantee testing for component based system
O. A. L. Lemos et al. [10]	Software testing Evaluation studies in the Brazilian Symposium on Software engineering	Software testing evaluation
C. Mao and Y. Lu [11]	Regression testing based on improved changes in component information	Regression Test
K. H. S. Hla et al [12]	Particle Swarm Optimization to Prioritizing Test Cases	Embedded Real Time Software Retesting
B. Korel et al [13]	Model-based test prioritization heuristics	Simplicity and effectiveness
Weiqun Zheng and Gary Bundell [14]	Contract for Testability	UML based software component testing
Karambir and R. Rani[15]	Overview of test case prioritization	Measure effectiveness of prioritization
S. Mohanty et al. [16]	Software retesting for component using UML state chart diagram	Retesting component based software
S. Elbaum et al. [17]	Prioritizing Test Cases for Regression Testing	Fault detection of test suit
A. M. P'erez and S. Kaiser [24]	First balancing and then compensating delay for multi-level test cases	Facing delay problem
I. Yoon et al. [25]	It is highly costly when to perform compatibility test from the start when a component new version is released	Testing component compatibility

The FANP methodology is step by step presented in the following diagram.

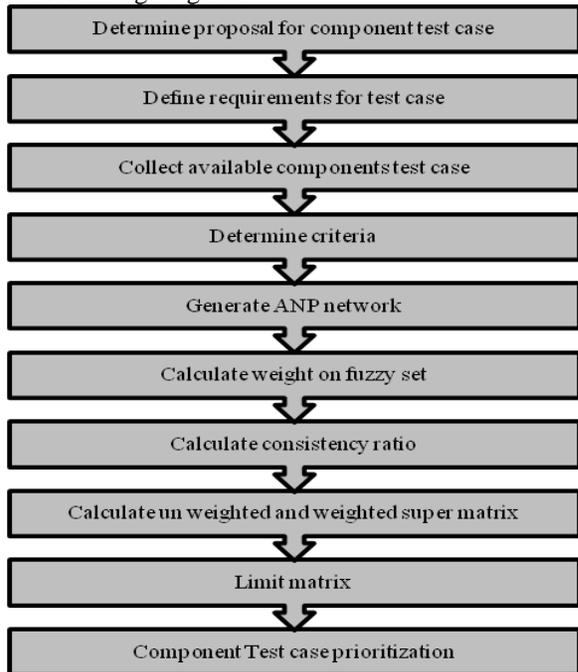


Figure 3. Steps of the proposed method

Further the methodology is explained in the following algorithm.

Step 1. Determine proposal for component test case:

In the very first step the proposal for component test case is to determine that for what propose the system should be designed from the available components. Every attributes of the project for components proposal should be explicitly clear so that to avoid any misunderstanding for the stack holder of the system.

Step 2. Define requirements for test case: In project proposal the user and customer requirements should be explicitly mentioned, so that the project will complete within the specified duration. According to software engineering regulations if changes occur after the designing and development of the system, increases the cost of the system and upsurge toward late delivery of the system.

Step 3. Collect available components test case: When requirements of the customer are well known and clearly understandable, then according to that requirements select the appropriate and suitable components from the available repository.

Step 4. Determine criteria: On the basis of customer requirements criteria and their sub criteria should be mentioned for the proper designing and development of the system.

Step 5. Generate ANP network: In this step the elements of network are drawn and plotted. The elements are plotted in such a way that their feedback among elements should be possible. The relationship among elements should be clearly shown.

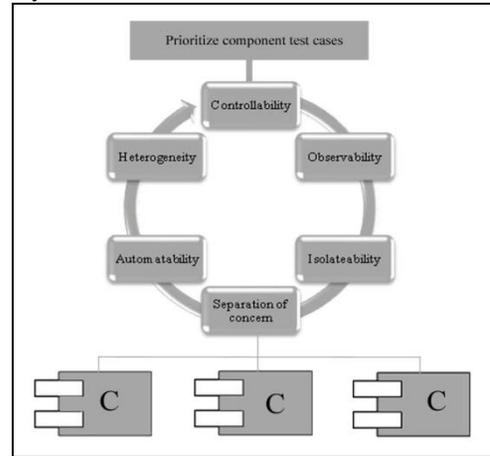


Figure 4. Proposed FANP approach

Step 6. Calculating weights based on fuzzy set: In this step pair wise comparisons of elements are calculated based on fuzzy set.

Table 2. Linguistics scale for weights

Linguistics scale	Fuzzy number	Triangular fuzzy number
Equal	1-	(1,1,2)
Weakly important	3-	(1,3,5)
Essential important	5-	(3,5,7)
Very strong important	7-	(5,7,9)
Absolutely important	9-	(7,9,9)
Intermediate values	2 ⁻ 4 ⁻ 6 ⁻ 8 ⁻	

Triangular fuzzy number is obtained by geometric average of experts opinions of numbers

$$\left(\bar{M} = \bar{l}, \bar{m}, \bar{u} \right) \quad (1)$$

Where

$$\bar{l} = \left(\prod_{i=1}^n l_i \right)^{\frac{1}{n}}$$

$$\bar{m} = \left(\prod_{i=1}^n m_i \right)^{\frac{1}{n}}$$

$$\bar{u} = \left(\prod_{i=1}^n u_i \right)^{\frac{1}{n}}$$

Pair wise comparison of element along with their criteria and alternatives are calculated in the following tables.

Table 3. Fuzzy pair wise comparison for test case of component test case 1

	C	O	I	S	A	H	(E.V.)
C	1,1,2	1,3,5	1,2,3	3,5,7	2,4,6	3,5,7	0.37
O	1,1/3,1/5	1,1,2	1,1,2	1,3,5	1,3,5	3,5,7	0.22
I	1,1/2,1/3	1,1/2,1/3	1,1,2	3,5,7	1,2,3	1,3,7	0.18
S	1/3,1/5,1/7	1,1/3,1/5	1/3,1/5,1/7	1,1,2	1,2,3	1,2,3	0.08
A	1/2,1/4,1/6	1,1/3,1/5	1,1/2,1/3	1,1/2,1/3	1,1,2	2,4,6	0.09
H	1/3,1/5,1/7	1/3,1/5,1/7	1,1/3,1/5	1,1/2,1/3	1/2,1/4,1/6	1,1,2	0.05

CI = 0.09

Step 7. Calculate consistency ratio: Consistency ratio and random consistency ratio are found. Random consistency (RI) table is given by Saaty [32].

$$C_i = \lambda_{max} - n / n - 1 \quad (2)$$

$$CR = CI / RI \quad (3)$$

Table 4. Random consistency index

n	1	2	3	4	5	6	7	8	9
RI	0	0	0.6	0.9	1.1	1.2	1.3	1.4	1.5

Table 5. Fuzzy pair wise comparison for test case of component test case 2

	C	O	I	S	A	H	(E.V.)
C	1,1,2	1,2,3	3,5,7	1,3,5	2,4,6	5,7,9	0.39
O	1,1/2,1/3	1,1,2	1,2,3	1,2,3	1,3,5	3,5,7	0.22
I	1/3,1/5,1/7	1,1/2,1/3	1,1,2	3,5,7	1,2,3	1,3,5	0.17
S	1,1/3,1/5	1,1/2,1/3	1/3,1/5,1/7	1,1,2	1,2,3	1,2,3	0.10
A	1/2,1/4,1/6	1,1/3,1/5	1,1/2,1/3	1,1/2,1/3	1,1,2	1,2,3	0.08
H	1/5,1/7,1/9	1/3,1/5,1/7	1,1/3,1/5	1,1/2,1/3	1,1/2,1/3	1,1,2	0.05

CI = 0.07

Table 6. Fuzzy pair wise comparison for test case of component test case 3

	C	O	I	S	A	H	(E.V.)
C	1,1,2	3,5,7	1,2,3	3,5,7	2,4,6	4,6,8	0.41
O	1/3,1/5,1/7	1,1,2	1,2,3	1,2,3	2,4,6	3,5,7	0.21
I	1,1/2,1/3	1,1/2,1/3	1,1,2	1,3,5	1,2,3	1,3,5	0.17
S	1/3,1/5,1/7	1,1/2,1/3	1,1/3,1/5	1,1,2	1,2,3	1,2,3	0.09
A	1/2,1/4,1/6	1/2,1/4,1/6	1,1/2,1/3	1,1/2,1/3	1,1,2	1,2,3	0.07
H	1/4,1/6,1/8	1/3,1/5,1/7	1,1/3,1/5	1,1/2,1/3	1,1/2,1/3	1,1,2	0.05

CI = 0.08

Table 7. Fuzzy pair wise comparison for Controllability

$$\begin{pmatrix} & \text{CTC1} & \text{CTC2} & \text{CTC3} \\ \text{CTC1} & 1,1,2 & 1,1/3,1/5 & 1,1/3,1/5 \\ \text{CTC2} & 1,3,5 & 1,1,2 & 1,2,3 \\ \text{CTC3} & 1,3,5 & 1,1/2,1/3 & 1,1,2 \end{pmatrix} \begin{pmatrix} \text{E.V.} \\ 0.14 \\ 0.52 \\ 0.33 \end{pmatrix}$$

CI = 0.05

Table 8. Fuzzy pair wise comparison for Observe ability

$$\begin{pmatrix} & \text{CTC1} & \text{CTC2} & \text{CTC3} \\ \text{CTC1} & 1,1,2 & 1,1/3,1/5 & 1,1/2,1/3 \\ \text{CTC2} & 1,3,5 & 1,1,2 & 1,2,3 \\ \text{CTC3} & 1,2,3 & 1,1/2,1/3 & 1,1,2 \end{pmatrix} \begin{pmatrix} \text{E.V.} \\ 0.16 \\ 0.54 \\ 0.30 \end{pmatrix}$$

CI = 0.01

Table 9. Fuzzy pair wise comparison for Isolate ability

$$\begin{pmatrix} & \text{CTC1} & \text{CTC2} & \text{CTC3} \\ \text{CTC1} & 1,1,2 & 1/2,1/4,1/6 & 1/2 \\ \text{CTC2} & 2,4,6 & 1,1,2 & 1,3,5 \\ \text{CTC3} & 1,2,3 & 1,1/3,1/5 & 1,1,2 \end{pmatrix} \begin{pmatrix} \text{E.V.} \\ 0.14 \\ 0.62 \\ 0.24 \end{pmatrix}$$

CI = 0.02

Table 10. Fuzzy pair wise comparison for Separation of Concern

$$\begin{pmatrix} & \text{CTC1} & \text{CTC2} & \text{CTC3} \\ \text{CTC1} & 1,1,2 & 1/2,1/4,1/6 & 1,1/3,1/5 \\ \text{CTC2} & 2,4,6 & 1,1,2 & 1,3,5 \\ \text{CTC3} & 1,3,5 & 1,1/3,1/5 & 1,1,2 \end{pmatrix} \begin{pmatrix} \text{E.V.} \\ 0.12 \\ 0.61 \\ 0.27 \end{pmatrix}$$

CI = 0.08

Table 11. Fuzzy pair wise comparison for automat ability

$$\begin{pmatrix} & \text{CTC1} & \text{CTC2} & \text{CTC3} \\ \text{CTC1} & 1,1,2 & 1/2,1/4,1/6 & 1/2,1/4,1/6 \\ \text{CTC2} & 2,4,6 & 1,1,2 & 1,2,3 \\ \text{CTC3} & 2,4,6 & 1,1/2,1/3 & 1,1,2 \end{pmatrix} \begin{pmatrix} \text{E.V.} \\ 0.11 \\ 0.54 \\ 0.35 \end{pmatrix}$$

CI = 0.06

Table 12. Fuzzy pair wise comparison for heterogeneity

$$\begin{pmatrix} & \text{CTC1} & \text{CTC2} & \text{CTC3} \\ \text{CTC1} & 1,1,2 & 1,1/3,1/5 & 1,1/2,1/3 \\ \text{CTC2} & 1,3,5 & 1,1,2 & 1,2,3 \\ \text{CTC3} & 1,2,3 & 1,1/2,1/3 & 1,1,2 \end{pmatrix} \begin{pmatrix} \text{E.V.} \\ 0.16 \\ 0.54 \\ 0.30 \end{pmatrix}$$

CI = 0.01

Step 8. Calculate un-weighted and weighted super matrix: All the pair wise comparisons are summarized in a matrix in the form of un-weighted super matrix. Here the sum of the column total is greater than 1, so normalize the matrix until the total of this column is less than or equal to 1 and hence forming weighted super matrix.

Table 13. Weighted super matrix

$$\begin{pmatrix} & \text{C} & \text{O} & \text{I} & \text{S} & \text{A} & \text{H} & \text{CTC1} & \text{CTC2} & \text{CTC3} \\ \text{C} & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.367 & 0.388 & 0.409 \\ \text{O} & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.224 & 0.217 & 0.212 \\ \text{I} & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.184 & 0.172 & 0.167 \\ \text{S} & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.084 & 0.102 & 0.091 \\ \text{A} & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.093 & 0.076 & 0.074 \\ \text{H} & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.046 & 0.045 & 0.047 \\ \text{CTC1} & 0.142 & 0.164 & 0.137 & 0.120 & 0.110 & 0.164 & 0.000 & 0.000 & 0.000 \\ \text{CTC2} & 0.525 & 0.539 & 0.623 & 0.608 & 0.544 & 0.539 & 0.000 & 0.000 & 0.000 \\ \text{CTC3} & 0.334 & 0.297 & 0.239 & 0.272 & 0.346 & 0.297 & 0.000 & 0.000 & 0.000 \end{pmatrix}$$

Step 9. Limit matrix: Limit matrix is obtained by raising the power of weighted super matrix. Raise the power until the row elements of the weighted super matrix are

become stable and same. In limit matrix the decision maker can make decision about the components.

Table 14. Limit matrix

	C	O	I	S	A	H	CTC1	CTC2	CTC3
C	0.39	0.39	0.39	0.39	0.39	0.39	0.00	0.00	0.00
O	0.22	0.22	0.22	0.22	0.22	0.22	0.00	0.00	0.00
I	0.17	0.17	0.17	0.17	0.17	0.17	0.00	0.00	0.00
S	0.10	0.10	0.10	0.10	0.10	0.10	0.00	0.00	0.00
A	0.08	0.08	0.08	0.08	0.08	0.08	0.00	0.00	0.00
H	0.05	0.05	0.05	0.05	0.05	0.05	0.00	0.00	0.00
CTC1	0.00	0.00	0.00	0.00	0.00	0.00	0.14	0.14	0.14
CTC2	0.00	0.00	0.00	0.00	0.00	0.00	0.55	0.55	0.55
CTC3	0.00	0.00	0.00	0.00	0.00	0.00	0.30	0.30	0.30

Step 10. Prioritize component test case: After completing all the necessary steps, prioritize the components test case according to the calculations done. The most prioritize components test case among available are adopted for the system to be design. Such prioritization of components test case leads the developer to a successful system.

to meet performance goal. Test case prioritization also enhances cost-effectiveness of the test cases [33]. The proposed techniques amalgamate fuzzy and ANP toward component test case prioritization under the condition of uncertain environment. FANP has the benefits of multi criteria decision in the vague situation where there are certain alternatives available and we have to select the most suitable and appropriate one.

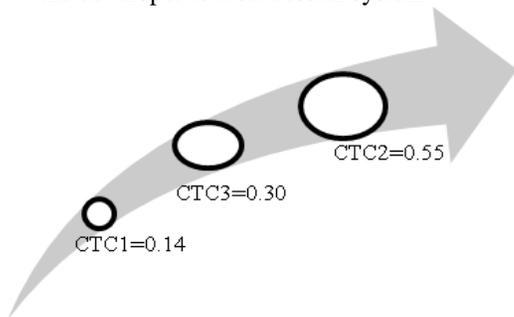


Figure 5. Graphical representation of test cases score

4. Discussion

The resultant output in step 10 (reference to figure 5) clearly shows that the component test case (CTC 2) is the top most test case followed by the component test case 3 and then followed by component test case 1. This component test case prioritization is obtained after the calculation of proposed FANP which is explained in the 10- step FANP process.

5. Conclusion

Software component testability is one of the important concerns of software testing which is the integration of some pre designed and tested software components. Structuring software from pre-tested and testable components ultimately leads to a highly quality software system. Component test case prioritization is basically done for the problems of fault detection in a test case, reliability, faster test rate and satisfaction of user. Component test case prioritization plan is aimed at optimizing the order in which the test cases are to be executed which, in turn, amplifies their effectiveness

References

- [1] Gao J Z, Gupta K K, Gupta S, and Shim S S Y, On building testable software components, in *Proceedings of the First International Conference on COTS-Based Software Systems*, London, UK. Springer-Verlag, 2002, pp. 108-121.
- [2] Weyuker E J. Testing Component-Based Software: A Cautionary Tale. *IEEE Software*, 1998 vol. 15, pp. 54-59.
- [3] Elbaum S, Malishevsky A G, and Rothermel G. Test case prioritization: A family of empirical studies. *IEEE Transactions on Software Engineering*, 2002 vol. 28, pp. 159-182.
- [4] Korel B, Tahat L H, and Harman M, Test prioritization using system models, in *Proceedings of the 21st IEEE International Conference on Software Maintenance, 2005. ICSM'05*, 2005, pp. 559-568.
- [5] Nazir S, Shahzad S, Nazir M, and Rehman H U, Evaluating security of software components using analytic network process, in *11th International Conference on Frontiers of Information Technology (FIT)* Islamabad, Pakistan: IEEE 2013.
- [6] Mikhailov L and Singh M G. Fuzzy analytic network process and its application to the development of decision support systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 2003 vol. 33, pp. 33-41.
- [7] Cheong C W and Way C C, Fuzzy linguistic decision analysis for Web server system future planning, in *TENCON 2000. Proceedings*, 2000, pp. 367-372 vol.1.
- [8] Gao J, Gopinathan D, Mai Q, and He J, A Systematic Regression Testing Method and Tool For Software Components, in *Proceedings of the 30th Annual International Computer Software and Applications Conference (COMPSAC'06)*, 2006, pp. 1-9.

- [9] Giannakopoulou D, Pasareanu C S, and Blundell C. Assume-guarantee testing for software components. *IET Software* 2008 vol. 2, pp. 547-562.
- [10] Lemos A L, Ferrari F C, Eler M M, Maldonado J C, and Masiero P C. Evaluation studies of software testing research in Brazil and in the world: A survey of two premier software engineering conferences. *The Journal of Systems and Software*, 2012 vol. 86, pp. 951–969.
- [11] Mao C and Lu Y, Regression Testing for Component-based Software Systems by Enhancing Change Information, in *Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSEC'05)*: IEEE, 2005.
- [12] Hla K H S, Choi Y, and Park J S, Applying Particle Swarm Optimization to Prioritizing Test Cases for Embedded Real Time Software Retesting, in *IEEE 8th International Conference on Computer and Information Technology Workshops, 2008. CIT Workshops 2008*, 2008, pp. 527-532.
- [13] Korel B, Koutsogiannakis G, and Tahat L H, Application of system models in regression test suite prioritization, in *IEEE International Conference on Software Maintenance*, 2008, pp. 247-256.
- [14] Zheng W and Bundell G, Test by Contract for UML-Based Software Component Testing, in *International Symposium on Computer Science and its Applications*, 2008, pp. 377-382.
- [15] Karambir and Rani R. Prioritize Test Case Survey for Component-Based Software Testing. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2013 vol. 3, pp. 815-820.
- [16] Mohanty S, Acharya A A, and Mohapatra D P, A model based prioritization technique for component based software retesting using uml state chart diagram, in *3rd International Conference on Electronics Computer Technology (ICECT), 2011*, pp. 364-368.
- [17] Elbaum S, Malishevsky A G, and Rothermel G, Prioritizing Test Cases for Regression Testing, in *International Symposium of Software Testing and Analysis* August 2000, pp. 102-112.
- [18] Zheng W and Bundell G, Model-Based Software Component Testing: A UML-Based Approach, in *6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007)*: IEEE, 2007, pp. 1-8.
- [19] Michael C C, Reusing Tests of Reusable Software Components, Technical report RSTR-96-003-06, RST Corporation 1996.
- [20] Nazir S, Anwar S, Khan M A, Khan H, and Nazir M, A Novel Fuzzy Logic Based Software Component Selection Modeling, in *International Conference on Information Science and Application (ICISA)*, Korea, 2012, pp. 1-6.
- [21] Jacob T P and Ravi D T. Optimal Regression Test Case Prioritization using genetic algorithm. *Life Science Journal*, vol. 10, 2013.
- [22] Alghabban W G and Qureshi M R J. The Proposal of Improved Component Selection Framework. *Life Science Journal*, vol. 10, 2013, pp. 3538-3544.
- [23] Rothermel G, Untch R H, Chengyun C, and Harrold M J. Prioritizing test cases for regression testing. *IEEE Transactions on Software Engineering*, 2001 vol. 27, pp. 929-948.
- [24] P´erez A M and Kaiser S, Reusing Component Test Cases for Integration Testing of Retarding Embedded System Components, in *First International Conference on Advances in System Testing and Validation Lifecycle, 2009. VALID '09*, 2009, pp. 1-6.
- [25] Yoon I, Sussman A, Memon A, and Porter A. Testing component compatibility in evolving configurations. *Information and Software Technology*, 2013 pp. 445-458.
- [26] Asan U, Soyer A, and Serdarasan S, "A Fuzzy Analytic Network Process Approach," in *Computational Intelligence Systems in Industrial Engineering*. vol. 6: Atlantis Press, pp. 155-179.
- [27] Zadeh L. Fuzzy Logic. *Computer*, 1988 vol. 1, pp. 83-93.
- [28] Saaty T L. Fundamentals of the Analytic Network Process – Dependence and Feedback In Decision-Making with A Single Network. *Journal of systems science and systems engineering*, 2004 vol. 13, pp. 129-157.
- [29] Yüksel İ and Dagdeviren M. Using the analytic network process (ANP) in a SWOT analysis – A case study for a textile firm. *Information Sciences*, 2007 vol. 177, pp. 3364-3382.
- [30] Riley T, Goucher A, and Adam G, *Beautiful Testing: Leading Professionals Reveal How They Improve Software*. United States of America: O'Reilly Media, Inc., 2009.
- [31] Gurung P M B, "Adaptation of Software Testability Concept for Test Suite Generation- A Systematic Review." vol. Master Thesis Software Engineering Karlskrona Sweden: School of Computing Blekinge Institute of Technology SE-371 79 March 2012.
- [32] Saaty T. L. Relative measurement and its generalization in decision making why pairwise comparisons are central in mathematics for the measurement of intangible factors the analytic hierarchy/network process. *RACSAM - Revista de la Real Academia de Ciencias Exactas, Fisicas y Naturales. Serie A. Matematicas*, 2008 vol. 102, pp. 251-318.
- [33] Srivastava A and Thiagarajan J, "Effectively prioritizing tests in development environment," in *Proceedings of the 2002 ACM SIGSOFT international symposium on Software testing and analysis (ISSTA '02)*, ACM, New York, NY, USA 2002, pp. 97-106.