International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 11 | Nov 2021 www.irjet.net p-ISSN: 2395-0072

RESOLVING RANKING ISSUE BY OPTIMIZING C&K METRICS FOR SOFTWARE MAINTENANCE EFFORT ESTIMATION

[1]***Ms. Kiran***, *Research Scholar, BMU, Asthal Bohar, Rohtak*
[2]***Dr. Mukesh Singla***, *Prof. Deptt. of Computer Science, BMU, Asthal Bohar, Rohtak*

---***---

Abstract: Software maintenance effort estimation is considered as a process that has been performed to estimate the actual maintenance effort during software development. Maintenance effort estimation has been considered time consuming and expensive activity. It is used to prioritize and calculate effort using different soft computing techniques. Techniques of soft computing that could be used during effort estimation have been described. Research work has considered traditional techniques customer requirement, Method coverage, Cost factors, Faulty severity. The contribution of the existing research has been described. The research work has provided the overall view of existing techniques of maintenance effort estimation is provided through literature survey. Techniques used in previous research papers of maintenance effort estimation are stated.

Research has proposed technique for effort estimation by optimizing the C&K metrics for object oriented testing. Proposed work is capable to prioritization for effort estimation in case of Object Oriented modules are performed considering C & K metrics having attribute NOC, WMC, DIT, RFC, CBO, and LCOM. Such technique supposed to be suitable to estimate the effort of maintance during object oriented programming. Research has enhanced complexity based approach for effort estimation by applying PSO to obtain optimal value of weight. Complexity of object oriented metrics namely WMC, DIT, NOC, LCOM, RFC, CBO would be utilized to obtain ranks for various modules. Proposed work provided a solution for effort estimation. Moreover, proposed work has reduced the possibility of same rank using weight mechanism. The total weight of all C& K parameter has been considered and ranking is made accordingly.

Keywords: DIT, CBO, WMC, LCOM, NOC, RFC, C&K Metrics

1. INTRODUCTION

Maintenance in software engineering is a change after delivery to fix defects, boost functionality or other characteristics of the software system. Maintenance is commonly used when only faults are fixed. A research has nevertheless shown that over 80% of the repair effort is being utilized for non-corrective measures. The users send bug feedback that interface improvements to the device actually perpetuate this perception. Latest reports also brought the bug-setting ratio to 21%. In 1969, Meir M. Lehman first discussed the management and development of software systems.

For 20 years his thesis has led Lehman's Laws to be drafted (Lehman 1997). Key empirical results show that maintenance is really evolutionary and maintenance choices are helped by an appreciation of what happens to processes (and software).

Lehman has shown that networks are evolving over time. They get increasingly complicated as they progress, until such measures are taken such as code reconstitution in order to minimize complexity. In the late 1970s, a well-known and frequently quoted Lientz and Swanson report revealed the very high level of repair costs.

The survey found that about 75% of repair was carried out in the first two forms and error correction consumed approximately 21%. Many experiments subsequently show a similar scale of problem. Studies have shown that end-user contribution is important for the collection and review of new requirement.

This is the primary source of all device development and maintenance problems. Computer servicing is significant because it takes a huge portion of the total cost of living which often ensures that market prospects are losing due to the failure to update software easily and efficiently.

Maintenance is described as the attempt to repair software system faults after general availability (GA).

Maintenance has been categorized in three sections:

1. Fixed errors that would be found in the code after being publicly functional are corrected for repair. The developer who patches a Java method that triggers a compilation error is an example of a corrective maintenance operation.

2. Adaptive maintenance means adapting the device to operate in a new setting, such as another topology network, platform or operating system. The developer who fixes a Java method which works on BEA WebLogic but is never on the IBM Websphere is a model of adaptive maintenance operation.

3. Perfect maintenance includes modifications which enable the programme to meet the same but more appropriate specifications. For instance, the creator might simply modify the code to make the device easier to manage or functional.

Then AME = 0,15 x 100 = fifteen persons-months is the basic annual maintenance effort calculation. In other terms, 15 months a year can be spent in this particular software project.

1.1 SOFT COMPUTING TECHNIQUES

In the simulation and modelling of the production processes of MMCs, the soft computer technology will run both numerical and experimental results.

1. Genetic Algorithm
2. Response Surface Methodology
3. Artificial Neural Network
4. Taguchi Method
5. Fuzzy Logic Optimization
6. Particle Swarm Optimization

1.2 C&K METRICS

C&K Metrics (DIT, CBO, WMC, LCOM, NOC, RFC**)** The Chidamber & Kemerer metrics suite is consisting of 6 metrics calculated for every class: DIT, WMC, NOC, CBO, RFC and LCOM. Three stages of Boochs recommended OOD process is discussed below:

1. Class identification (NOC, DIT, WMC)
2. Class Semantics (WMC, LCOM, RFC)
3. Relationships among classes (CBO, RFC)

Counting related to variety of classes which are connected in the company of definite class is represented by CBO. It means a situation in which methods related to particular class need or use factors of other. DIT represents how many classes are followed by an individual class. On the basis of example deepness, C&K suggests next results. Within a class of paired method which remains uncounted are represented by the LCOM cohesion. Definition of NOC is given by C&K number which is related to next subclasses of a class. Dimension related to feedback group are represented by the RFC. Definition related to feedback group in support of class is given by C&K in the form of groups connected in the company of methods. Complication of methods is represented by the WMC.

DIT, CBO, WMC, LCOM, NOC, RFC are explained below:

NOC: This is one of categorization of level oriented design in OOPS based system. This is a count of immediate derived classes subordinate to a super class in hierarchy of class.

WMC: It is a mechanism count in object oriented based system. This is known as count of function performed inside class.

RFC: Whenever a single object of a class gets information it carried out group of methodology depending upon on the information. Response for a class exactly relates to this group of methodology. **Depth of inheritance tree (DIT):** It becomes an optimal and traditional route of the class in the direction of its parent class in case of inheritance. The parent class of DIT has zero value.

Coupling between object classes (CBO): It deduces a close relationship with RFC. In this number of those categories are counted which have connected to special type of category.

Lack of cohesion methods (LCOM): Cohesion has been considered as a measure in case of lot of disconnected method pairs in a class.

1.3 PSO

PSO evolves into a technique of evaluation. It exists in the form of a technique that is easy to apply and put into practice on a regular basis. It has already been determined that such evaluation approaches uncover the best potential solution in a timely way. This strategy may be defined as a way that can optimize any issue in the realm of information technology. It has been noticed that in a PSO-based model, attempts are made one at a time to improve the performance of the candidate solution. It addresses any population-related problem with possible solutions. The dubbed particles travel about in search-space. This methodology works by applying an arithmetical rule on the particle's location and velocity. Its well-known domestic location has a significant influence on its mobility. This site has been modified with improved locations. Other particles may readily identify these sites. The swarm is predicted to migrate toward the best options as a result of this. PSO is a good heuristic because it makes minimal, if any, assumptions about the issue that has to be solved. Meta heuristics like PSO, on the other hand, do not ensure that an optimum solution will ever be identified. In the current circumstance, met heuristics are the most significant and beneficial since they have shown success in a variety of optimization issues when implemented. It's a self-contained system. It defined the degree to which these complex systems were active. In order to deal with optimization issues, a cooperative and intelligent structure employs an exceedingly streamlined model of social behavior.

2. PROBLEM FORMULATION

During software development the effort estimation is considered difficult task. However there have been many researchers who have presented the research in this field but they failed to provide optimized solution to detect the effort estimation. Moreover there is need to introduce optimization mechanism that should provide accurate result in less time. Thus there is need to introduce hybrid mechanism to achieve this objective. The proposed model is supposed to provide optimized solution by integration of PSO.

3. PROPOSED WORK

This section proposed a technique for effort estimation by optimizing the C&K metrics for object oriented testing. The proposed work has been proposed step by step. The whole procedure is described in detail. The proposed work has been defined by flow chart. Estimation of effort in case of Object Oriented modules is performed considering C & K metrics having attribute NOC, WMC, DIT, RFC, CBO, and LCOM. The proposed technique is suitable to prioritize effort during object oriented programming. The proposed methodology is shown:

ALGORITHM

```
1. Input object oriented module to get C&K metrics.
2. Get DIT, CBO, WMC, LCOM, NOC, RFC using "STAN TOOL [32]"
3. Create a table presenting DIT, CBO, WMC, LCOM, NOC, RFC for each package.
4. Get Upper bound (UB), lower bound (LB) and mean(M) of each column.
5. Get best optimized solution (b) using PSO for each column.
6. set i=1 repeat step 7,8,9 until i<=6
```

```
7. if LB(i)<b(i)<M(i)
      set weight(i)=1-value(i)/1000
8. if M(i)<b(i)<UB(i)
      set weight(i)=2-value(i)/1000
9. i=i+1
10. Create a selection matrix representing selection as 1 and not selected as 0.
11. Place weight in associated column of selection matrix.
12. Get sum of weight for each column of every package.
13. The package having maximum weight would be prioritized.
14. Stop
```

Calculation of threshold value for C & K metrics

1. Get the C & K metrics having attribute DIT, CBO, WMC, LCOM, NOC, RFC.
2. Find upper and lower bounds of each column.
3. Set the objective function [33] for PSO optimization.
4. Pass the dataset in objective function.
5. Get the optimal value of each attribute by executing PSO algorithm [34].

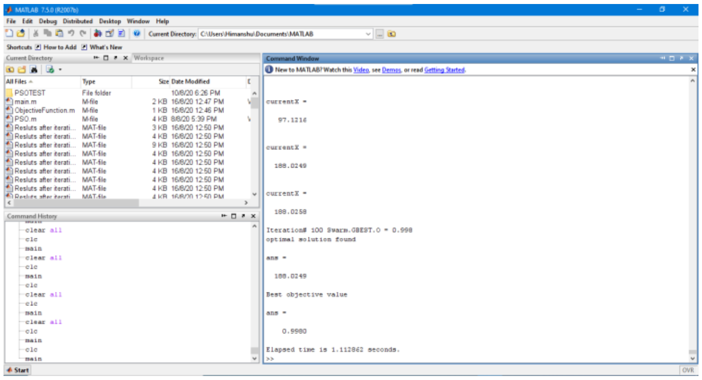
Assignment of weight to C &K metrics

1. Make the selection metrics considering 1 as selected if correspond value is less than optimal value and 0 as unselected if corresponding value is greater than optimal value. [11]
2. Get the lower, upper, optimal value and mean of all columns.
3. If optimal value lies between lower bound and mean, then set weight=1
4. If optimal values lie between mean and upper bound, then set weight=2 [9]
5. Set weight= weight- corresponding value/1000

4. SIMULATION AND RESULTS

RESULT OF PSO FOR WMC

After setting "main" command in MATLAB command prompt following result is shown below


MATLAB 7.5.0 (R2007b)
File Edit Debug Distributed Desktop Window Help
Current Directory: C:\Users\Himanshu\Documents\MATLAB
Shortcuts How to Add What's New
Current Directory
Workspace
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
Start
OVR

All Files	Type	Size	Date Modified
PSOTEST	File folder		10/8/20 6:26 PM
main.m	M-file	2 KB	16/8/20 12:47 PM
ObjectiveFunction.m	M-file	1 KB	16/8/20 12:46 PM
PSO.m	M-file	4 KB	8/8/20 5:39 PM
Resluts after iterati...	MAT-file	3 KB	16/8/20 12:50 PM
Resluts after iterati...	MAT-file	4 KB	16/8/20 12:50 PM
Resluts after iterati...	MAT-file	9 KB	16/8/20 12:50 PM
Resluts after iterati...	MAT-file	4 KB	16/8/20 12:50 PM
Resluts after iterati...	MAT-file	4 KB	16/8/20 12:50 PM
Resluts after iterati...	MAT-file	4 KB	16/8/20 12:50 PM
Resluts after iterati...	MAT-file	4 KB	16/8/20 12:50 PM
Resluts after iterati...	MAT-file	4 KB	16/8/20 12:50 PM
Resluts after iterati...	MAT-file	4 KB	16/8/20 12:50 PM

```
clear all
clc
main
clear all
clc
main
clc
clear all
clc
main
clear all
clc
main
clc
main
```

```
currentX =

   97.1216


currentX =

  188.0249


currentX =

  188.0258

Iteration# 100 Swarm.GBEST.O = 0.998
optimal solution found

ans =

  188.0249

Best objective value

ans =

    0.9980

Elapsed time is 1.112862 seconds.
>>
```

Fig 2 MATLAB Simulation for WMC

The optimal WMC solution is 188.02

Convergence curve shown during PSO implementation of WMC is show below:

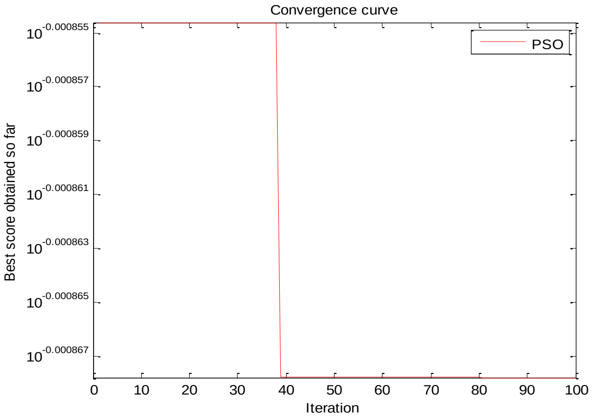
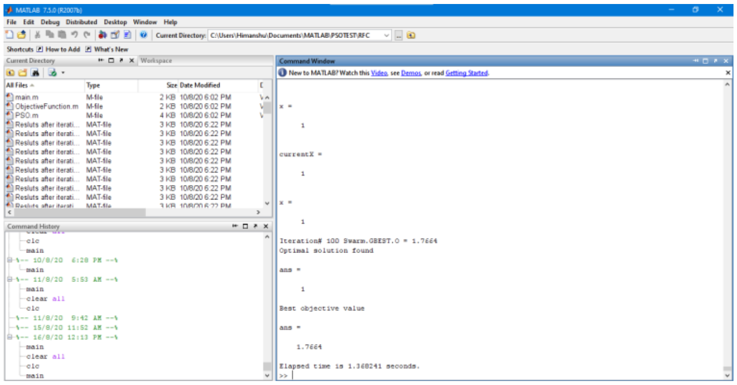


Fig 3 Convergence curve for WMC

RESULT OF PSO FOR RFC

After setting “main” command in MATLAB command prompt following result is shown below


MATLAB 7.5.0 (R2007b)
File Edit Debug Distributed Desktop Window Help
Current Directory: C:\Users\Himanshu\Documents\MATLAB\PSOTEST\RFC
Shortcuts How to Add What's New
Current Directory
Workspace
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History

All Files	Type	Size	Date Modified
main.m	M-file	2 KB	10/8/20 6:02 PM
ObjectiveFunction.m	M-file	2 KB	10/8/20 6:02 PM
PSO.m	M-file	4 KB	10/8/20 6:02 PM
Resluts after iterati...	MAT-file	3 KB	10/8/20 6:22 PM
Resluts after iterati...	MAT-file	3 KB	10/8/20 6:22 PM
Resluts after iterati...	MAT-file	3 KB	10/8/20 6:22 PM
Resluts after iterati...	MAT-file	3 KB	10/8/20 6:22 PM
Resluts after iterati...	MAT-file	3 KB	10/8/20 6:22 PM
Resluts after iterati...	MAT-file	3 KB	10/8/20 6:22 PM
Resluts after iterati...	MAT-file	3 KB	10/8/20 6:22 PM
Resluts after iterati...	MAT-file	3 KB	10/8/20 6:22 PM
Resluts after iterati...	MAT-file	3 KB	10/8/20 6:22 PM

```
  clc
  main
%-- 10/8/20  6:28 PM --%
  main
%-- 11/8/20  5:53 AM --%
  main
  clear all
  clc
%-- 11/8/20  9:42 AM --%
%-- 15/8/20 11:52 AM --%
%-- 16/8/20 12:13 PM --%
  main
  clear all
  clc
  main
```

```
x =

     1


currentX =

     1


x =

     1

Iteration# 100 Swarm.GBEST.O = 1.7664
Optimal solution found

ans =

     1

Best objective value

ans =

    1.7664

Elapsed time is 1.368241 seconds.
>>
```

Fig 4 MATLAB Simulation for RFC

The optimal RFC solution is 1

Convergence curve shown during PSO implementation of RFC is show below:

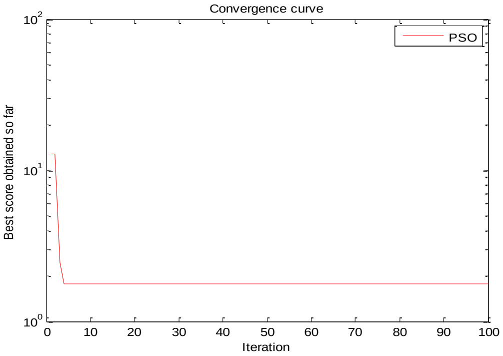
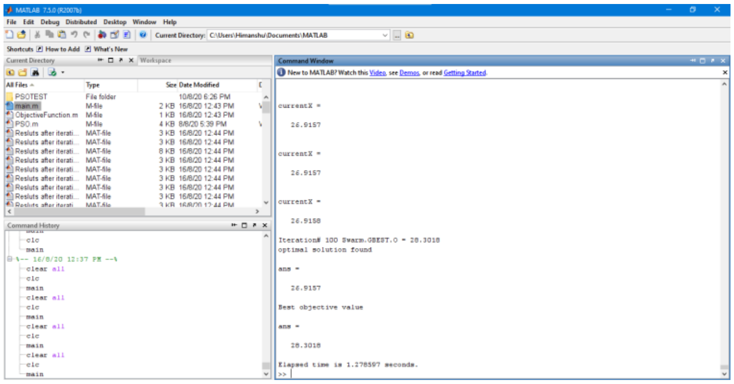


Fig 5 Convergence curve for RFC

RESULT OF PSO FOR NOC

After setting “main” command in MATLAB command prompt following result is shown below


MATLAB 7.5.0 (R2007b)
File Edit Debug Distributed Desktop Window Help
Current Directory: C:\Users\Himanshu\Documents\MATLAB
Shortcuts How to Add What's New
Current Directory
Workspace
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History

All Files	Type	Size	Date Modified
PSOTEST	File folder		10/8/20 6:26 PM
main.m	M-file	2 KB	16/8/20 12:43 PM
ObjectiveFunction.m	M-file	1 KB	16/8/20 12:43 PM
PSO.m	M-file	4 KB	8/8/20 5:39 PM
Resluts after iterati...	MAT-file	3 KB	16/8/20 12:44 PM
Resluts after iterati...	MAT-file	3 KB	16/8/20 12:44 PM
Resluts after iterati...	MAT-file	8 KB	16/8/20 12:44 PM
Resluts after iterati...	MAT-file	3 KB	16/8/20 12:44 PM
Resluts after iterati...	MAT-file	3 KB	16/8/20 12:44 PM
Resluts after iterati...	MAT-file	3 KB	16/8/20 12:44 PM
Resluts after iterati...	MAT-file	3 KB	16/8/20 12:44 PM
Resluts after iterati...	MAT-file	3 KB	16/8/20 12:44 PM
Resluts after iterati	MAT-file	3 KB	16/8/20 12:44 PM

```
  clc
  main
%-- 16/8/20 12:37 PM --%
  clear all
  clc
  main
  clear all
  clc
  main
  clear all
  clc
  main
  clear all
  clc
  main
```

```
currentX =

   26.9157


currentX =

   26.9157


currentX =

   26.9158

Iteration# 100 Swarm.GBEST.O = 28.3018
optimal solution found

ans =

   26.9157

Best objective value

ans =

   28.3018

Elapsed time is 1.278597 seconds.
>>
```

Fig 6 MATLAB Simulation for NOC

The optimal NOC solution is 26.9157

Convergence curve during PSO implementation of NOC is show below:

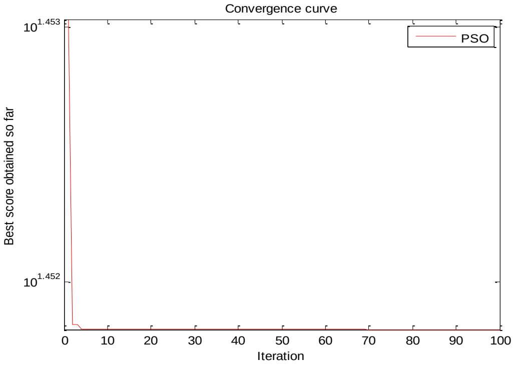
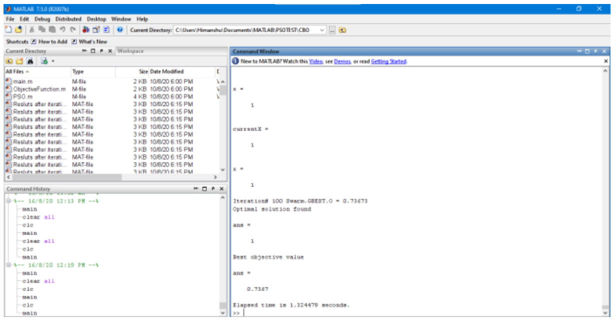


Fig 7 Convergence curve for NOC

RESULT OF PSO FOR CBO

After setting “main” command in MATLAB command prompt following result is shown below


MATLAB 7.5.0 (R2007b)
File Edit Debug Distributed Desktop Window Help
Current Directory: C:\Users\Himanshu\Documents\MATLAB\PSOTEST\CBO
Shortcuts How to Add What's New
Current Directory
Workspace
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History

```
All Files              Type      Size Date Modified
main.m                 M-file    2 KB 10/8/20 6:00 PM
ObjectiveFunction.m    M-file    2 KB 10/8/20 6:00 PM
PSO.m                  M-file    4 KB 10/8/20 6:00 PM
Resluts after iterati... MAT-file 3 KB 10/8/20 6:15 PM
Resluts after iterati... MAT-file 3 KB 10/8/20 6:15 PM
Resluts after iterati... MAT-file 3 KB 10/8/20 6:15 PM
Resluts after iterati... MAT-file 3 KB 10/8/20 6:15 PM
Resluts after iterati... MAT-file 3 KB 10/8/20 6:15 PM
Resluts after iterati... MAT-file 3 KB 10/8/20 6:15 PM
Resluts after iterati... MAT-file 3 KB 10/8/20 6:15 PM
Resluts after iterati... MAT-file 3 KB 10/8/20 6:15 PM
Resluts after iterati... MAT-file 3 KB 10/8/20 6:15 PM
```

```
%-- 16/8/20 12:13 PM --%
   main
   clear all
   clc
   main
   clear all
   clc
   main
%-- 16/8/20 12:19 PM --%
   main
   clear all
   clc
   main
   clc
   main
```

```
x =

     1


currentX =

     1


x =

     1

Iteration# 100 Swarm.GBEST.O = 0.73673
Optimal solution found

ans =

     1

Best objective value

ans =

    0.7367

Elapsed time is 1.324479 seconds.
>>
```

Fig 8 MATLAB Simulation for CBO

The optimal CBO solution is 1

Convergence curve shown during PSO implementation of CBO is show below:

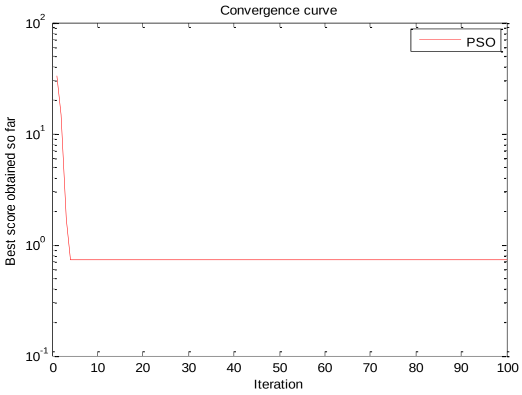
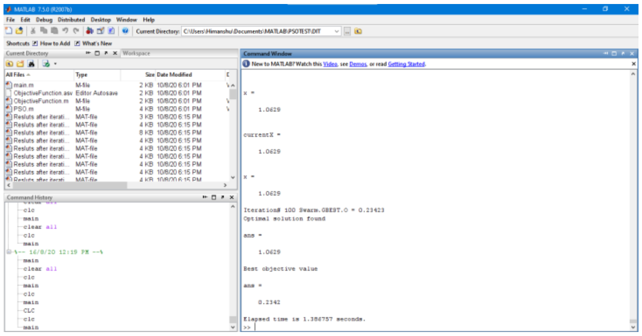


Fig 9 Convergence curve for CBO

RESULT OF PSO FOR DIT

After setting “main” command in MATLAB command prompt following result is shown below


MATLAB 7.5.0 (R2007b)
File Edit Debug Distributed Desktop Window Help
Current Directory: C:\Users\Himanshu\Documents\MATLAB\PSOTEST\DIT
Shortcuts How to Add What's New
Current Directory
Workspace
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History

```
All Files              Type            Size Date Modified
main.m                 M-file          2 KB 10/8/20 6:01 PM
ObjectiveFunction.asv  Editor Autosave 2 KB 10/8/20 6:01 PM
ObjectiveFunction.m    M-file          2 KB 10/8/20 6:01 PM
PSO.m                  M-file          4 KB 10/8/20 6:01 PM
Resluts after iterati... MAT-file      3 KB 10/8/20 6:15 PM
Resluts after iterati... MAT-file      4 KB 10/8/20 6:15 PM
Resluts after iterati... MAT-file      8 KB 10/8/20 6:15 PM
Resluts after iterati... MAT-file      4 KB 10/8/20 6:15 PM
Resluts after iterati... MAT-file      4 KB 10/8/20 6:15 PM
Resluts after iterati... MAT-file      4 KB 10/8/20 6:15 PM
Resluts after iterati... MAT-file      4 KB 10/8/20 6:15 PM
Resluts after iterati... MAT-file      4 KB 10/8/20 6:15 PM
```

```
   clc
   main
   clear all
   clc
   main
%-- 16/8/20 12:19 PM --%
   main
   clear all
   clc
   main
   clc
   main
   CLC
   clc
   main
```

```
x =

    1.0629


currentX =

    1.0629


x =

    1.0629

Iteration# 100 Swarm.GBEST.O = 0.23423
Optimal solution found

ans =

    1.0629

Best objective value

ans =

    0.2342

Elapsed time is 1.386757 seconds.
>>
```

Fig 10 MATLAB Simulation for DIT

The optimal DIT solution is 1.0629

Convergence curve shown during PSO implementation of DIT is show below:

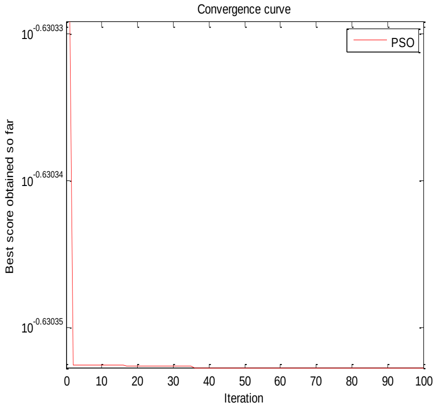
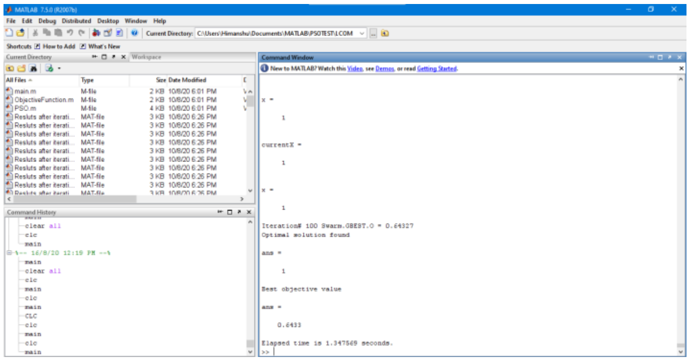


Fig 11 Convergence curve for DIT

RESULT OF PSO FOR LCOM

After setting "main" command in MATLAB command prompt following result is shown below



Current Directory: C:\Users\Himanshu\Documents\MATLAB\PSOTEST\LCOM

All Files	Type	Size	Date Modified
main.m	M-file	2 KB	10/8/20 6:01 PM
ObjectiveFunction.m	M-file	2 KB	10/8/20 6:01 PM
PSO.m	M-file	4 KB	10/8/20 6:01 PM
Resluts after iterati...	MAT-file	3 KB	10/8/20 6:26 PM
Resluts after iterati...	MAT-file	3 KB	10/8/20 6:26 PM
Resluts after iterati...	MAT-file	3 KB	10/8/20 6:26 PM
Resluts after iterati...	MAT-file	3 KB	10/8/20 6:26 PM
Resluts after iterati...	MAT-file	3 KB	10/8/20 6:26 PM
Resluts after iterati...	MAT-file	3 KB	10/8/20 6:26 PM
Resluts after iterati...	MAT-file	3 KB	10/8/20 6:26 PM
Resluts after iterati...	MAT-file	3 KB	10/8/20 6:26 PM
Resluts after iterati...	MAT-file	3 KB	10/8/20 6:26 PM

```
-clear all
-clc
-main
%-- 16/8/20 12:19 PM --%
-main
-clear all
-clc
-main
-clc
-main
-CLC
-clc
-main
-clc
-main
```

```
x =

     1


currentX =

     1


x =

     1

Iteration# 100 Swarm.GBEST.O = 0.64327
Optimal solution found

ans =

     1

Best objective value

ans =

    0.6433

Elapsed time is 1.347569 seconds.
>>
```

Fig 12 MATLAB Simulation for LCOM

The optimal LCOM solution is 1

Convergence curve shown during PSO implementation of LCOM is show below:

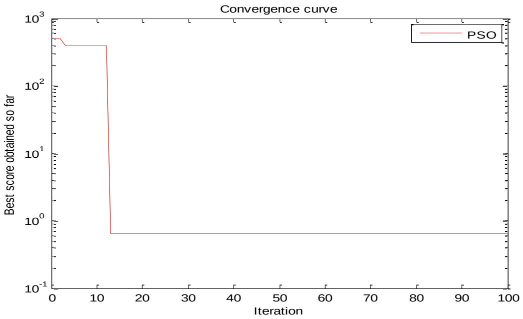


Fig 13 Convergence curve for LCOM

COMPARATIVE ANALYSIS

The comparative analysis of proposed work to previous researches considering various factors such as test case selection, maintenance effort estimation, mechanism used, use of optimization, Weight age considered, influencing factor for maintenance effort estimation and whether approach is object orient or not is shown in Table 2.

Table 2 Comparative analysis of proposed work with existing works

	An effective test case prioritization method based on fault severity[34]	**Assessment of Object Oriented Metrics for Software Reliability[36]**	**Test Case Prioritization Technique For Object Oriented Software Using Method Complexity [37]**	**Predicting Maintainability of Object-oriented Software Using Metric Threshold [38]**	**Proposed work**
Objective	Test case prioritization	Assessment of OOP metrics for software reliability	Test case prioritization	Predicting maintainability	Maintenance effort estimation
Test case selection	Yes	No	Yes	No	Yes
Test case prioritization	Yes	No	Yes	No	Yes
Mechanism	Random selection considering fault severities	Computation of CK metrics	Path prioritization value is calculated by method complexity	Computation of CK metrics	PSO, Computation of CK metrics ,weight age
Optimization	No	No	No	No	Yes
Weight age considered during prioritization	No	Yes	Yes	No	Considered
Factors considered	test costs, faults severities, faults types	WMC, DIT, NOC	Method complexity	DIT, CBO, WMC, LCOM, NOC, RFC.	DIT, CBO, WMC, LCOM, NOC, RFC.
Object oriented approach	No	Yes	Yes	Yes	Yes

5. CONCLUSION

Research has enhanced complexity based approach for effort estimation by applying PSO to obtain optimal value of weight. Complexity of object oriented metrics namely WMC, DIT, NOC, LCOM, RFC, CBO would be utilized to obtain ranks for various modules. Proposed work provided a solution for effort estimation. Moreover, proposed work has reduced the possibility of same rank using weight mechanism. The total weight of all C& K parameter has been considered and ranking is made accordingly.

6. FUTURE SCOPE

Proposed research could play significant role in effort estimation in object oriented program. OOPS is frequently used in software development. Thus, there is increase in demand of suitable and efficient mechanism for prioritization during effort estimation. The modules of object oriented programming such as Java, C++, C# could be prioritizing with support of C&K Metrics (DIT, CBO, WMC, LCOM, NOC, RFC). This research is capable to play significant role to distinguish the rank of cases considering sum of weight which module were showing same rank.

REFERENCES

[1] Mohamed Abdullahi Ali, Ng Keng Yap, “Software Component Quality Model”, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958,Volume-9, Issue-1, October 2019

[2] Kaur, Arvinder, Kamaldeep Kaur, and Ruchika Malhotra. "Soft computing approaches for prediction of software maintenance effort." International Journal of Computer Applications 1, no. 16 (2010): 69-75.

[3] Rani, Reenu, Dipen Saini, and Amandeep Kaur Cheema. "OPEN SOURCE SOFTWARE MAINTENANCE COST EVALUATION USING MAINTAINABILITY INDEX AND CODE METRICS."

[4] Punia, Mamta, and Amandeep Kaur. "Software maintainability prediction using soft computing techniques." IJISET 1, no. 9 (2014): 431-442.

[5] Narang, Kiran, Puneet Goswami, and K. Ram Kumar. "Maintainability Configuration for Component-Based Systems Using Fuzzy Approach." In Computational Methods and Data Engineering, pp. 243-258. Springer, Singapore, 2021.

[6] Edson, Bruce, Bill Hansen, and Pat Larter. "Software reliability, availability, and maintainability engineering system (SOFT-RAMES)." In Proceedings of 1996 Annual Reliability and Maintainability Symposium, pp. 306-311. IEEE, 1996.

[7] Singh, Charu, Amrendra Pratap, and Abhishek Singhal. "An estimation of software reusability using fuzzy logic technique." In 2014 International Conference on Signal Propagation and Computer Technology (ICSPCT 2014), pp. 250-256. IEEE, 2014.