



Feature selection and evaluation for software usability model using modified moth-flame optimization

Deepak Gupta¹ · Anil K. Ahlawat² · Arun Sharma³ ·
Joel J. P. C. Rodrigues^{4,5} 

Received: 29 April 2018 / Accepted: 20 March 2020 / Published online: 5 May 2020
© Springer-Verlag GmbH Austria, part of Springer Nature 2020

Abstract

This paper introduces a nature-inspired optimized algorithm called modified moth-flame optimization (MMFO) for usability feature selection. To determine quality of software usability plays a significant role. This model contains various usability factors that are divided into several features, which have some characteristics, thus making a hierarchical model. Here, the authors have introduced MMFO (Modified Moth-flame optimization algorithm) for the selection of usability features to get an optimal solution MMFO is an extension of moth-flame optimization algorithm (MFO), which is based on the navigation method of moths called transverse orientation and to the best of our knowledge; this algorithm is introduced in software engineering practices. The selected features and accuracy of proposed MMFO is compared with the original MFO and other related optimization techniques. The results shows that the proposed nature-inspired optimization algorithm outperforms the other related optimizers as it generates a fewer number of selected features and having low accuracy.

✉ Joel J. P. C. Rodrigues
joeljr@ieee.org

Deepak Gupta
deepakgupta@mait.ac.in

Anil K. Ahlawat
a_anil2000@yahoo.com

Arun Sharma
arunsharma2303@gmail.com

- 1 Maharaja Agrasen Institute of Technology, Delhi, India
- 2 Krishna Institute of Engineering and Technology (KIET) Ghaziabad, Ghaziabad, India
- 3 Indira Gandhi Delhi Technical University for Women (IGDTUW), Delhi, India
- 4 Federal University of Piau  (UFPI), Teresina, PI, Brazil
- 5 Instituto de Telecomun c es, Covilh , Portugal

Keywords Moth-flame optimization · Software usability model · Optimization · Feature selection

Mathematics Subject Classification 91C05

1 Introduction

Usability, an essential software quality factor, used during developments of software. There are different definitions of the term Usability by different experts. The term ‘usability’ means the simplicity of learnability, remembrance and use of the software’s by the end-users. The technical definition of usability is the degree to which a software product is employed by particular groups to achieve the goal of efficiency, effectiveness, satisfaction and many other features. ISO/IEC 9126 standard defined usability as the ease with which users understand the software under specific conditions [1]. The ISO 9241-11 standard describes usability as effectiveness and efficiency in specified circumstances [2]. The IEEE Std.610.12-1990 standard describes usability as learnability, input and output efficiency of the system [3]. Usability as a model of quality that is made of five characteristics from the interaction outcomes when a product is used described by ISO/IEC 25010 (2011). ISO 9241-11 (2015) defines usability as the results from the individual and organizational interactions.

The usability is a vital quality model that should be used to produce a quality product. The main purpose of this paper is to define usability in terms of hierarchical model so different factors, attributes and characteristics are combined in hierarchical manner. To achieve our motive the evolutionary algorithm Modified Moth-flame optimization Algorithm (MMFO) is proposed. Evolutionary algorithms imitate mutation, reproduction and selection to provide optimized solution to a complex problem. Some of well-known evolutionary algorithm that has been used to provide optimal solutions are: Genetic algorithms (GA) imitates natural selection, Cuckoo search (CS) is driven by cuckoo’s deceiving other birds into raising their young by depositing their eggs into host nests [4]. Bat algorithm mimics echolocation behaviors of microbats [5].

This work uses the following approaches:

A. Feature selection: It is a method to select the number of features from a noisy dataset (original set of features) and retain only the important subset of features without downgrading the performances of the system [6]. Some of Evolutionary algorithms that have presented work on feature selection are Binary Chaotic Crow Search Algorithm [7], Binary Bat Algorithm [8], Binary Grey Wolf Optimization [9].

B. Binary Moth-flame optimization feature selection [10]: MFO mimics the behaviour of moths in a space. Moths move spirally to change their positions in the space, This spiral movement (SM) is the integral part of MFO algorithm as it evaluates the new positions of moths around flames, allowing moths to fly around the flame instead of the space between them. MFO uses wrapper-based method for feature selection by using k-nearest neighbor (KNN).

Many standards that describe usability are given in [1–3, 11–25], but they have different definitions creating more confusion among experts to understand usability.

In this paper, our main motive is to define ‘usability’ in a hierarchical manner. The model contains the usability factors that are divided into features and each feature has some characteristics thus eliminating confusions and inconsistency among the experts. We have also presented a bio-inspired optimization algorithm for selection of usability feature called MMFO. MMFO is a modification of MFO for feature selection. To the best of our knowledge, we would also like to state that MFO is never applied for selection of usability features. A dataset is also prepared for the same by doing a survey. The main contributions of the proposed work are provided below:

- This work uses the term ‘usability’ in the context of software development using hierarchical model, which has been defined by the authors in their existing work [26].
- A modified nature-inspired optimized algorithm is being presented for selection of usability features.
- The introduced MMFO (Modified Moth-flame optimization Algorithm) for selection of usability features generates an optimal solution of useful usability features out of the original set of usability features.
- The proposed algorithm has also been implemented to predict the total number of selected features, accuracy and convergence.
- The results obtained in this work has been compared with the original Moth-Flame Optimization (MFO), Binary BAT (BBAT), Modified Binary BAT (MBBAT) Algorithms.
- The proposed algorithm outperforms the original MFO, MBBAT, and BBAT Algorithms as the resultant have fewer numbers of selected features and low accuracy.

A live auction portal and complete software development life cycle model can be seen in research articles where fuzzy logic expert system is used [27–30]. In few articles, live auction and SDLC datasets are discussed [31, 32]. Similarly, algorithms inspired by biological models [33] like crow search algorithm [34], binary bat algorithm [7], and modified whale optimization algorithms [35] are also utilized in many research articles to solve the problems associated with Decision Support System (DSS).

This paper is structured as FOLLOWS. Section 2 proposed the hierarchical based usability model and Sect. 3 describes the Moth-flame optimization algorithm (MFO) and other techniques used that is followed by Sect. 4 which proposed MMFO for usability feature selection. The implementation of proposed MMFO is given in Sect. 5. Results and comparison between original MFO and proposed MMFO has been presented in Sect. 6. Section 7 concludes the paper and suggests future research works.

2 The hierarchical based usability model

Many standards that describe usability are given in [1–3, 11–25], but they have different definitions creating more confusion among experts to understand usability. Same characteristics are explained in different manner in different model leading to inconsistency. The motive of this paper is to eliminate this inconsistency and propose a new generic hierarchical model that contains usability factors, usability features and

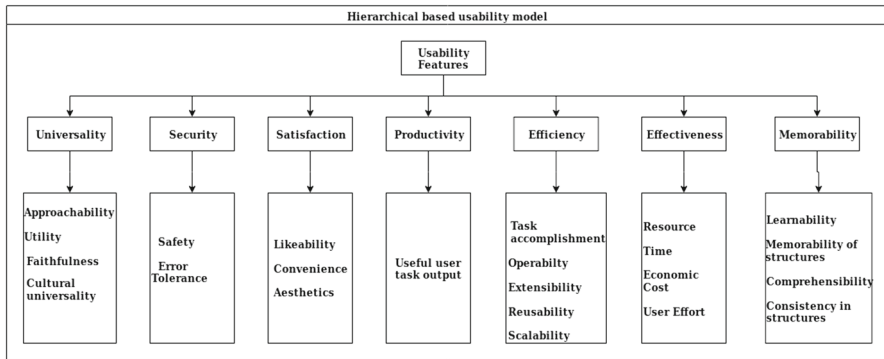


Fig. 1 Detailed taxonomy in hierarchical manner

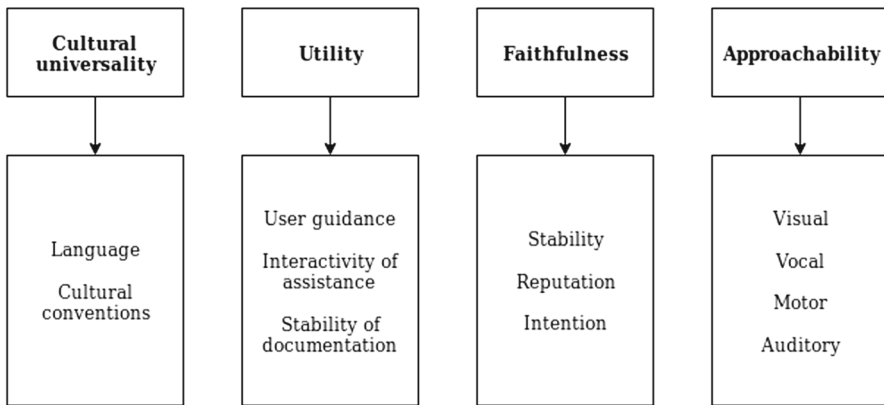


Fig. 2 The usability features and their characteristics of the factor, *Universality*

characteristics in hierarchical manner. The proposed model can be used by developers without any confusion to which characteristics they should consider while developing softwares so as to deliver a quality product. The hierarchical based usability model is given in the Fig. 1.

The model has seven sets of factors that are divided into features, which are further divided into characteristics in hierarchical manner [26]. The hierarchical based usability model is built by examining the existing taxonomy of usability and then constructing the features and their characteristics in hierarchical way. This model has 7 usability factors that are classified into 23 features as it can be seen in Fig. 1. The 7 usability factors are: *Universality*, *Security*, *Satisfaction*, *Productivity*, *Efficiency*, *Effectiveness*, and *Memorability*. Each of these factors has some features and have been described below.

1. *Universality* can be defined as a measurement to relate many cultural backgrounds of users measuring the real utilization of the product. This factor has 4 features that contain their own characteristics. The features are Cultural universality, utility, faithfulness and approachability. Its characteristics are given in Fig. 2.

Fig. 3 The usability features and their characteristics of the factor, *Security*

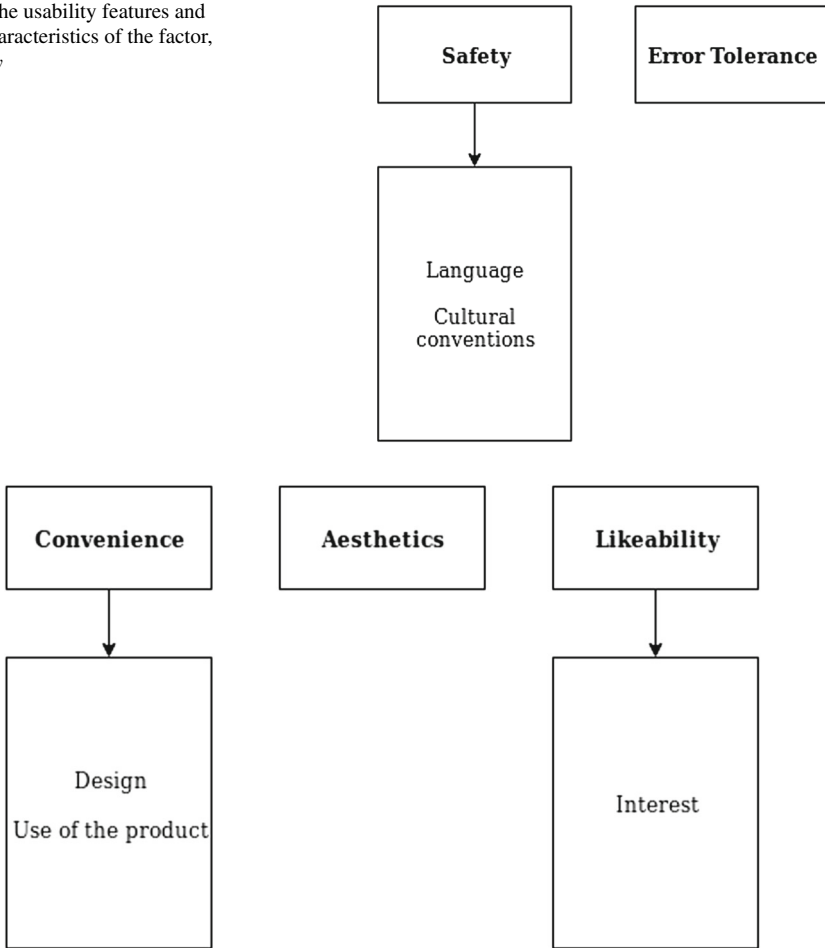


Fig. 4 The usability features and their characteristics of the factor, *Satisfaction*

2. *Security* is the measurement of degree of risk, failures (hardware and software), and damages. This usability factor has 2 features safety and Error Tolerance. The features and their characteristics are given in Fig. 3.
3. *Satisfaction* is the measurement of the satisfied feeling, response, likeability of the end users. This usability factor contains 3 features Convenience Aesthetics, and Likeability. The features and their characteristics are given in Fig. 4.
4. *Productivity* is the measurement of output obtained by the user from softwares. This usability factor has 1 feature viz. useful user task output. This feature does not have any characteristics, as it is self-defined.
5. *Efficiency* is the measurement of the ratio of desired output evaluated by user and total resources invested. This usability factor has 4 usability features: resources, economic cost, time and user effort. The features and their characteristics are given in Fig. 5.

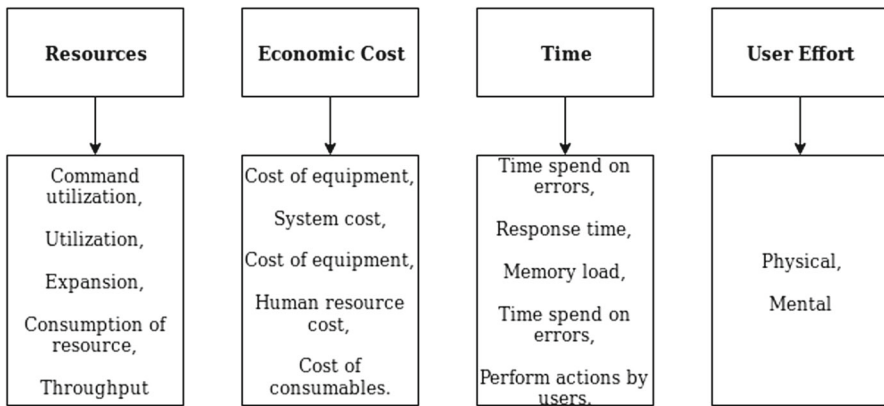


Fig. 5 The usability features and their characteristics of the factor, *Efficiency*

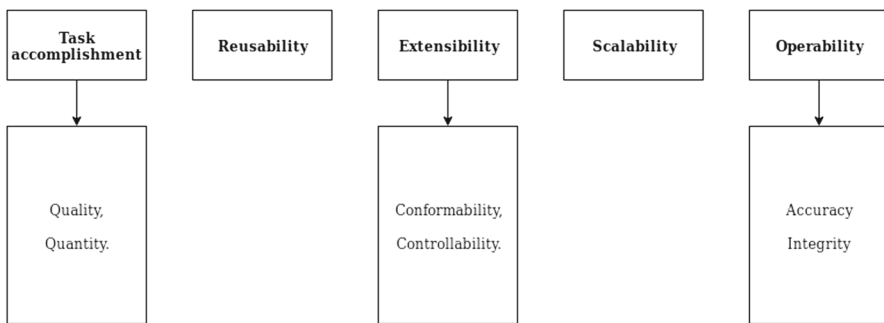


Fig. 6 The usability features and their characteristics of the factor, *Effectiveness*

6. *Effectiveness* is the measurement of accomplishment achieved by a user while performing a task with completeness and surety. This usability factor has 5 features Task accomplishment, Reusability, Extensibility, Scalability, Operability. The features and their characteristics are given in Fig. 6.
7. *Memorability* is the measurement of the ability of a user to learn or remember its elements with some clarity. This usability factor has 4 usability features: Comprehensibility, Learnability, and Consistency in structures, Memorability. The features and their characteristics are given in Fig. 7.

This hierarchical model as described above has 7 usability factors which is classified into 23 usability features and each features have some characteristics. The 7 usability factors have been explained above. The images are provided for all the characteristics of the 23 features. Thus making a hierarchical model that clears the confusion and inconsistency among the experts. Now these set of 23 features are used to prepare evaluation dataset [26]. This private usability dataset has been used for usability feature/attribute selection using the new proposed algorithm called Modified Moth-Flame Optimization algorithm.

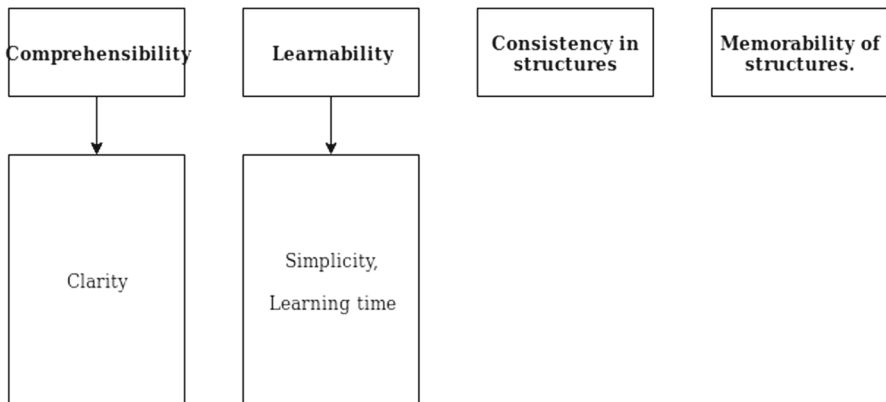


Fig. 7 The usability features and their characteristics of the factor, *Memorability*

3 Preliminaries

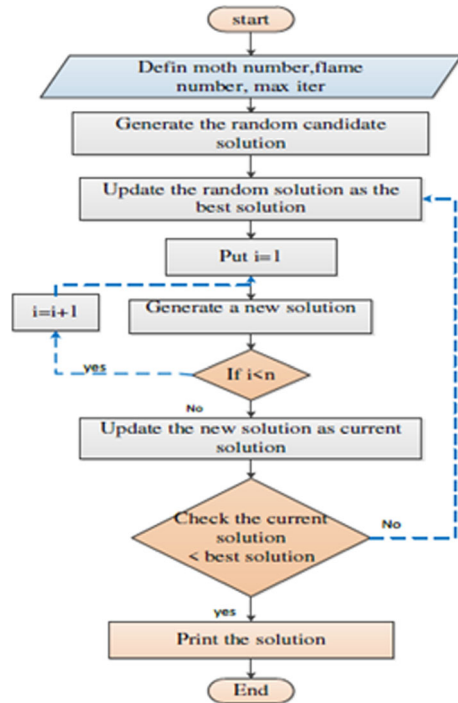
3.1 Moth-flame optimization algorithm

Moths are the fancy insects; they are immensely like to the family of butterflies. The navigation method of moths i.e. transverse orientation and the mathematical model of spiral flight update the position of moths and flame is the main inspiration of this optimizer. Also, this fantastic orientation mechanism makes moths being able to move in a straight line for the long distance.

A new nature-inspired optimization algorithm called Moth-flame optimization algorithm has been developed by Mirjalili [36] based on the behavior of the moths. To model this algorithm, Mirjalili [36] has idealized some necessary steps, as follows (Fig. 8):

- (1) Set the parameters and the range to be optimized and algorithm termination conditions;
- (2) Determining the fitness function of the MFO algorithm. The fitness function is an important index to describe the quality of population;
- (3) Initialize the algorithm parameters. Set the maximum number of iterations T , moth population size n , the search space dimension d and the maximum number of flame N , the current number of iterations $l=0$;
- (4) individual fitness value of the moth is calculated, and the individual position of the current best moth is found and saved as the flame fitness value matrix. Judging whether the algorithm iteration termination condition is satisfied, if it is, then go to step 8, otherwise, execute step 5;
- (5) Iterative process. The flame number is updated; the distance between flame and moth is calculated, and the moth-flame position is updated;
- (6) The calculation of the fitness value of individual moths;
- (7) To finds the current best moth individual spatial location. If the current position is better than the previous position, the current flame position is retained as optimal.

Fig. 8 Flowchart of Moth-flame optimization technique



Judging whether the iterative termination condition of the algorithm is satisfied, if yes, then go to step 8; otherwise, set $l = l + 1$; and perform steps 5 to 8;

- (8) The optimal individual value and the extreme global value are output. That is the final position of the flame and the corresponding fitness value.

3.2 Binary moth-flame optimization

The problem of binary encoding to obtain the optimal feature subset is an NP-complete problem. The Moth-flame optimization algorithm has been integrated into this binary encoding technique, so it is named as the Binary Moth-flame optimization algorithm [10].

MFO mimics the behavior of moths in a space. Moths move spirally to change their positions in the space; this spiral movement (SM) is the integral part of MFO algorithm as it evaluates the new positions of moths around flames, allowing moths to fly around the flame instead of the space between them. MFO uses wrapper-based method for feature selection by using k-nearest neighbor (KNN).

There is a problem with this feature selection approach i.e. at every iteration; the moth's position has to be a binary string as proposed by H. M. Zawbaa et al. [10]. Now, to ensure that change vector (δ) is binary, the position can be updated as given by Eqs. (1) and (2):

$$\delta_{M_i, F_j, d}^t = D_i \cdot e^{bt} \cdot \cos(2\pi t) \quad (1)$$

$$f_{\theta} = \alpha * E + (1 - \alpha) \frac{\sum_i \theta_i}{N} \quad (2)$$

4 Proposed MMFO for usability feature selection

In this section, the moth-flame optimization has been modified and extended for selection of usability features. We have already explained the hierarchical based usability model as a combination of seven factors, 23 attributes and 42 characteristics in a hierarchical manner in Sect. 2. The values of 7 factors are dependent on the 23 attributes, which are further dependent on 42 characteristics known as evaluating set. The evaluating set has its own set of values, 0 s and 1 s that means the absence and presence of features respectively.

Here, the modified moth-flame optimization algorithm (MMFO) for selection of usability features has been presented and explained. The main aim of MMFO is to select a subset Y_S of the complete set of features $Y = \{23 \text{ features from the 7 usability factors}\}$ in such a way that the resultant subset Y_S can predict the output Z with accuracy and convergence that is better compared to the performance when complete set Y is taken, and with significant decrease in the computational cost. The algorithm for the modified moth-flame optimization (MMFO) is given below in Algorithm 1.

ALGORITHM 1: Modified moth-flame optimization algorithm (MMFO) for usability feature selection

INPUT: n number of moths with a training set of 23 attributes/features and T number of iterations.

OUTPUT: A reduced set of important features.

```

1. Initialize parameters of moths
2.   Population size ← 23 (No of Features/Training set)
3.   Maximum number of Iterations ← 10
4.   Search domain ← [0, 1]
5.   b ← 0.75 -- defining the shape of the logarithmic spiral in MFO
6.   t ← (a - 1) * rand + 1
7.   a ← -1 + iteration * (-1/maximum iteration number)
8. Randomly generate an initial population
9. for i ← 1: n
10.  for j ← 1: d
11.    M [i, j] ← (ub(i) - lb(i)) * rand() + lb(i)           -- initial
solution
12.  end
13. end
14. set iteration ← 1
15. While(Iteration ≤ Maximum number of iterations)
16.   Flame no ← round (N - Iteration * N-1/T)           -- Update flame no
17.   OM ← FitnessFunction(M);                          -- calculate fitness
function
18. if iteration == 1
19.   F ← sort(M);
20.   OF ← sort(OM);
21. else
22.   F ← sort(Mt-1, Mt);
23.   OF ← sort(Mt-1, Mt);
24. end
25. for i ← 1: n
26.   for j ← 1: d
27.     Update r and t           -- r is convergence constant,
and t is a random no
28.     D ← |Fj - Mi|           -- calculate D for
corresponding moth
29.     Mi ← S (Mi, Fj)       -- Update M(i, j)
30.     S (Mi, Fj) ← Di * ebt * cos (2πt) + Fj
31.   end
32. end
33. For each feature j (∀j ← 1, ..., n), do
34.   Feature ← flameposition[j]
35.  global ← Group Feature into datasets # [population]

```

Following changes has been proposed in the modified moth-flame optimization algorithm as compared to standard moth-flame optimization:

- In lines 1–7, the parameters, range and termination condition for the moth-flame is initialized.
- In Lines 9–13, the initial solution is generated, i.e., M[i, j] has been calculated using the following Eq. 3.

$$M[i, j] \leftarrow (ub(i) - lb(i)) * rand() + lb(i) \quad (3)$$

- Also, in lines 15–17, the flame number is updated and determines fitness function until the number of iterations.

Flame number can be computed as given in Eq. (4).

$$Flameno = round\left(N - l * \frac{N - 1}{T}\right) \quad (4)$$

- In lines 18–32, updates the values of r, and t. The flame number is updated; the distance between flame and moth is calculated, and the moth-flame position is updated. The iteration continues until the termination condition is reached. D is calculated using following Eq. 4.

$$D = D_i = |F_j - M_i| \quad (5)$$

- In lines 33–35, computes an optimal subset of important usability attributes/features. Furthermore, the selection algorithm has been used to select the best dataset using the following Eq. (6):

$$Sum_i \leftarrow Sum_i + \frac{fact_main_i[j]}{global[j]} \quad (6)$$

where

fact_maini ← Group facti into datasets #[moths for each dataset]

facti[j] ← Assign attributes present in dataset

5 Implementation of MMFO for usability feature selection

In this section, we will discuss the implementation details of the proposed optimization algorithm. The proposed modified MFO has been implemented using MATLAB. Experimental setup, input parameters, and the dataset have been discussed as follows.

5.1 Experimental setup

The experimental setup includes a personal computer to test the algorithm with 1.80 GHz Core (TM) 2 CPU and 2 GB of memory under Ubuntu 10.04.4. The proposed MMFO algorithm has been implemented in two main stages/rounds: In the first round, the set of important usability features using the MMFO algorithm have been determined, and the accuracy and convergence have been computed during the second round.

5.2 Input parameters

The input parameters to implement the binary version of MFO and MMFO algorithms:

- t should be in the range of [− 1, 1].

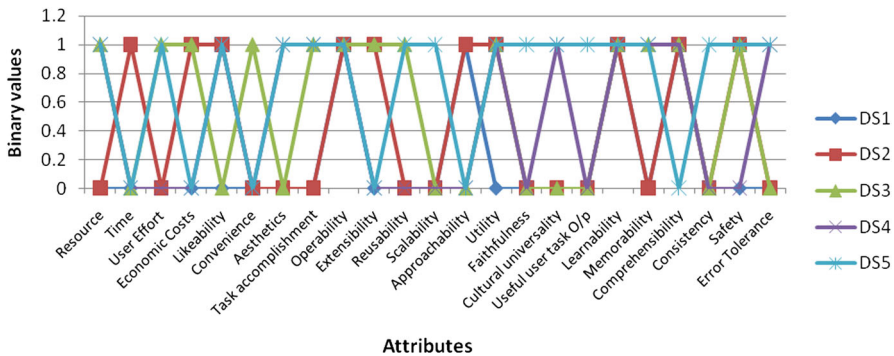


Fig. 9 Illustration of the used dataset

- b for the shape of a spiral is 0.75.
- random search domain is $[0,1]$.

During the implementation, both the algorithms have the problem dimension (d) equal to the number of features with maximum number of iterations is 100.

5.3 The dataset

The complete set of 23 features identified for hierarchical based software usability model depicted in Sect. 2 is evaluating set. Each evaluating set has binary values i.e. either 1 s or 0 s, indicating the presence or absence of a feature respectively. The training set has been prepared as the result of a survey given to the people. The people involved in the survey are either usability experts or engineering faculties or undergraduate engineering students. Five dataset related to the software applications has been given to the people to specify the presence and absence of the 23 attributes defined in the hierarchical software usability model. A total of 312 participants have been involved in the survey.

The participants were given a form to fill their views for five different software applications i.e. DS1, DS2, DS3, DS4, and DS5. The surveys have been conducted and it consists of 39 questions based on the characteristics of the different software applications. The result of the surveys/study has been used to prepare the dataset having 23 features. Figure 9 represents the values of the 23 features based on the survey. These five datasets will be applied after selecting the important features using proposed algorithm to identify the best dataset.

6 Results analysis and discussion

In this section, we will discuss the results when the dataset has been applied to the MMFO algorithm. We have conducted cross-validation over ten runs. The plot of selected attributes and accuracy over ten iterations has been presented in Figs. 10 and 11 respectively.

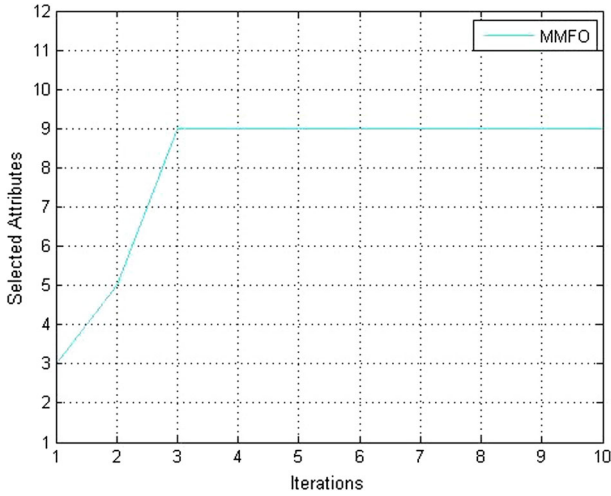


Fig. 10 Plot of selected attributes versus 10 iterations

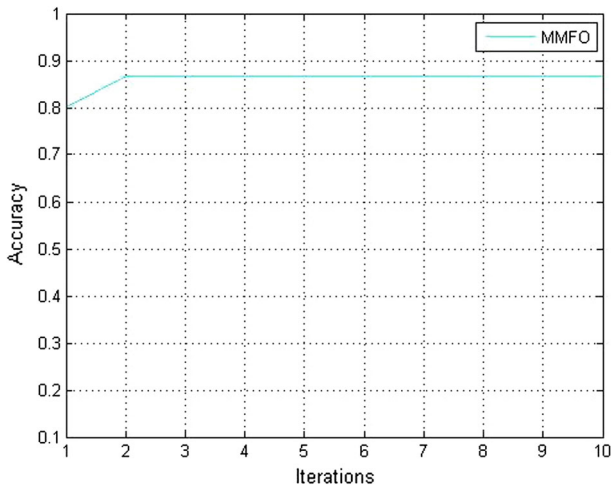


Fig. 11 Plot of accuracy versus 10 iterations

MMFO algorithm selects nine attributes with maximum accuracy and convergence as presented in Figs. 10, 11 and 12 respectively. The selected nine features/attributes are Economic cost, Task accomplishment, Operability, Scalability, Likeability, Learnability, Safety, Utility, Cultural universality.

7 Comparison of the proposed MMFO with MFO, BBAT and MBBAT [7]

In this section, we have presented a comparison of the proposed optimization algorithm with the related another optimization algorithm in the literature. The results of the

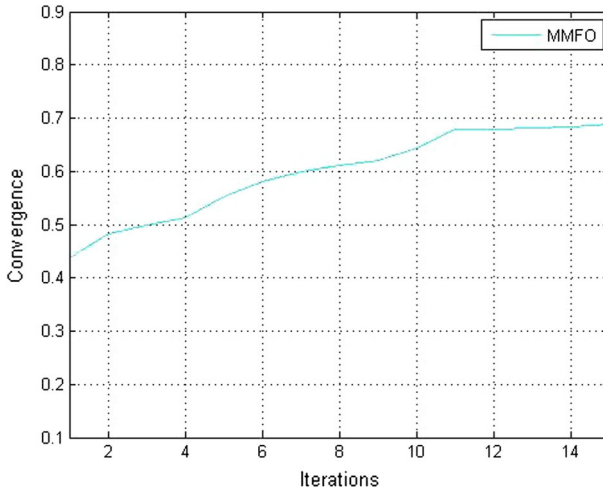


Fig. 12 Plot of convergence versus iterations

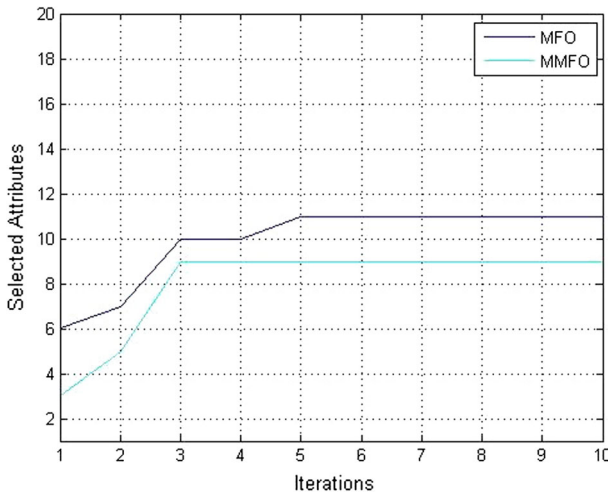


Fig. 13 Plot of selected attributes versus 10 iterations for MFO and MMFO

MMFO algorithm have been compared with the results of standard MFO. The plot of selected attributes of MMFO and MFO over ten iterations is presented in Fig. 13. The accuracy and convergence of selected attributes are presented in Figs. 14 and 15 respectively.

Figure 13 shows the number of selected attributes by MFO i.e. 11 attributes is higher as compared to MMFO i.e. 9 attributes. The accuracy has been determined and can be seen in Fig. 14. Hence, the proposed MMFO algorithm outperforms the standard MFO.

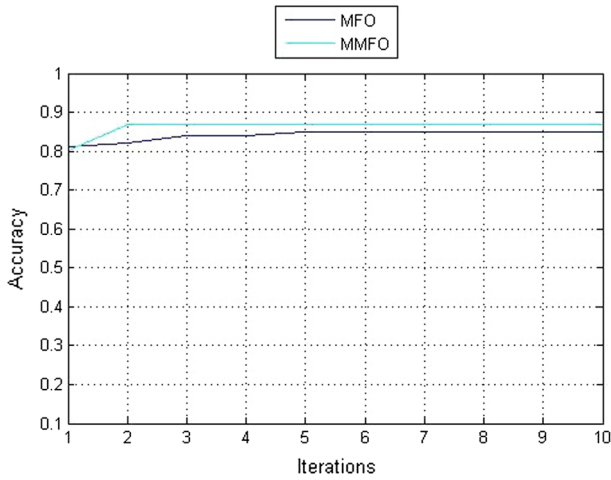


Fig. 14 Plot of accuracy versus 10 iterations for MFO and MMFO

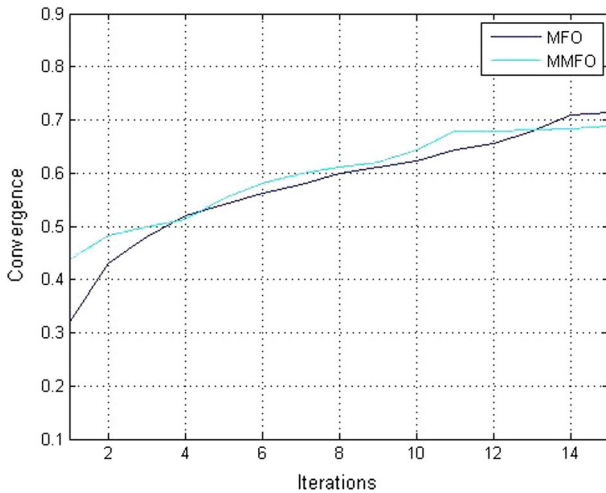


Fig. 15 Plot of convergence versus iterations for MFO and MMFO

Now, the proposed MMFO have been compared with BBAT and MBBAT [7]. The plot of selected attributes of MMFO, MFO, BBAT, and MBBAT over ten iterations is presented in Fig. 16. The accuracy over selected attributes is presented in Fig. 17.

As seen in Figs. 11 and 12, MMFO outperforms MFO, BBAT, and MBBAT algorithms as the number of selected attributes in MMFO is minimum and accuracy is maximum.

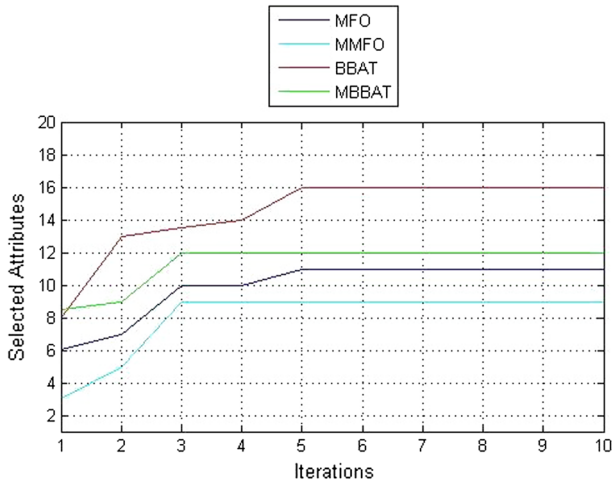


Fig. 16 Plot of selected attributes versus 10 iterations for MFO, MMFO, BBAT, and MBBAT

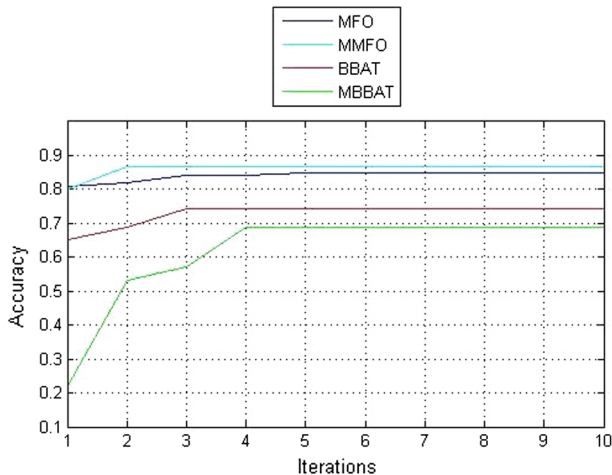


Fig. 17 Plot of accuracy versus 10 iterations for MFO, MMFO, BBAT, and MBBAT

8 Conclusion and future works

In this work, the term ‘software usability’ using hierarchical usability model has been defined. We have identified 7 factors with their 23 features in the hierarchical based usability model. This paper introduces a modified moth-flame optimization (MMFO) algorithm for usability feature selection. The MMFO algorithm reduces the number of features and retains a subset of relevant attributes without degrading the performances of the system. The modified algorithm (MMFO) has been introduced using Algorithm 1, which are used to generate the subset of features and the accuracy, convergence values. The proposed optimization algorithm outperforms the other related algorithm as it predicts the fewer number of feature and have better accuracy. The proposed

method can be used by researchers for predicting the usability of various softwares or Web applications.

Acknowledgements This work was supported by FCT/MCTES through national funds and when applicable co-funded EU funds under the Project UIDB/EEA/50008/2020; and by Brazilian National Council for Research and Development (CNPq) via Grant No. 309335/2017-5.

References

1. ISO 9126: Information Technology-Software Product Evaluation-Quality Characteristics and Guidelines for their Use. Geneva (1991)
2. International Organization for Standardization.: ISO 9241-11:1998, Ergonomic requirements for office work with visual display terminals (VDTs), Part 11: Guidance on usability. Geneva, Switzerland: Author, 1998
3. Institute of Electrical and Electronics Engineers.: IEEE Standard Glossary of Software Engineering Terminology, IEEE Std. 610.12-1990. Los Alamitos, CA: Author (1990)
4. Yang, X.-S., Deb, S.: Cuckoo search via Levy flights. In: Proceedings of World Congress on Nature and Biologically Inspired Computing (NaBIC), Coimbatore, India, p. 210E4 (2009)
5. Yang, X.-S.: A new metaheuristic bat-inspired algorithm. Department of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, UK (2010)
6. Sammut C, Webb GI (2010) Feature selection. In: Sammut C, Webb GI (eds) Encyclopedia of Machine Learning. Springer, New York, pp 429–433
7. Gupta D, Ahlawat A (2017) Usability feature selection via MBBAT: a novel approach. *J Comput Sci.* <https://doi.org/10.1016/j.jocs.2017.06.005>
8. Emary E, Zawbaa H, Hassanien A (2016) Binary gray wolf optimization approaches for feature selection. *Neurocomputing* 172:371–381
9. He YY, Zhou JZ, Li CS (2008) A precise chaotic particle swarm optimization algorithm based on improved tent map. *ICNC* 7:569–573
10. Hossam Mohammed Zawbaa Ismail (2016). Computational intelligence modeling of pharmaceutical roll compaction. Ph.D. thesis, Faculty of Mathematics and Computer Science, Babes-Bolyai University, May (2016)
11. Seffah A, Donyaee M, Kline RB, Padda HK (2006) Usability measurement and metrics: a consolidated model. *Software Qual J* 14:159–178
12. Abran A, Khelifi A, Suryan W (2003) Usability meanings and interpretations in ISO standards. *Software Qual J* 11:325–338
13. Alonso-Rios D, Vazquez-Garsia A, Mosqueria E, Moret-Bonillo V (2010) Usability: a Critical Analysis and a Taxonomy. *International Journal of Human-Computer Interaction* 26(1):53–74
14. Boëhm B (1978) Characteristics of Software Quality. Vol 1 of TRW Series on Software Technology. North-Holland, Amsterdam
15. Shackel B (1991) Usability—context, framework, definition, design, and evaluation. In: Shackel B, Richardson SJ (eds) *Human Factors for Informatics Usability*. Cambridge University Press, New York, pp 21–37
16. Nigel Bevan. Quality in use: Meeting user needs for quality. *Journal of System and Software* (1999)
17. Shneiderman B, Plaisant C (2005) *Designing the user interface: strategies for effective human-computer interaction*. Addison-Wesley, Boston
18. McCall JA, Richards PK, Walters GF (1977) *Factors in Software Quality*, vol II. Rome Aid Defence Centre, Amsterdam
19. Nielsen J (1993) *Usability Engineering*. Academic Press, London
20. Preece J, Benyon D, Davies G, Keller L, Rogers Y (1993) *A guide to usability: human factors in computing*. Addison-Wesley, Reading
21. Bass L, John BE (2003) Linking usability to software architecture patterns through general scenarios. *J Syst Softw* 66(3):187–197
22. Donyaee, M., Seffah, A.: QUIM: an integrated model for specifying and measuring quality in use. In: Eighth IFIP conference on human-computer interaction, Tokyo, Japan (2001)

23. Bevan, N., Kirakowsk, I. J., Maissel, J.: What is usability? In: Proceedings of the 4th International Conference on HCI, pp. 651–655 (1991)
24. Dix A, Finaly J, Abowd D, Beale R (1998) Human-Computer Interaction, 2nd edn. Prentice-Hall, Upper Saddle River (ISBN:978-0-13-239864-0)
25. Boehm B (1988) A spiral model of software development and enhancement. *IEEE Computers* 21(5):61–72
26. Gupta, D., Ahlawat, A., Sagar, K.: A critical analysis of a hierarchical based usability model. In: 2014 International Conference on Contemporary Computing and Informatics (IC3I), 27–29 Nov. 2014, Mysore, IEEE. <https://doi.org/10.1109/ic3i.2014.7019810>
27. Gupta D, Ahlawat A (2018) Taxonomy of GUM and usability prediction using GUM multistage fuzzy expert system. *Int. Arab J. Inf. Technol.* 16:357–363
28. Gupta D, Ahlawat A (2016) Usability determination using multistage fuzzy system. *Procedia Comput. Sci.* <https://doi.org/10.1016/j.procs.2016.02.042>
29. Gupta D, Ahlawat A (2016) Usability evaluation of live auction portal. *Int. J. Control Theory Appl.* 9(40):491–499
30. Gupta D, Ahlawat A (2017) Usability prediction of live auction using multistage fuzzy system. *Int. J. Artif. Intell. Appl. Smart Dev.* 5(1):11–20
31. Gupta D, Ahlawat A, Sagar K (2017) Usability prediction and ranking of SDLC models using fuzzy hierarchical usability model. *Open Eng. (Central Eur. J. Eng.) ESCI, SCOPUS* 7(1):161–168
32. Gupta D, Khanna A (2017) Software Usability Datasets. *Int. J. Pure Appl. Math. SCOPUS.* 117(15):1001–1014
33. Gupta D, Sagar K (2010) Remote file synchronization single-round algorithm. *Int J Comput Appl* 4(1):32–36
34. Gupta D, Rodrigues JJPC, Sundaram S, Khanna A, Korotaev V, Albuquerque VHC (2018) Usability feature extraction using modified crow search algorithm: a novel approach. *Neural Comput Appl.* <https://doi.org/10.1007/s00521-018-3688-6>
35. Jain, R., Gupta, D., Khanna, A.: Usability feature optimization using MWOA. In: International Conference on Innovative Computing and Communication (ICICC), vol. 2 (2018)
36. Mirjalili Seyedali (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. *Knowl. Based Syst.* 89(2015):228–249

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.