



# Coalescence of Evolutionary Multi-Objective Decision Making approach and Genetic Programming for Selection of Software Quality Parameter

Manuj Darbari  
Department of  
Information  
Technology, BBD  
University, Lucknow,  
India

Himanshu  
Pandey  
Department of  
Information  
Technology, BBD  
University, Lucknow,  
India

V.K Singh  
Department of  
Information  
Technology, BBD  
University, Lucknow,  
India

Gaurav Kumar  
Srivastava  
Department of  
Information  
Technology, BBD  
University, Lucknow,  
India

## ABSTRACT

Selection of quality parameters for software according to customer expectation is a complex task which can be prospected as a constrained multi-objective optimization and a multiple criteria decision making problem. For a Software Quality: Usability, Reliability, Complexity, Capability, Durability, Maintainability are the major factors affecting its performance. We proffer a concept of a Multi-Objective Decision making approach using Genetic Programming to appraising the Software Quality Parameters. The paper highlights estimating the Quality Parameters of Software using Multi objective Decision Making approaches and Genetic Programming. The outcome of a Multi objective fed into Genetic Programming for further mutation, to find out the perfect combination of variables of these quantities. The above work is substantiating an optimum trade-off needs to be reached in the formation of good software.

## Keywords

Software Quality Parameters, Multi objective Decision Making approach and Genetic Programming.

## 1. INTRODUCTION

Selection of a most incorporable Quality parameter for any software is vital to its quality standard. Software Quality is one of the most consequential targets to attain in the software development process for the successful software project. Software Quality activities are directed in whole process of project life cycle to deliver into the quality and reliability of the software phases. In each traditional development process software quality activities are performed.

Software quality is most important factor now days. It is now days no more a luxury rather than an optional requirement. A component, process or system fulfils the requirements, needs or expectations of customer then the degree is called Software Quality. Software organization tested the software by various algorithms and methods to increase the Quality and Reliability of the software. To achieve the quality standards of the software, we must have known the most incorporable factors and parameters of the software. These are the parameters that add up to the qualities of software are:-

### 1.1 Capability (functionality)

Capability describes the function amount which is combined by a product or software. Functional requirement occupies the

entire customer requirement. Basically capability of software tells us that for which reason software is being developed. There is some sub characteristics of functionality are: Interoperability, Suitability, Accuracy, Compliance, and Security.

### 1.2 Usability

Software Usability is as the extent to which the product is convenient and particular to use. Software Usability is a grouped from of Learnability, Operability, Understandability of the product or software to the final user (customer). Usability depends on these parameters- comfort level, ease to use, simplicity etc.

### 1.3 Performance

Communication Failure, Hardware of component failure affects the performance of software. We can evaluate the performance of software by these parameters: Execution Time, Service Unit Reduction, Idle Time Reduction, Number of tasks completed. By these technologies like (multiple servers running on different machines, thread management, improved page design), we can create high performance application.

### 1.4 Maintainability

Software Maintainability has a capability to modify the software product. Modification includes corrections, improvements or adaptation to change in requirements and environment. Sub attributes of maintainability are: Analyzability, Stability, and Testability.

### 1.5 Durability

Software usability improves Software Durability. It has two factors- Data Durability and Session Durability. To enhance the durability of the software we used technologies like Data Replication and Data Repair.

### 1.6 Serviceability

Ability of commit services by the software or application is called Serviceability. It deals with software in terms of technical help, user manual and problem devolvement.

### 1.7 Availability

It is the measure of how likely the system is ready for use. It provides the repair or restart time into account.



## 1.8 Complexity

It has 2 types

### 1.8.1 Apparent Complexity

A design or implementation is typical to comprehend and verify so that type of complexity is called Apparent Complexity.

### 1.8.2 Inherent Complexity

The factors like the number and intricacy of interfaces and number of conditional branches are responsible for Inherent Complexity.

One of the strongest critiques of current customer is that they are not able to express their needs and expectations for the product or software. So it is very perplexed to identify and list these most incorporable factors and parameters from various factors and parameters.

In Software Engineering, Multi objective Decision Making is an approach that capable to filter those software parameters which plays major role to achieve the quality standard for software. By implementing the Multi objective Decision Making approaches in various software parameters and user requirements, Software organizations become capable to identify the prioritized customer requirements and software parameters. But after applying Multi objective Decision Making approaches we are not able to identify all the software quality parameters and factors. In order to increase the percentage of identification of software quality parameters we used Genetic Programming.

In evolutionary algorithm Genetic Programming provides a multipurpose and most powerful optimization tool which starts from initial set of solution to derive new and possibly better solution. In order to mutate Software Quality parameters we use Genetic Programming.

The combination of Multi objective Decision making approach and Genetic Programming help us to identify the appropriate software quality parameters to achieving the quality standard of the software and this combination would be a useful tool for managers of large software project.

## 2. RELATED WORK

The paper by Faez Ahmed, KalyanmoyDeband, Abhilash Jindal on 'Evolutionary Multi-Objective Optimization and Decision Making Approaches to Cricket Team Selection' derived a novel representation scheme and a multi-objective approach using the NSGA-II algorithm. We are inspired by the work of Hashem [1], M.M.A. which highlights the Global optimization through a new evolutionary algorithm. TIAN Na, CHE A-da [7] in his paper 'Goal Programming in Quality Function Deployment Using Genetic Algorithm' clubbed the methods QFD and Genetic Algorithm. We have used this relation of combination to identify the software quality parameters. K.Y. Chan1, T.S. Dillon1, C.K. Kwong2 and S.H. Ling [8] proposed that GP based method produce a more accurate and interpretable models than the other commonly used methods like QFD. They show this on his paper "Using Genetic Programming for Developing Relationship between Engineering Characteristics and Customer Requirements in New product". Norberto EijiNawa and Takeshi Furuhashi [2] used bacterial evolutionary algorithm to discover Fuzzy system parameters. The paper 'Feature Selection and Clustering in Software Quality Prediction' by Qi Wang, Jie Zhu, Bo Yu [9] presents a new software quality prediction model based on genetic algorithm (GA) in which outlier

detection and feature selection are executed simultaneously. Antonio Gonzalez and Francisco Hemera[3] worked on Iterative Rule Learning Approach in 1997. The concept of Fuzzy logic techniques are utilized in software reliability engineering. This concept is given by Xu, Z in 2001. Yi. Liu, TaghiM.Khoshgoftaar [12] in Genetic Programming Model for Software Quality Classification compared the two methodologies LRM and GP and shows that GP model is much better than LRM. S. Keshavarz and Reza Javidan [13] deals with Software Quality Control with criteria of covering application and proposed a new method based on genetic algorithm for generating optimal test data. In this paper 'Comparison of Software Quality Models: An Analytical Approach' S.K Dubey, SoumiGhosh, A Rana [11] analyse the qualitative characteristics and side-by-side determine the software quality. Paper by Salah Bouktif, Bal'azsK'egl, HouariSahraouiIn [14] combines Software Quality Predictive Models. We are inspired by the work on Combining Models because combined models works well on the particular system or in the particular type of organization. David A and Gary B. Lamont [5] focus on Messy Genetic Algorithms for Multi Objective Optimization. We are inspired by the work of Linda Murphy, Hoda S. Abdel-Aty-Zohd, M.Hashem-Sherif [10] they tracked the Genetic Algorithm Model for Product Deployment in Telecom Services. They have used the genetic algorithm to update the parameters applied to the input measurements to find the optimum solution for the defect tracking model system. HillolKargupta [15] introduces the gene expression messy genetic algorithm (GEMGA). It directly searches for relations among the member of the search space. We have used the Messy Genetic Algorithm to find the appropriate customer requirements and engineering characteristic in our paper. Lastly the paper by Kohei Arai [16] compares the conventional simple Genetic Algorithm to Messy Genetic Algorithm. This highlights the importance of Messy Genetic Algorithm for clustering. We have extended the concept of indexing from messy genetic algorithm to identify the highly prioritized customer requirements and engineering characters.

## 3. OUR FRAMEWORK

The design of software application focuses making a decision such that end user's objective is maximized without increasing the complexity of the system. In simpler terms it means a decision has to be generated which can provide a balance between "Ease of Use" and "Feature Rich Capabilities". In order to find out the best outcome of relation between two parameters, we have to follow two main objectives:

To gather information satisfies the objective using available alternatives.

Secondly assigning the priority on each of the available alternatives.

Multi objective deals with selection of the best alternative from Universe of Discourse elements as:

$$S = \{S_1, S_2, \dots, S_i, \dots, S_n\}$$

and set of "m" alternatives as:

$$A = \{a_1, a_2, \dots, a_i, \dots, a_m\}$$

Where  $a_i$  indicates the  $i_{th}$  objective. The degree of alternatives 's' in  $a_i$ , denoted  $ua_i(s)$ .



Keeping the software quality variables into alternative and objective we get the equation as:

$$A = \{ \text{Understandability, Operability, Learnability, Context Shift, Navigational Guidance, System Feedback, and Identification of Inconsistency} \} \dots (1)$$

$$S = \{ \text{Usability, Reliability} \} \dots (2)$$

Using the above two equations we can form a Decision Function (DF) taking case that it satisfies the entire decision objective. The set of Objectives are:

$$DF = \{ \text{Usability} \cap \text{Reliability} \} \dots (3)$$

The grade of membership that Decision Function for each alternative 'a' is defined

$$\mu_{DF}(\text{Understandability}) = \min [\mu_{\text{Usability}}(\text{Understandability}), \mu_{\text{Reliability}}(\text{Understandability})]$$

$$\mu_{DF}(\text{Operability}) = \min [\mu_{\text{Usability}}(\text{Operability}), \mu_{\text{Reliability}}(\text{Operability})]$$

$$\mu_{DF}(\text{Learnability}) = \min [\mu_{\text{Usability}}(\text{Learnability}), \mu_{\text{Reliability}}(\text{Learnability})]$$

$$\mu_{DF}(\text{Context Shift}) = \min [\mu_{\text{Usability}}(\text{Context Shift}), \mu_{\text{Reliability}}(\text{Context Shift})]$$

$$\mu_{DF}(\text{Navigational Guidance}) = \min [\mu_{\text{Usability}}(\text{Navigational Guidance}), \mu_{\text{Reliability}}(\text{Navigational Guidance})]$$

Similarly we can generate for all the "alternatives" of Objectives.

The Optimal Decision a\* will then be the alternatives that satisfies the equation

$$\mu_{DF}(a^*) = \max_{a \in A} [\mu_{DF}(a)]$$

$$\mu_{DF}(\text{alternative})_{\text{optimal}} = \max_{a \in A} [\mu_{DF}(A)]$$

(Where A= Alternative)

The performances set P contains the parameters  $Z_i, i = 1$  to  $m$ .

$$\{P\} = \{ \text{Prioritized Voice of Customer, Prioritized feature that Company can offer} \}.$$

The preferences are than corresponded with each of the objectives in order to generate Decision Measure (DM) which involves objective and preferences.

$$DM = DM(\text{Objectives, Preferences})$$

$$\text{Since } DF = DM(o_1, b_1) \wedge DM(o_2, b_2) \wedge \dots \wedge DM(o_m, B_m) \dots (4)$$

From the above set we are able to develop the

$$O_1 = \text{High Usability}$$

$$O_2 = \text{High Reliability.}$$

On solving this we get a fuzzy set consisting set of Alternative and Universe of Discourse as

$$O_{1(\text{Usability})} = 0.9/\text{Understandability} + 0.9/\text{Operability} + 0.8/\text{Learnability} + 0.9/\text{Context Shift} + 0.9/\text{Navigational Guidance} + 0.4/\text{System Feedback} + 0.1/\text{Identification of Inconsistency}$$

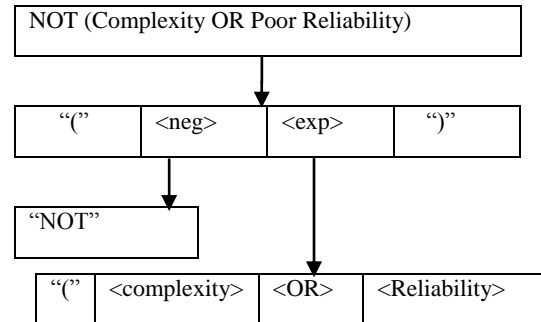
$$O_{2(\text{Reliability})} = 0.1/\text{Understandability} + 0.9/\text{Operability} + 0.1/\text{Learnability} + 0.4/\text{Context Shift} + 0.1/\text{Navigational Guidance} + 0.6/\text{System Feedback} + 0.9/\text{Identification of Inconsistency}$$

In order to maximize DF we have the function:

$$O_1 \wedge O_2 = 0.1/\text{Understandability} + 0.9/\text{Operability} + 0.1/\text{Learnability} + 0.4/\text{Context Shift} + 0.1/\text{Navigational Guidance} + 0.4/\text{System Feedback} + 0.1/\text{Identification of Inconsistency}$$

The Resultant output parameters are the final value of DF (Decision Function).

In order to find out exact optimal solution for the above given solution of Decision Value we apply the result to Derivation tree using Genetic Programming.



$$f(F) = \sum_{i=1}^N (y_i - F(x_i))^2 \dots (5)$$

The above formulae can be enhanced by the value obtained from the successive mutation of the function:-

$$f_{\text{Reliability}}(\text{Usability}) = f(\text{Usability}) -$$

$$\max_{j=1}^m f(\text{Usability})$$

For maximum value of the function

$$f_{\text{Reliability}}(\text{Usability}) = f(\text{Usability}_{j,t}) -$$

$$\min_{j=1}^m f(\text{Usability}_{j,t})$$

For  $f$  has to be minimized ..... (6)

On solving the above equation for maximum parameters we get the following result.

Let population size  $N=20$

Iteration cycle = 25

Crossover probability  $p_c = 0.7$

Mutation probability  $p_m = 0.01$

Average Fitness =  $16/4 = 4.0$

Chromosome

O: Operability: 0001

U: Understandability: 1010

L: Learnability: 1100

N: Navigational Guidance: 1110

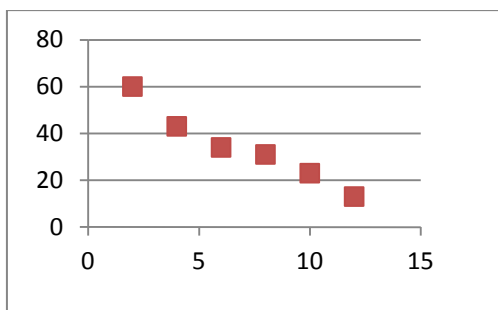


**Table 1- Fitness Value Calculation**

Fitness Value	Probability of string $p_i$	% Probability
05	5/16	5/16 *100= 31.2
06	6/16	6/16*100= 37.5
04	4/16	4/16*100= 25
01	1/16	1/16*100= 6.25

Now from the Table 1 using the Crossover Probability ‘ $p_c$ ’ and Mutation Probability ‘ $p_m$ ’ we are able to generate the final outcome using Matlab as:

“ON and LN” which means after iteration the final outcome is a combination of Operability and Navigational Guidance along with Learnability and Navigational Guidance.



**Fig 1: Realisation of Fitness Value.**

#### 4. CONCLUSION

The paper highlights selection of “Feature Rich” and “Ease of Use” values by using the concept of Multi Objective optimisation. The use of Decision measure concept generates the Decision Function in the above two clusters. These two clusters are then genetically mutated using the concept of Genetic Programming which gives the Final Fitness value in terms of combination of Operability and Navigational Guidance and Learnability and Navigational Guidance as the prime factors on which output is dependent.

#### 5. REFERENCES

[1] Hashem, M.M.A: 1999 “Global Optimization Through a new class of Evolutionary Algorithms”, PhD Dissertation, Saga University, Japan.

[2] H Ruotsalainen: January 2010. “Interactive Multiobjective Optimization in Model-based Decision Making with Applications”.

[3] S. RIOS-INSUA, J.G PACHON, A. MATEOS 1994: “A Method Of Multiobjective Decision Making Using a Vector Value Function”.

[4] M Srivastava, H Pandey, S Shukla, B.K Thakur, “A Literature Review of E- Learning Model Based on Semantic Web Technology”, International Journal of Scientific & Engineering Research, Volume 5, Issue 10, October -2014.

[5] J Verma, S Bansal, H Pandey, “Develop Framework for Selecting Best Software Development Methodology”,

International Journal of Scientific & Engineering Research, Volume 5, Issue 4, April -2014.

[6] Yonghua Zhou Yuliu Chen, “The QFD-based Decision-making Approach for Strategic BPR” Beijing 100084, P. R. China.

[7] TIAN Na, CHE A-da, August 20-22, 2007 “Goal Programming in Quality Function Deployment Using Genetic Algorithm”, International Conference on Management Science & Engineering (14th).

[8] K.Y. Chan<sup>1</sup>, T.S. Dillon<sup>1</sup>, C.K. Kwong<sup>2</sup> and S.H. Ling,<sup>”</sup> Using Genetic Programming for Developing Relationship between Engineering Characteristics and Customer Requirements in New Products” 1Digital Ecosystems and Business Intelligence Institute, Curtin University of Technology, Perth, Australia.

[9] Qi Wang, Jie Zhu, Bo Yu, 2007 “Feature Selection and Clustering in Software Quality Prediction”, Evaluation and Assessment in Software Engineering.

[10] L Murphy, H S. Abdel-Aty-Zohd, M. Hashem-Sherif, 2005 “A Genetic Algorithm Tracking Model For Product Deployment in Telecom Services” ,0-7803-9197-7/05/© IEEE.

[11] S K Dubey, S Ghosh, A Rana ,”Comparison of Software Quality Models: An Analytical Approach”, ISSN 2250-2459, Volume 2, Issue 2, February 2012.

[12] Yi. Liu, TaghiM.Khoshgoftaar , “Genetic Programming Model for Software Quality Classification”, 6th IEEE International Symposium on High Assurance Systems Engineering (HASE’01) 2001.

[13] S Keshavarz and R. Javidan, August 2011 “Software Quality Control Based on Genetic Algorithm”, Vol. 3, No. 4,.

[14] S Bouktif, Bal’azsK’egl, H. Sahraoui, “Combining Software Quality Predictive Models: An Evolutionary Approach”, Dept. of Computer Science and Op. Res., University of Montreal C.P. 6128 Succ. Centre-Ville, Canada.

[15] H. Kargupta, , 1996 “The Gene Expression Messy Genetic Algorithm”, published in IEEE conference on Evolutionary Computation, Nagoya, Japan.

[16] K. Arai, Vol. 1, No. 8, 2012 “Clustering Method Based on Messy Genetic Algorithm: GA for Remote Sensing Satellite Image Classifications”, International Journal of Advanced Research in Artificial Intelligence.

[17] R. Sobiesiak and Tim, “Complexity analysis: a quantitative approach to usability engineering”, IBM Rochester Laboratory.

[18] M Darbari, N Dhanda “Applying Constraints in Model Driven Knowledge Representation Framework.”, International Journal of Hybrid Information Technology 3 (3), 4, 2010.

[19] M Darbari, S Medhavi, AK Srivastava “Development of effective Urban Road Traffic Management using workflow techniques” for upcoming metro cities like Lucknow (India),Development 2 (2)4,2008.

[20] IA Siddiqui, M Darbari: “Application of Use Case for Identification of Root Cause of the Dependencies and



Mutual Understanding and Cooperation Difficulties in Software Systems”, *International Journal of Applied Software Engineering*, 4, 10-20.

- [21] IA Siddiqui, M Darbari, S Bansal, “Application of Activity Theory and Particle Swarm Optimization Technique in Cooperative Software Development”, *International Review on Computers & Software* 7 (5)
- [22] IA Siddiqui, M Darbari, “A Group Awareness and collaboration in Distributed Software Development”,

*International Journal of Scientific & Engineering Research* Volume 3, Issue 3, March -2012.

- [23] N EijiNawa and T Furuhashi, “Fuzzy system Parameters: Discovery by bacterial Evolutionary Algorithm”, *IEEE Transaction on Fuzzy system*. Volume 7, No 5, 1991.
- [24] A Gonzalez and F Hemera , “Multi- stage Genetic fuzzy systems based on the Iterative Rule Learning Approach “, *Math ware & Soft Computing* 4(1997).