# Streamlining Object Oriented Programming Approach based on Cuckoo Search Back Propagation Neural Network

V. Mallaiah
Research Scholar
Dept. of CSE
Acharya Nagarjuna University
Guntur

A. Vinaya Babu
Professor
Dept. of CSE
Stanley College of Engineering &
Technology for Women,
Hyderabad

S. Nagaprasad, PhD
Faculty of Comp. Sci.
Dept.of CS & CA
Tara Govt. College
Sangareddy

## ABSTRACT
Software reuse is helps to maintain the software and reduce much time in delivering, and cost and also improve quality. Most organization is considering software reuse is the crucial task and lot of algorithm is used for maintaining the software. The efficient management of monitoring data is essential for many software industries. The evolutionary test generation's development in the recent years helps to test most of Object Oriented Program (OOP). In this method, Cuckoo Search Back Propagation Neural Network (CSBP) facilitates Class Responsibility Assignment in the software. The Cinema Booking System (CBS) is used to evaluate the function of the proposed system and existing system. Cuckoo Search (CS) algorithm find best solution for the class arrangement and Back Propagation Neural Network (BPNN) analyze best solution in backward direction. The proposed method gives better result than the existing technique used in the OOP for software reuse. The parameters evaluated from the techniques are Cohesion, Complexity, Cost function and Coupling. The cohesion value is the measure of element in the software belongs together and value of cohesion in proposed method is achieved up to 0.5862.

## Keywords
Cinema Booking System, Cohesion, Coupling, Cuckoo Search Back Propagation Neural Network, Object Oriented Program.

## 1. INTRODUCTION
The software modularization quality significantly affects quality of the software system, such as understandability and maintainability [1]. For Object Oriented Programming (OOP), the modularization is largely dependent on classes, which consist collection of data and associated methods [2]. In the smaller software system, classes are considering as a module and for larger software system, a set of collaborative classes is considering as better module for organizing system than a class [3]. OOP creates test cases automatically, which helps to have specified coverage criterion includes branch coverage for software under test (SUT). Random testing (RT) and Evolutionary testing (ET) are existing system provide considerable method in developing workable test generators for OOP [4]. This provide little focus for test generation for SUT with special Object oriented features, such as Inherited class hierarchies (ICH) and Non-public methods (NPM) [5]. ICH is collaborative classes that arrange classes with help of inheritance mechanism, whereas NPM is visible only in some classes due to encapsulation mechanism.

The software can be control, measured, evaluated, and improved with help of software metrics and lot of research has been carried out in software metrics [6]. The software design is one of main process, which is conducted before software development [7]. The internal structure and behavior of software are presented in this step and there are some problems in software design, each of them has its own activity [8]. The assigning attributes to class is crucial task in OOP and automated method is required for effective arrangement of classes. IT aims to identify method for best class assignment. There are numerous researches is done on class responsibility assignment with different metaheuristic algorithm. Aim of research is to design method for assigning responsibilities of classes in software. The CS back propagation neural network based algorithm for solving problem in class responsibilities arrangements. The experiment is carried with and without Formal Concepts Analysis (FCA) and compared it with state-of-art method. The software metrics such as cohesion and coupling is evaluated from proposed technique [9], [10]. Estimated result shows that proposed method delivers better outcome than other state-of-art method.

## 2. LITERATURE REVIEW
The latest research papers are reviewed in this section for understanding current techniques in the object oriented software. OOP system has been applied in the various programs for the maintainability.

H. Wei, et al. [11] gives the method to generate test cases for the OOP using integrate evolutionary test with the reinforcement learning. The ICH and NPM were two process of isomorphism substitution are processed and Q-value matrix was maintained to help the evolutionary test generation. The prototype was designed using this technique, which was named as EvoQ and it was applied to the original Java programs. This method was conducted and result showed that it has the higher branch coverage than the two state-of-art method within the same time budget. The EvoQ with distinct parameter of SUT helps to improve the performance of the method.

Ferreira, et al. [12] aim to identify the thresholds values for the various OOP metrics like number of public of fields, DIT, number of public methods, afferent couplings and LCOM. The various number of JAVA software available in the open source, which are presented in different size and applications are taken for this study. The proposed threshold is used to analyze the software quality, which shows proposed techniques helps to identify classes affects design principle and also well-defined classes. The threshold values can be used to measure the quality of the software in order to find the reference values. These values are not to be ensured that the best interest of the common practice.

Christopher L. Simons and Ian C. Parmee, et al. [13] proposed the four novel elegance measures to investigate the contribution of elegance and symmetry in software design. This technique was based on the evenness of distribution of design elements and the designers are provided the visualizations of object-oriented software designs in the controlled IEC environment, which they order according to the elegance. The reward has been elicited for the three out of four proposed measure, which are all shows the significant correlation between the elegance values. The proposed measures were proved to have the symmetrical elegance was in some way significant in software design and this can be applied in dynamic and multi-objective IEC to have elegant software design. The output of this technique are attained are highly depend on the design problem used.

Wang, et al. [14] identify the hierarchical relationships among the monitoring objects in the seven categories of property objects, two objects that indicates and manage the construction progress, are proposed to meet a requirement for geotechnical monitoring, data flow and the monitoring objectives of the site. The object oriented method in Visual C++ environment was used to provide the corresponding data structure and database. The different types of information and document management schemes are integrated using the software. Examples of document management schemes are CAD drawing visualization, data input and processing, data modeling and prediction and also an early warning function. The evaluation gives that the software is highly flexible and reliable, which can be used for the various applications. The structural computation and numerical analysis module is needed to measure in the software to design a computer-aided design system.

Kakarontzas, et al. [15] establish new metric for reuse of object oriented software, which is based on popular existing suite. The new metrics was derived using linear regression applied on the different OOS java projects. The new metrics was compared with the other metrics and reuse of the class is important for the future use of software. The aim of the proposed method is to help the software developer to achieve reusability of classes and also to assist in re-architecting and componentization activities of existing system. The coding standards followed are not considered in the research, which is also important for the reusability of white-box.

The proposed method is used to solve the problems mentioned in the literature of these researches. The proposed system is discussed details in the next section.

## 3. PROPOSED METHOD

In OOP, objects are collaborated closely to provide system function and for better performance of the object oriented developer should understand service of the class. In this context, meta-heuristic algorithm of CS back propagation neural network is proposed for OOP. The document has been analyzed by the algorithm in manner of use cases, attributes and methods as input and produces a class diagram of the system as output. The method is separated into three major categorizes, Formal Concepts Analysis (FCA) is applied and proposed algorithm is used for optimization of initial raw assignments, then setting relationships between classes. The schematic diagram of this method is shown in Fig (1).
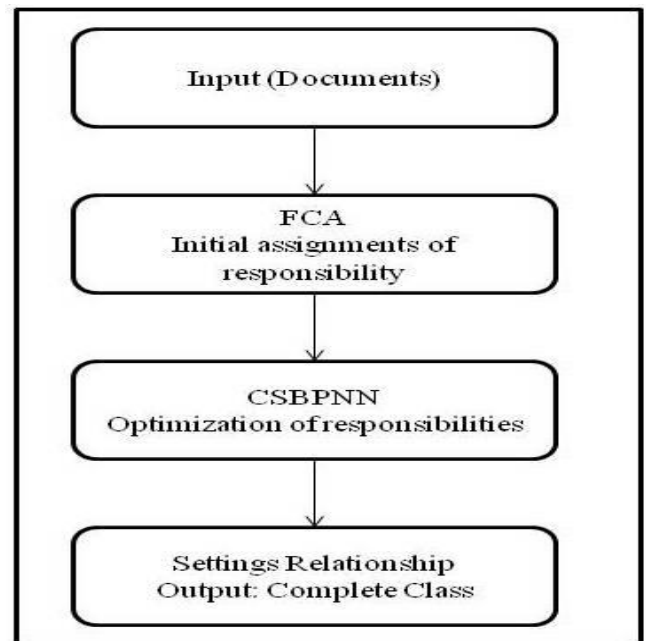


**Figure 1 Overview of the method**

## 3.1. Formal Concept Analysis:

FCA is the method of data analysist, which has increasing popularity in the various processes. FCA analyzes the data and provides relationship between particular set of object and particular set of attributes. The relationships help in OOP for arrangement of class respiratory. The output of the FCA is in the form of concept lattice. A concept lattice is a collection of formal concepts in data which are in hierarchically ordered by a sub concept to super concept relation. Formal concepts are particular cluster. The Cinema Booking System (CBS) is used for evaluation of this method in OOP function with other methods. FCA provides model of CBS as conceptual relationships using lattice diagram. In FCA, formal things and features are used and relationships between the classes are created. The context for CBS case study depicted in candidate classes shown in Fig. (2). each row is a use case and each column is a responsibility in the system. The lattice diagram for this context is displayed in Fig (3). Each node in this lattice is considered as the concepts or candidate classes.

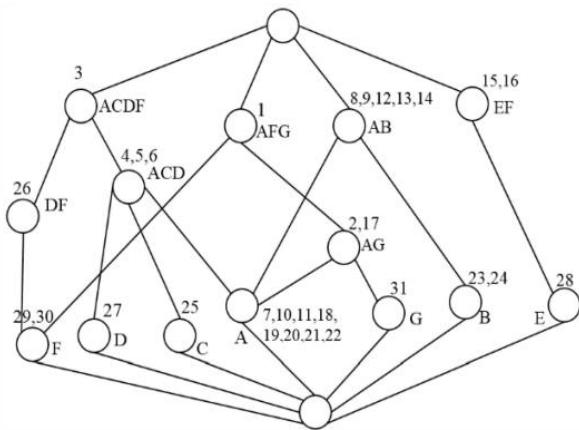**Figure 2 Context diagram of CBS case study**



**Figure 3 Concept lattice of CBS**

The initial assignment is created with help of a domain expert and it is used as input for the optimization algorithm. In the next process, optimization algorithm tries to optimize the initial solutions using CSBPNN.

## 3.2.  Back propagation Neural Network:

Rumelhart, Hinton and Williams introduce the Back propagation Neural Network (BPNN) in 1986, which is one of the most popular supervised learning algorithms. The error of output layer is used by the BPNN for calculating error of hidden layers. It can be applied to problems which has no relationship between output and input layer due to the back propagation ability. The weight and adjustment are calculated using gradient descent method is made to the network to reduce the error in output. This is one of standard algorithm for training multilayer perception and criterion is minimized by least mean squared equal to the sum of the squares of errors between actual and desired outputs. This principle is:

$$E_p = \sum_{i=1}^{j}(e_i)^2 \qquad (1)$$

Where the nonlinear error signal is

$$e_i = d_i - y_i \qquad (2)$$

The desire and the current outputs for the $i^{th}$ unit is $d_i$ and $y_i$ respectively. j is the number of output units and P denotes in Eq. (1) the $p^{th}$ pattern. The gradient descent method is displayed in the Eq. (3).

$$w_{ki} = -\mu \frac{\partial E_p}{\partial w_{ki}} \qquad (3)$$

Where $w_{ki}$ is the weight of the $i^{th}$ unit in the $(n-1)^{th}$ layer to the $k^{th}$ unit in the $n^{th}$ layer. The BP calculates errors in the output layer $\partial_l$, and the hidden layer, $\partial_j$ are using the formulas in Eq. (4) and Eq. (5) respectively:

$$\partial_l = \mu(d_i - y_i)f'(y_i) \quad (4)$$

$$\partial_j = \mu \sum_i \partial_l w_{lj} f'(y_i) \quad (5)$$

Here $d_i$ is the desired output of the $i^{th}$ output neuron, $y_i$ is the actual output in the output layer, $y_i$ is the actual output value in the hidden layer, and k is the adjustable variable in the activation function. The back propagation error is used to update weight and biases in both output and hidden layers. The weights, $w_{ij}$ and biases, $b_i$, are then adjusted using the following formulae;

$$w_{ij}(k + 1) = w_{ij}(k) + \mu\partial_j y_i \ (6)$$

$$w_{lj}(k + 1) = w_{lj}(k) + \mu\partial_j x_l \ (7)$$

$$b_i(k + 1) = b_i(k) + \mu\partial_j \ (8)$$

Where k is the number of epoch and $\mu$ is the learning rate. The BPNN is combined with the CS algorithm for the optimization solution of class representation in the software. The CS algorithm is illustrated below.

## 3.3.Cuckoo Search Algorithm

Cuckoo Search (CS) algorithm is inspired from obligate brood parasitism of some cuckoo species by laying their eggs on other host birds. Some host nest has direct difference and in that case, the egg is found by the host birds and it will throw unrecognizable egg.  Some species are evolved in the way such that female cuckoos are very specialized in mimic the egg similar in color and pattern of the eggs of few chosen host species. This helps to reduce the probability of the eggs being thrown away and increase their population. The algorithm of CS is follows three idealized rules.

  a.  Each cuckoo lays one egg at a time and chose the random nest to place its egg.

  b.  The nest with high quality of eggs is the best nest, which carried to next generation.

  c.  The total nest available is fixed and the probability for the host bird to discover the cuckoo egg is given by $pa \in [0,1]$.

If the egg is found by the host bird, it will throw away or abandon the egg and construct the complete new nest. The replacement of new nest is approximated by the fraction of pa $\in$ [0,1] given in the rule-c. The new random solution is calculated from the new and for maximization problem, the quality or fitness of a solution can simply be proportional to the value of the objective function. The solution is provided by each egg in the nest and cuckoo egg  provide new solution, the best solution is replaced not so good solution in the nest. The three rules of the CS algorithm are summarized in the pseudo code given below. The CS is combined with BPNN, which is explained below.

The pseudo code for the CS algorithm is given below:

Step 1: Generate initial population of N host nest $i = 1 ..... N$

Step 2: While ($f_{min} < MaxGeneration$) or ($stop\ criterion$).

Do

Step 3: Get a cuckoo randomly and evaluate it fitness $F_i$

Step 4: Choose randomly a nest $j$ among $N$.

Step 5: if $F_i > F_j$ Then

Step 6: Replace $j$ by the new solution

End if

Step 7: A fraction $pa$ of worse nest are abandoned and new ones are built.

Step 8: Keep the best solutions (or nest with quality solutions).

Step 9: Rank the solutions and find the current best.

End while.

## 3.4. Cuckoo Search Back Propagation Neural Network

The CS algorithm is one of the population based algorithm and it starts with a random initial population like other meta-heuristic algorithms. The three steps of CS algorithm are providing better solution by keeping best solution, replace them with respect to the quality of the cuckoo egg or solution generated randomly and finding of cuckoo egg by the host bird and replacing them with quality of random walks. The various processes are present in each cycle such as initialization of the best nest or solution, host nest count is fixed constant and egg laid by the cuckoo is discovered by the host bird in the probability of pa ∈ [0,1].

Each best nest in the proposed CSBP algorithm represents best possible solution. The weight optimization problem is used to calculate the quality of solution and population size indicates solution quantity. The weight and bias are fixed in the first epoch and CS finds the best and the weight value is passed to BPNN. In BPNN, the weights are measured and best solution given by CS is analyzed in the backward direction. In the next cycle, the best solution is updated and searching is done for best solution until the last cycle or epoch of the network is attained or either MSE is achieved.

The pseudo code of the proposed CSBP algorithm is displayed below.

Step 1: Initialize CS and pass the best weight to BPNN

Step 2: Load the training data

Step 3: While $MSE < stopping\ criteria$

Step 4: Initialize all cuckoo nests

Step 5: Pass the cuckoo nests as weight to the network

Step 6: Feed forward neural network runs using the CS weight initialized

Step 7: Calculate the error backward

Step 8: CS keeps on calculating the best possible weight at each epoch until the network is converged.

The CSBP is used for the representing the classes in OOP with the better solution obtained from CS and tested in the backward direction using BPNN. This method is evaluated using the dataset of the CBS software and result is compared with state-of-art method in the next section. The final output of the class diagram is also given in the experimental result section.

## 3.5. Setting Relationships

A class diagram consists of some classes and their relationship and types of relationship between classes are generalization, association, abstract/realization and dependency. A dependency is the type of relationship between two elements of design model and any change in one element affects another element. The association relationship is a connection between classes and Bowman et al. [16] technique is used for set associative relationship between the element. The method proposed by Masoud et al. [17], which implements the inheritance for generalization.

The element relationships are set to construct the final class diagram of CBS, which is shown in the result section. The performance of the proposed is measured in the manner of cohesion, coupling, complexity, cost function and Class Size Standard Deviation (CSSD).

## 4. EXPERIMENTAL STUDY

The case study is discussed and also parameter settings are given in this section. The parameter measurements are displayed with equation. The evaluation of the proposed system from these settings of case study and BPNN.

## 4.1. Case Studies and parameter Evaluation

The CBS are used as case study from Javidi, et al. [18] used for the evaluation of proposed method. The brief description of parameters of CBS is displayed in Table. (1). The FCA is processed in CBS and fed into CSBP algorithm. The Netbeans Java tool in the computer consists of 3.0 GHz processor and 4 GB of RAM with windows 10 OS, is used to function this method with CSBP in OOP.

**Table 1 CBS Parameter settings**

| No. of | Attributes | Methods | Responsibilities | Classes | Usage |
|---|---|---|---|---|---|
| Cinema Booking System (CBS) | 16 | 14 | 31 | 5 | 39 |

The performance of the proposed method is compared with the existing method. The CS algorithm is combined with BPNN to find the best solution in backward direction. The generation number for GA and country number for ICA is intended 100 and size of Tabu list in TS is set 0.4 of all possible action. Weights for metric value given in relation 2 are set as follows and values of these weights are obtained experimentally.

$$w_1 = w_2 = w_4 = 0.2 \; and \; w_3 = 0.4$$

The parameters assigned for the BPNN is given in the Table. (2). this provides the layers, epoch and learning rate of the network. The different neurons are set for various layers.

**Table 2 Parameter settings in BPNN**

| Model No. | No of neurons in three layers | Learning rate η | Moment um rate α | Gain | Epoc hes |
|-----------|------------------------------|-----------------|------------------|------|----------|
| BPNN | 6,19,2 | 0.1 | 0.4 | 1 | 368 |

## 4.2. Evaluation Metrics

The object-oriented metrics is an efficient way to evaluate the quality of an architectural design method and metrics are used for evaluate the software. There are three categories of object oriented metrics: coupling between classes, inner cohesion of the class and complexity of design. The proposed method is evaluated with help of three metrics. To calculate the cost function, weighted combination of four criteria is measured. The value of cohesion has to be maximum and other metrics has to be minimum for the better function of the system. Hence relation composed of combination of minimization metrics and reverse of maximization metric.

$$CostFunction \, (D) = w_1 Cop(D) + w_2 Complexity(D) + w_3 ClassSizeStandardDevication(D) + \frac{w_4}{Coh(D)}$$
(9)

Where:

$$\omega 1 + \omega 2 + \omega 3 + \omega 4 = (10)$$

The coupling, complexity of design, cohesion, and standard deviation of the class size denoted as Cop(D), Complexity(D), Coh (D) and Class Size Standard Deviation(D). The right choice of weights depends on design priorities and empirically metrics are taken. These parameters are evaluated from several methods and their output is shown in the experimental result section.

## 5. EXPERIMENTAL RESULT

The proposed method CSBP is process on the CBS for OOP and other existing method also evaluated in the program. The CS algorithm tries to find better solution for the system and BPNN analysis the best solution from backward direction. The metrics are calculated from function of the method. The Table. (3) gives the comparison of proposed method and other existing method. The proposed method has the higher cohesive value and lower in other three parameters. The expert in the OOP designs the class manually in the CBS. The class diagram from the expert and from proposed system is shown in the Fig (4) and (5).
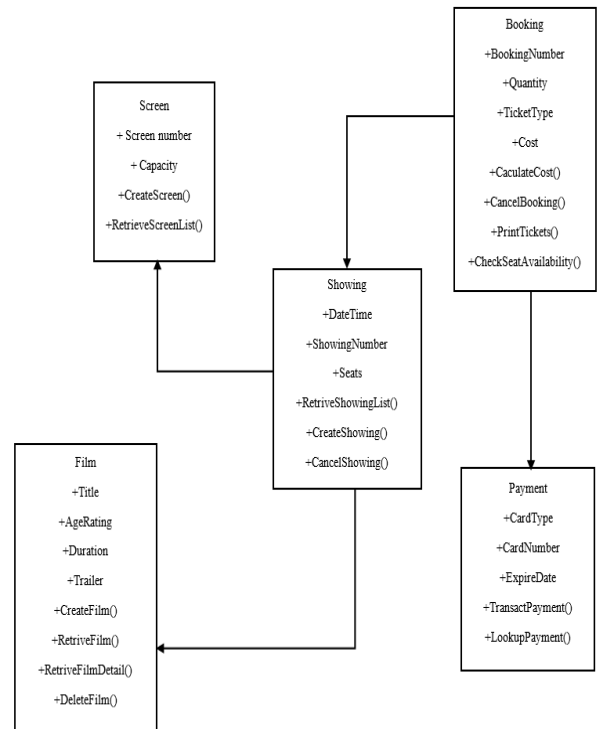
**Figure 4 Class diagram by experts**
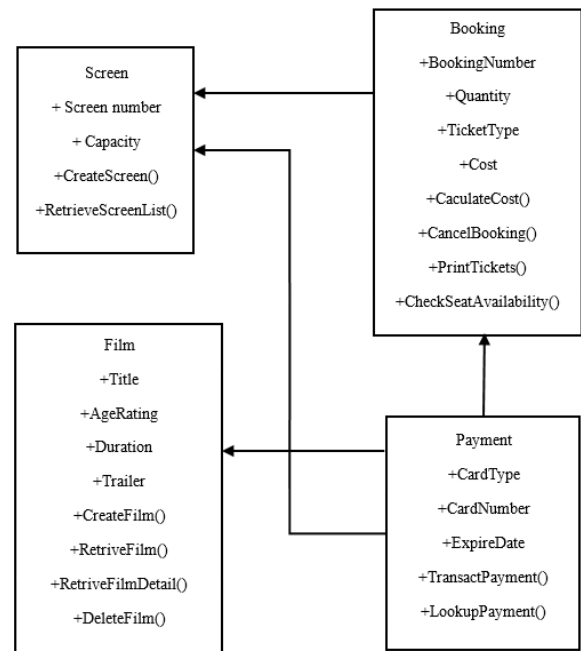
**Figure 5 Class diagram from proposed system**

**Table 3 Performance measurement**

| Metho d | Criteria | | | | |
|---------|---------|---|---|---|---|
| | | Cost functi on | Cohes ion | Coupl ing | Comple xity | CSS D |
| **GA** | Be | 0.497 | 0.550 | 0.315 | 0.29446 | 0.031 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **[18]** | st | | 45 | 79 | | 48 |
| | Avg | 0.526 | 0.53576 | 0.3512 | 0.30468 | 0.05453 |
| | St.D | 0.03 | 0.01618 | 0.03935 | 0.01602 | 0.04722 |
| **ICA [18]** | Best | 0.497 | 0.55045 | 0.2823 | 0.26221 | 0.03148 |
| | Avg | 0.512 | 0.5378 | 0.33481 | 0.29664 | 0.03422 |
| | St.D | 0.015 | 0.01291 | 0.03916 | 0.01587 | 0.00822 |
| **ICA-TS [18]** | Best | 0.497 | 0.55045 | 0.31579 | 0.29446 | 0.03148 |
| | Avg | 0.502 | 0.54827 | 0.32632 | 0.29774 | 0.03148 |
| | St.D | 0.012 | 0.00572 | 0.03158 | 0.00983 | 0 |
| **ICA-TS with FCA [18]** | Best | 0.469 | 0.54072 | 0.25 | 0.15292 | 0.03099 |
| | Avg | 0.473 | 0.5357 | 0.25849 | 0.17834 | 0.03099 |
| | St.D | 0.007 | 0.01288 | 0.00594 | 0.00934 | 0 |
| **BPNN** | Best | 0.442 | 0.5682 | 0.2156 | 0.12678 | 0.02972 |
| | Avg | 0.442 | 0.5522 | 0.2266 | 0.1426 | 0.03006 |
| | St.D | 0.061 | 0.014 | 0.00522 | 0.00913 | 0 |
| **CSBPNN** | **Best** | **0.050** | **0.8** | **0.34** | **0.967** | **2.214** |
| | **Avg** | **0.017** | **0.355** | **0.072** | **0.933** | **0.785** |
| | **St.D** | **0.0040** | **0.0535** | **0.0036** | **0.0014** | **0.1537** |
| **Expert Design [18]** | | 0.886 | 0.2787 | 0.33851 | 0.25631 | 0.12592 |



**Figure 6 Performance measure**

The graphical representation of the proposed and existing method for various parameters is shown in Fig. (6). The methods compared in the graph are Genetic Algorithm (GA), ICA-TS, Imperialist Competitive Algorithm Tabu Search (ICA-TS) with FCA, ICA, expert and BPNN. The various metrics are calculated and compared with best, average and standard deviation values. The cost function is very high for the expert design while compared to the other methods.



**Figure 7 Cohesion Parameter for several methods**

The performance of the various methods is measured in terms of cohesive and compared in the graph, which is shown in Fig. (7). the cohesive measure should be high for the better performance of OOP. The CSBP provides the higher cohesive value and BPNN has the second higher value.

The CSBPNN gives better performance than the existing system in terms of Cohesive, cost function, coupling, complexity and CSSD. The three types of values are shown in the table such as Best, average and standard deviation. The expert result is also given in the Table. (3), for various metrics. The CSBPNN gives the high performance and low performance is given by expert. The second best performance is provided by the BPNN.
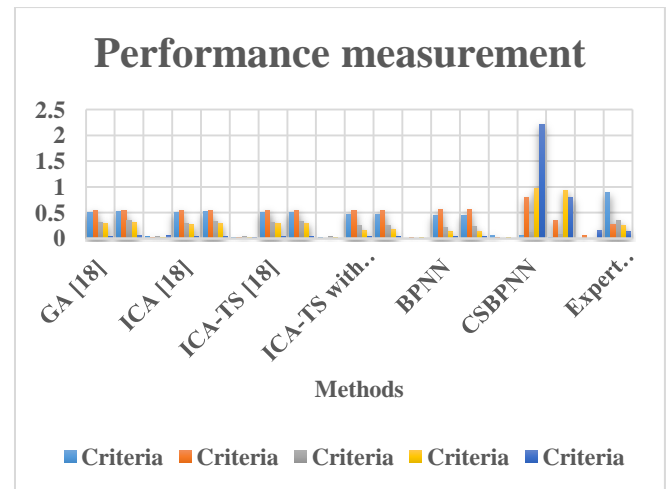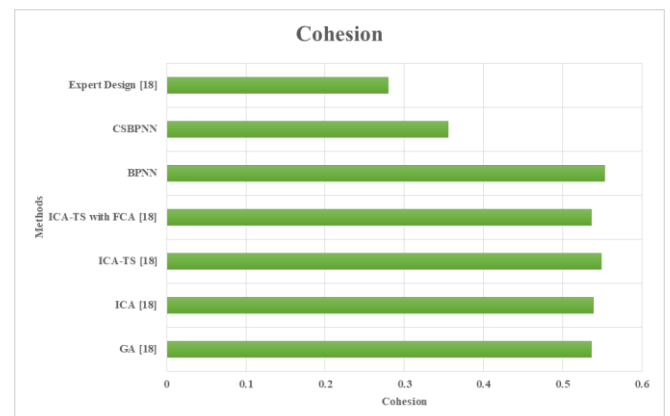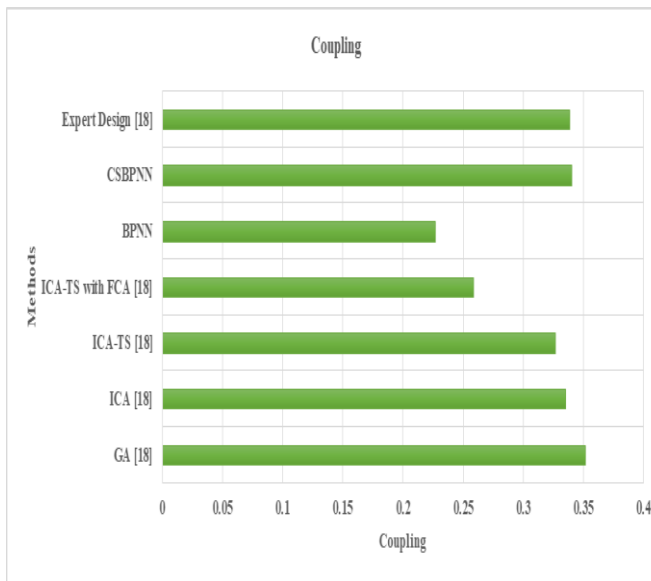
**Figure 8 Coupling measure for several methods**

The coupling value is calculated from the existing method and proposed method, then the compared in the graph, shown in Fig. (8). the lower the coupling values provides higher the performance of the system. The proposed system achieves the lower coupling values and expert have the higher coupling value.

This clearly shows that the proposed system attains higher performance than the existing system. The five metrics are calculated from the experiment and shown in the performance measure table. The CS algorithm finds the better solution for OOP and BPNN is used to replace the better solution by searching the algorithm in the backward. This proposed system can be applicable for the software maintains functions and also for the OOP method.

# 6. CONCLUSION

The software maintains is difficult task in the software industries and this increases the needs of object oriented techniques with better performance. A lot of existing software should have to maintain by the software industries and OOP techniques is the better way to assign the class in the software. The various algorithms is used in the OOP technique like GA, ACO and ICA, which solves the optimization problem. In this method, CSBPNN gathers the document as an input and FCA is applied in the input data. The FCA arranges the data in the lattice manner, which helps the proposed method to find the better solution for the class effectively. The CSBPNN is used in the data to find the better solution and class diagram is created. The CBS is used as input for the evaluation of the proposed method and existing method. The result shows that the proposed method shows high performance compared to other existing system. In the future work, the dynamic components are added to extent the function and also increase the test program.

# 7. REFERENCES

[1] Vazquez, G., Pace, J.A.D. and Campo, M., 2014. Reusing design experiences to materialize software architectures into object-oriented designs. Information Sciences, 259, pp.396-411.

[2] Malhotra, R. and Chug, A., 2014. Application of group method of data handling model for software maintainability prediction using object oriented systems. International Journal of System Assurance Engineering and Management, 5(2), pp.165-173.

[3] Araujo, W., Briand, L.C. and Labiche, Y., 2014. On the effectiveness of contracts as test oracles in the detection and diagnosis of functional faults in concurrent object-oriented software. IEEE Transactions on Software Engineering, 40(10), pp.971-992.

[4] Kaplanoğlu, V., 2016. An object-oriented approach for multi-objective flexible job-shop scheduling problem. Expert Systems with Applications, 45, pp.71-84.

[5] Al Dallal, J., 2017. Predicting Fault-Proneness of Reused Object-Oriented Classes in Software Post-Releases. Arabian Journal for Science and Engineering, pp.1-14.

[6] de AG Saraiva, J., De França, M.S., Soares, S.C., Fernando Filho, J.C.L. and de Souza, R.M., 2015. Classifying metrics for assessing object-oriented software maintainability: A family of metrics' catalogs. Journal of Systems and Software, 103, pp.85-101.

[7] Simons, C.L. and Parmee, I.C., 2012. Elegant object-oriented software design via interactive, evolutionary computation. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(6), pp.1797-1805.

[8] Malekan, M., Silva, L.L., Barros, F.B., Pitangueira, R.L. and Penna, S.S., 2018. Two-dimensional fracture modeling with the generalized/extended finite element method: An object-oriented programming approach. Advances in Engineering Software, 115, pp.168-193.

[9] Bidve, V.S. and Sarasu, P., 2016. Coupling Measures and its Impact on Object-Oriented Software Quality. Indian Journal of Science and Technology, 9(21).

[10] Kartha, G.P., Anjali, C., Nair, R.V. and Venkateswari, S., 2017, July. Prediction of defect susceptibility in object Oriented Software. In Networks & Advances in Computational Technologies (NetACT), 2017 International Conference on(pp. 467-472). IEEE.

[11] He, W., Zhao, R. and Zhu, Q., 2015. Integrating evolutionary testing with reinforcement learning for automated test generation of object-oriented software. Chinese Journal of Electronics, 24(1), pp.38-45.

[12] Ferreira, K.A., Bigonha, M.A., Bigonha, R.S., Mendes, L.F. and Almeida, H.C., 2012. Identifying thresholds for object-oriented software metrics. Journal of Systems and Software, 85(2), pp.244-257.

[13] Chhabra, J.K., 2017. Improving package structure of object-oriented software using multi-objective optimization and weighted class connections. Journal of King Saud University-Computer and Information Sciences, 29(3), pp.349-364.

[14] Wang, H., Li, L., Jiao, Y.Y., Ge, X.R. and Li, S.C., 2014. A relationship-based and object-oriented software for monitoring management during geotechnical

excavation. Advances in engineering software, 71, pp.34-45.

[15] Kakarontzas, G., Constantinou, E., Ampatzoglou, A. and Stamelos, I., 2013. Layer assessment of object-oriented software: A metric facilitating white-box reuse. Journal of Systems and Software, 86(2), pp.349-366.

[16] Bowman, M., Briand, L.C. and Labiche, Y., 2010. Solving the class responsibility assignment problem in object-oriented analysis with multi-objective genetic algorithms. IEEE Transactions on Software Engineering, 36(6), pp.817-837.

[17] Masoud, H. and Jalili, S., 2014. A clustering-based model for class responsibility assignment problem in object-oriented analysis. Journal of Systems and Software, 93, pp.110-131.

[18] Javidi, Z., Akbari, R. and Bushehrian, O., 2017, March. Semi-automatic object-oriented software design using metaheuristic algorithms. In Swarm Intelligence and Evolutionary Computation (CSIEC), 2017 2nd Conference on (pp. 123-128). IEEE.