This article has been accepted for publication in a future issue of a journal/proceedings, but has not been fully edited. Content may change prior to final publication. Citation information: Belli, F., Eminov, M., Goekce, N. , Model-Based Test Prioritizing - A Comparative Soft Computing Approach and Case Studies, LNAI 5803, Springer-Verlag, pp. 427-434

Model-Based Test Prioritizing – A Comparative Soft-Computing Approach and Case Studies

Fevzi Belli[1], Mubariz Eminov[2], and Nida Gokce[3]

[1] Department of Computer Science, Electrical Engineering and Mathematics, University of Paderborn, Germany
belli@upb.de
[2] Faculty of Engineering, Department of Computer Engineering, Halic University, Istanbul, Turkey
mubarizeminli@halic.edu.tr
[3] Faculty of Arts and Sciences, Department of Statistics, Mugla University, Turkey
nidagokce@yahoo.com

Abstract. Man-machine systems have many features that are to be considered simultaneously. Their validation often leads to a large number of tests; due to time and cost constraints they cannot exhaustively be run. It is then essential to prioritize the test subsets in accordance with their importance for relevant features. This paper applies soft-computing techniques to the prioritizing problem and proposes a graph model-based approach where preference degrees are indirectly determined. Events, which imply the relevant system behavior, are classified, and test cases are clustered using (i) unsupervised neural network clustering, and (ii) Fuzzy c-Means clustering algorithm. Two industrial case studies validate the approach and compare the applied techniques.

Keywords: model-based test prioritizing, adaptive competitive learning, fuzzy c-means, clustering, neural networks.

1 Introduction and Related Work

Testing is one of the important, traditional, analytical techniques of quality assurance widely accepted in the software industry. There is no justification, however, for any assessment of the correctness of software under test (SUT) based on the success (or failure) of a single test, because potentially there can be an infinite number of test cases. To overcome this shortcoming of testing, formal methods model and visualize the relevant, desirable features of the SUT. The modeled features are either functional or structural issues, leading to *specification-* or *implementation-oriented* testing, respectively. Once the model is established, it "guides" the test process to generate and select test cases, which form sets of test cases (*test suites*). The test selection is ruled by an *adequacy criterion* for measuring the effectiveness of test suites [8]. Most of the existing adequacy criteria are *coverage*-oriented [18].

The test approach introduced in this paper is specification- and coverage-oriented. *Event sequence graphs* (*ESG*, [2,3]) are favored for modeling which view the

behavior of SUT and its interaction with user as events. Because of the large amount of features to be considered, the number of test cases and thus the costs of testing often tend to run out of the test budget. Therefore, it is important to test the most important items first which leads to the *test prioritization problem* [12]:

Given: A test suite *T*; the set of permutations of *T* (symbolized as *PT*); a function *f* from *PT* to the real numbers representing the preference of the tester while testing.

Problem: Find $T' \in PT$ such that $(\forall T'')\ (T'' \neq T')\ [f(T') \geq f(T'')]$

The ESG approach generates test suites through a finite sequence of discrete events. The underlying optimization problem is a generalization of the *Chinese Postman Problem* (*CPP*) [13] and algorithms given in [2,3] differ from the well-known ones in that they satisfy not only the constraint that a minimum total length of test sequences is required, but also fulfill the coverage criterion with respect to converging of all event pairs represented graphically. To overcome the problem that an exhaustive testing might be infeasible, the paper develops a *prioritized* version of the mentioned test generation and optimization algorithms, in the sense of "divide and conquer" principle. This is the primary objective and the kernel of this paper which is novel and thus, to our knowledge, has not yet been worked out in previous works, including ours [2,3].

The approaches assign to each test generated a *degree of its preference* which is estimated for all including events and qualified by several attributes that depend on the features of the project, and their values are justified by their significance to the user. For clustering, unsupervised neural networks (NN) [14] and Fuzzy c-Means (FCM) analysis [9,10] are used.

Section 2 explains the background of the approach, summarizing ESG notation and NN and FCM clustering algorithms. Section 3 describes the proposed prioritized graph-based testing approach. Two case studies empirically validate the approach in Section 4 which also identifies and analyzes its characteristic issues. Section 5 summarizes the results, gives hints to further researches and concludes the paper.

2 Background and Approach

2.1 Event Sequence Graphs for Test Generation

Basically, an *event* is an externally observable phenomenon, such as an environmental or a user stimulus, or a system response, punctuating different stages of the system activity. A simple example of an ESG is given in Fig.1.

Mathematically speaking, an ESG is a directed, labeled graph and may be thought of as an ordered pair $ESG=(\alpha, E)$, where α is a finite set of nodes (vertices) uniquely labeled by some input symbols of the alphabet Σ, denoting events, and E: $\alpha \rightarrow \alpha$, a

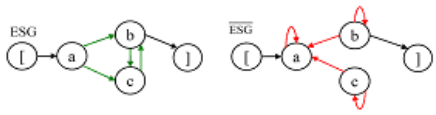

Fig. 1. An event sequence graph ESG and $\overline{ESG}$ as the complement of the given ESG

precedence relation, possibly empty, on α. The elements of E represent directed arcs (edges) between the nodes in α. Given two nodes a and b in α, a directed arc ab from a to b signifies that event b can follow event a, defining an *event pair* (*EP*) ab (Fig. 1). The remaining pairs given by the alphabet Σ, but not in the ESG, form the set of *faulty event pairs* (*FEP*), e.g., ba for ESG of Fig.1. As a convention, a dedicated, start vertex e.g., *[*, is the *entry* of the ESG whereas a final vertex e.g., *]* represents the *exit*. Note that *[* and *]* are not included in Σ. The set of FEPs constitutes the *complement* of the given ESG ($\overline{ESG}$ in Fig.1).

A sequence of $n+1$ consecutive events that represents the sequence of n arcs is called an *event sequence (ES) of the length n+1*, e.g., an *EP* (*event pair*) is an ES of length 2. An ES is *complete* if it starts at the initial state of the ESG and ends at the final event; in this case it is called a *complete ES* (*CES*). Occasionally, we call CES also as *walks* (or *paths*) through the ESG given [2,3].

Completeness Ratio (CR) is a metric which explains density of edges in the ESG and is defined as follows:

$$CR = |E|/|V|^2 \tag{1}$$

where $|E|$ is the number of edges in the ESG and $|V| = n$ is the number of nodes (vertex) in the ESG. CR takes the values between 0 and 1.Value 1 shows that ESG is completed graph and Value 0 means null graph. As the values are getting closer to 1, the density of the graph gets bigger. ESG concept and definitions informally introduced above are sufficient to understand the test prioritization approach represented in this paper. For more information on ESG refer to ([2,3,7,16]).

2.2 Neural Network-Based Clustering

In this study, we have chosen unsupervised NN where competitive learning CL algorithm can adaptively cluster instances into clusters. For clustering of an unstructured data set dealing especially with vector quantization, unsupervised learning based on clustering in a NN framework is fairly used. In clustering a data set $X = \{x_1, \dots, x_i, \dots x_n,\} \subset \mathbb{R}^p$, $\in \mathbb{R}^p$ *containing events* (nodes) is portioned into c *number of clusters* each of which contains a subset S_k defined as:

$$X = \cup_{k=1}^{c} S_k \text{ with } S_k \cap S_j = 0 \quad \forall k \neq j \tag{2}$$

Each S_k cluster is represented by a cluster center (*prototype*) that corresponds to a weight vector $\widetilde{w}_k = \left(\widetilde{w}_{k1}, \dots, \widetilde{w}_{kj}, \dots, \widetilde{w}_{kp}\right) \in \mathbb{R}^p$ and after finding a trained value of all weight vectors $\widetilde{W} = \{\widetilde{w}_1, \dots, \widetilde{w}_k, \dots, \widetilde{w}_c\} \subset \mathbb{R}^p$ the data set $X \epsilon \mathbb{R}^p$ is divided into k^{th} cluster by the condition:

$$S_k = \left\{ x \in \mathbb{R}^p \middle| \sum_{j=1}^{p} \tilde{x}_{ij} \tilde{w}_{kj} \geq \sum_{j=1}^{p} \tilde{x}_{ij} \tilde{w}_{gj} \ \forall k \neq g \right\} \quad i = 1, \dots, n \quad j = 1, \dots, p \quad k = 1, \dots, c \quad g = 1, \dots, c \in \mathbb{N} \tag{3}$$

Training: The initial values of the weight vectors are randomly allotted. It negatively influences the clustering performance of standard CL algorithm that is explained in literature [5,11,14]. In order to get a better clustering performance, in this paper we use the *adaptive* CL algorithm by deleting over specified weight vectors [15,16]. The main distinguishing properties of this algorithm from standard CL are: Both data points $\tilde{x}_i$ and weight vectors $\widetilde{w}_k$ are normalized to a unit length, i.e., they are presented as the unit vector the length of which is 1 (one). In adaptive CL algorithm, the winner weight vector $\widetilde{w}_w$ is determined by a dot product of data point $\tilde{x}_i$ and weight vector $\widetilde{w}_k$ as in (4), and the updating rule of a winner weight vector [5] is based on the adjusting equation expressed as in (5)

$$\widetilde{w}_w = arg\ max_k\{\sum_{j=1}^{p} \tilde{x}_{ij}\, \widetilde{w}_{kj}\} \quad i = 1, \ldots, n \quad k = 1, \ldots, c \tag{4}$$

$$\Delta\widetilde{w}_w(t) = \eta(t)(\tilde{x}_i/p - \widetilde{w}_w) \tag{5}$$

where η is a learning rate. There is a deletion mechanism [7,16] that eliminates one weight vector, w_s that corresponds to cluster which has a minimum intra-cluster partition error, i.e., $D_k \geq D_s$, for all k, and it proceeds until the number of weight vectors is equal to the predetermined one. D_k is determined as in (6):

$$D_k = \frac{1}{p}(\sum_{\tilde{x} \in S_k} \tilde{x}\, \widetilde{w}_w) \qquad k = 1, \ldots, c \tag{6}$$

2.3 FCM-Based Clustering

Fuzzy c-means (*FCM) algorithms* as a fuzzy version of hard c-means (HCM), crisp partition, as introduced in [9] and improved by ”fuzzifier m” in [1]. In real applications there are very often no sharp boundaries among clusters so that fuzzy clustering is often better suited for the data. FCM (probabilistic) clustering assigns the data into c fuzzy clusters each of which is represented by its center, called a *prototype*, as a representative of data. There every datum may belong simultaneously to several clusters with corresponding *membership degree (MD) that* has value between zero and one.In FCM, for optimal partitioning the alternating optimization algorithm for minimizing the objective function [6, 7, 10] presented in (7) is used:

$$J(X, U, V) = \sum_{k=1}^{c} \sum_{i=1}^{n} (u_{ki})^m d^2(v_k, x_i) \tag{7}$$

2.4 Prioritized ESG-Based Testing

We consider the testing process is based on the generation of a test suite from ESG that is a discrete model of a SUT. To generate tests, a set of ESGs is derived which are input to the generation algorithm to be applied. Our prioritized testing approach is based on the ESG-based testing algorithms as introduced in [2,3]. The constraints on total lengths of the tests [2,3] generated are enable a considerable reduction in the cost

of the test execution. To solve the test prioritizing problem, several algorithms have been introduced [4,8]. However, this kind of prioritized testing is computationally expensive and hence restricted to deal with short test cases only.

The ordering of the CESs is in accordance with their importance degree which is defined indirectly, i.e., by classification of events that are the nodes of ESG and represent objects (modules, components) of SUT. For this aim, firstly events are presented as a *multidimensional data vector* $x_i = (x_{i1}, \dots, x_{ip})$ then; a data set $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^p$ is constructed which divided into c groups. The groups are constructed by using both Adaptive CL algorithm and FCM clustering algorithm and then classification procedure as explained in detail in our previous works [7,16]. Afterwards, the importance degree of these groups has been assigned according to length of their corresponding weight vector.

Importance index $(Imp(x_i))$ of ith event belonging to k^{th} group is defined as follows:

$$Imp(x_i) = c - ImpD(S_k) + 1 \tag{8}$$

where c is the optimal number of the groups; $ImpD(S_k)$ is the importance degree of the group S_k where the ith event belongs to and this importance degree is determined by comparing the length of obtained groups weight vectors. Finally, the assignment of preference degrees to CESs is based on the rule that is given in [7,16].

Therefore, the preference degree of CES can be defined by taking into account both the importance of events and the frequency of occurrence of event(s) within them, and it can be formulated as follows:

$$PrefD(CES_q) = \sum_{i=1}^{n} Imp(x_i)\mu_{S_k}(x_i) f_q(x_i) \qquad q = 1, \dots r \in \mathbb{N} \tag{9}$$

where r is the *number of CESs*, $Imp(x_i)$ is importance index of the ith event (8), $\mu_{S_k}(x_i)$ is MD of the ith event belonging to the group S_k (it is 0 or 1 in NN-based clustering, and it takes any value between 0 and 1 in FCM-based clustering), and $f_q(x_i)$ is frequency of occurrence of event i^{th} within CES_q. Finally, the CESs are ordered, scaling of their preference degrees (9) based on the events which incorporate the importance group(s). We propose *indirect determination of preference algorithm* for prioritized ESG-based testing algorithm in our previous works [7,16].

3 Case Studies

Data borrowed from two industrial projects are used in the following case studies which are mentioned in our previous papers [7,16]: A marginal strip mower (Fig.2.(a)), a web-based tourist portal (Fig.2.(b)) and an example ESG (Fig.2.(c)). The construction of ESG and the generation of test cases from those ESGs have been explained in the previous papers of the first author [2,3]. Classification of the events using the results of Adaptive CL algorithm is accomplished according to equality (3)

where each event is assigned crisply to one group only. After FCM clustering, each event belongs to only one group according to its maximum MD. For two clustering approach, importance degrees ($ImpD(S_k)$) of groups are determined by comparing the length of corresponding center vectors (ℓ), and their values that are given in Table 1.

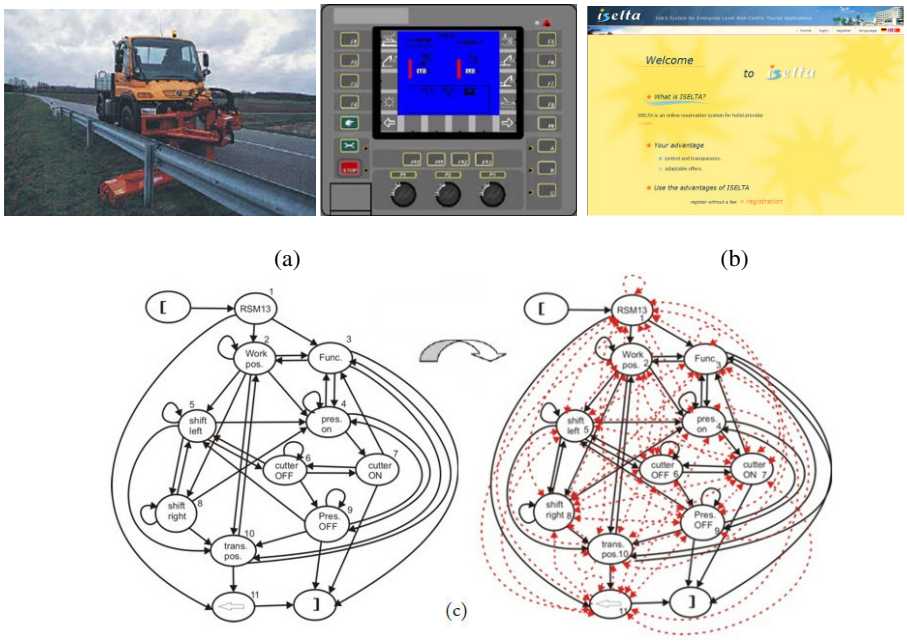

Fig. 2. (a) The marginal strip mower mounted on a vehicle and its display unit as a control desk. (b) The web-based system ISELTA. (c) Example ESG.

Table 1. Ranking of CESs of the example ESG using algorithms selected

Adaptive CL					*FCM*				
Groups	*Events*	*MD*	ℓ	*ImpD(S_k)*	*Groups*	*Events*	*MD*	ℓ	*ImpD(S_k)*
$S_1^{(A)}$	3 4 10	1 1 1	2,094	1	$S_4^{(F)}$	5 4 9	0,934 0,445 0,272	1,8208	4
$S_2^{(A)}$	1	1	1,344	6	$S_6^{(F)}$	11	0,999	1,9780	3
$S_3^{(A)}$	2 5	1 1	1,589	3	$S_5^{(F)}$	8 10	0,890 0,516	1,7341	5
$S_4^{(A)}$	9	1	1,788	2	$S_1^{(F)}$	3	0,996	2,2471	1
$S_5^{(A)}$	7 8 11	1 1 1	1,544	5	$S_2^{(F)}$	7 6 1	0,811 0,645 0,362	1,4073	6
$S_6^{(A)}$	6	1	1,550	4	$S_3^{(F)}$	2	0,985	2,0235	2

Table 2. Ordering of CESs of the example ESG using methods selected

CESs		***Adaptive CL***		***FCM***	
No	*Walks*	*PrefD(CES$_q$)*	*Order No*	*PrefD(CES$_q$)*	*Order No*
CES$_1$	[1 3 4 7 3 10 2 2 4 4 9 4 3]	64	**2**	36,147	**1**
CES$_2$	[1 2 2 5 4 7 6 6 7]	29	**5**	17,262	**6**
CES$_3$	[1 2 5 6 5 5 8 8 5 10 11]	36	**3**	25,734	**3**
CES$_4$	[1 2 3 2 8 10 3]	29	**4**	24,979	**4**
CES$_5$	[1 11]	3	**7**	4,359	**7**
CES$_6$	[1 2 10 2 5 6 9 9 8 4 9 10 2 4 9]	66	**1**	28,360	**2**
CES$_7$	[1 2 4 9 5 10 11]	28	**6**	15,270	**5**

The preference degrees of the CESs are determined by (9), and the ordering of the CESs is represented in Table 2. Consequently, we have a ranking of test cases to make the decision of which test cases are to be primarily tested. For all four considered ESGs examples, to compare the performance of clustering approaches the mean square error (MSE) (10) in accordance with corresponding level of CR are determined and compared in Table 3.

$$MSE = \frac{1}{np}\sum_{k=1}^{c}\sum_{x\in S_k} d(x,\bar{x}_k) \tag{10}$$

Table 3. Completeness Ratio (CR) and clustering error for each examples

EXAMPLES		**ESG1**	**ESG2**	**ESG3**	**ESG4**
CR		0,31	0,17	0,10	0,83
MSE (10)	**FCM**	0,022	0,037	0,032	0,014
	Adaptive CL	0,030	0,036	0,037	0,027

4 Conclusions

The model-based, coverage-and specification-oriented approach described in this paper provides a novel and effective algorithms for ordering test cases according to their degree of preference. Such degrees are determined indirectly through the use of the events specified by several attributes, and no prior knowledge about the tests carried out before is needed. Those are important and consequently, the approach introduced radically differs from the existing ones. This approach is useful when an ordering of the tests due to restricted budget and time is required.

In this paper, the events (nodes of ESG) are classified by using both unsupervised NN based adaptive CL and FCM clustering algorithm, and their classification results are investigated comparatively. After carrying out corresponding clustering, in order to classify the events, the nearest centroid (weight) rule and the maximum MD based deffuzzification procedure is applied, respectively. So the crisply and fuzzily assigning groups of events are produced. FCM clustering based prioritization of CESs is more plausible and convenient than the former due to construction of the fuzzy

qualified groups of events that are important when groups to be obtained are interloped. However, computational complexity of FCM clustering is higher than the other and hence choice of the appropriate clustering algorithm in dependence of data structure is needed. For this aim, usage of CR metric defined in (1) and evaluating its value is proposed. As seen from Table 5, for little CR value (less than approximately 0.18), the adaptive CL algorithm should be preferred as the simple method (no difference between MSE). But for greater CR value (more than nearly 0.31), FCM clustering is more suitable due to the little value of MSE.

References

1. Dunn, J.C.: A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. Journal of Cybernetics 3, 32–57 (1973)
2. Belli, F., Budnik, C.J., White, L.: Event-Based Modeling, Analysis and Testing of User Interactions – Approach and Case Study. J. Software Testing, Verification & Reliability 16(1), 3–32 (2006)
3. Belli, F., Budnik, C.J.: Test Minimization for Human-Computer Interaction. J. Applied Intelligence 7(2) (2007) (to appear)
4. Bryce, R.C., Colbourn, C.C.: Prioritized Interaction Testing for Pair-wise Coverage with Seeding and Constraints. Information and Software Technology 48, 960–970 (2006)
5. Rummelhart, D.E., Zipser, D.: Competitive Learning. J. Cognitive Science 9, 75–112 (1985)
6. Eminov, M.E.: Fuzzy c-Means Based Adaptive Neural Network Clustering. In: Proc. TAINN 2003, Int. J. Computational Intelligence, pp. 338–343 (2003)
7. Belli, F., Eminov, M., Gökçe, N.: Coverage-Oriented, Prioritized Testing – A Fuzzy Clustering Approach and Case Study. In: Bondavalli, A., Brasileiro, F., Rajsbaum, S. (eds.) LADC 2007. LNCS, vol. 4746, pp. 95–110. Springer, Heidelberg (2007)
8. Binder, R.V.: Testing Object-Oriented Systems. Addison-Wesley, Reading (2000)
9. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
10. Hoppner, F., Klawonn, F., Kruse, R., Runkler, T.: Fuzzy Cluster Analysis. John Wiley, Chichester (1999)
11. Fu, L.: Neural Networks in computer Intelligence. McGraw Hill, Inc., New York (1994)
12. Elbaum, S., Malishevsky, A., Rothermel, G.: Test Case Prioritization: A Family of Empirical Studies. IEEE Transactions on Software Engineering 28(2), 182–191 (2002)
13. Edmonds, J., Johnson, E.L.: Matching: Euler Tours and the Chinese Postman. Math. Programming, 88–124 (1973)
14. Eminov, M., Gokce, N.: Neural Network Clustering Using Competitive Learning Algorithm. In: Proc. TAINN 2005, pp. 161–168 (2005)
15. Kim, D.J., Park, Y.W., Park, D.J.: A Novel Validity Index for Clusters. IEICE Trans. Inf & System, 282–285 (2001)
16. Belli, F., Eminov, M., Gökçe, N.: Prioritizing Coverage-Oriented Testing Process- An Adaptive-Learning-Based Approach and Case Study. In: The Fourth IEEE International Workshop on Software Cybernetics, IWSC 2007, Beijing, China, July 24 (2007)
17. Memon, A.M., Pollack, M.E., Soffa, M.L.: Hierarchical GUI Test Case Generation Using Automated Planning. IEEE Trans. Softw. Eng. 27/2, 144–155 (2001)
18. Zhu, H., Hall, P.A.V., May, J.H.R.: Software Unit Test Coverage And Adequacy. ACM Computing Surveys 29(4) (December 1997)