SOFTWARE REUSE COMPONENT BASED SOFTWARE REPOSITORY SYSTEM USING GENETIC ALGORITHM

G. Maheswari1, Dr.K. Chitra2

ABSTRACT: *-Software reuse is significantly playing role to improve the qualityof software component, reliability and reduces the overall development. The main objective is to find the optimal component and define the evaluation criteria forthe reusable software component from the software repository. In existing systemof the software developmentthe importantproblem istime and cost for thecomponent selection is too low and very difficult to selectthe best matching component fromthe component repository. The component basedsoftwareis an areathatintegrates technicallike data mining, soft computing etc.Themain areasthat focused in software reuse are classification, clustering,searching, indexing and retrieval of softwarecomponents. This paper isfocusedongenetic algorithm-based search and retrieval ofsoftwarecomponents. The genetic algorithm is one of the optimization techniques. In this technique retrieval of component should be less time consuming and efficient. It is also tool forsoftware development and maintenance tasks. The WEKA tool isuse to measures the fitness valueof recall and precision for components usinggenetic algorithm and output is displays the relevant component for the proposedmethodologies.*

Keywords: *Software reuse, Genetic Algorithm, Fitness Evaluation, Software repository, Optimization techniques.*

1. INTRODUCTION

Reuse of programming is an improvement of reusability the portion of the item structures. The reusability section can have better qualified, low costly and upgraded execution. Segment put together programming reusability is the premise with respect to the assessing, choosing, planning different programming segments. The part store is the most ideal testing taskthat tends to the necessity with the least advancement time and cost. The principal issues in the segment-based framework which programming segment is chosen and sort of check required and sort of prior programming segments.

The product advancement procedure is chosen for the assessment and determination of the best ideal parts. The primary advantage of part based programming designing is to get parallel and dispersed advancement just as increment reusability at a lower cost with expanding adaptability. The various standards of part based improvement which impacts the advancement procedures and support that required significant change to achieve standard qualities.

The fundamental issues of programming reusability are the majority of associations is the capacity to find and recover the current programming segments. So as to beat this issue, a vital advance is the capacity to arrange and accumulations of programming segments, to rapidly look through a gathering to recognize contender for potential reuse.

The best quality reuse vault instrument is required to have a wide assortment of top quality parts and it is a productive

[a] *Part-Time Research Scholar, Department of Computer Science, Madurai Kamraj University, Madurai, Tamilnadu, India. Mail_id:-gm291276@gmail.com*

[b] *Assistant Professor, Department of Computer Science, Government Arts College, Melur, Madurai, Tamilnadu, India Mail_id : - manikandan.chitra@gmail.com*

Received: 10 Jul 2020| Revised: 20 Aug 2020| Accepted: 02 Sep 2020

way utilizing an arrangement system and ready to recover the best segments. This paper center around the new system is a recovery of programming segments from the reuse archive and it executes a hereditary calculation for the viable order of segments in the storehouse and recovers the best-fit qualities dependent on the necessities.

To create segment-based programming, engineers are confronting a test to discover programming parts among the segments that are recently assembled. The quantity of segment develops that the intricacy of segments winds up more noteworthy. The administration for the current segments is essentially required.

The segment store is making helpful access to reusable parts and gives reuse usefulness, for example, determination, investigation, adjustment, test and send. Thus, the parts ought to be aggregately put away and kept up for programming segments reuse. The reuse usefulness of the segment store is actualized and coordinated with part based application improvement instrument.

Reuse Software Component Repository

Apart is a fundamental component and in this manner, every segment ought to be created to have reusability with the high union and low coupling. In the CBSE (Component-Based Software Engineering), every individual segment can be utilized for segment gathering to build up the segment based programming, which is created in a composite segment. The recently created composite part additionally turns into a reusable segment like a unit segment. To reuse existing parts, there are utilitarian necessities that are outlined as following; putting away and perusing of segments, simple access and extra backings required in the reuse process.

Every part ought to be accessible to recognize exceptional and get to advantageously consistently of access demands. The distinguishing proof of reusable segment ought to be known to figure out which parts are to be mostly or completely altered. Relating to the normal change degree, the reusable parts can be isolated into the segment to be recently created and to be altered. Since the inside interface of a segment is a specialized technique for interconnecting other segments, the basic and practical data is given to check interface coordinate degree and gauge reusable pieces of a part.

The segment interconnection and made out of parts is the meaning of the segment-based framework. The structure of reusable parts and interconnecting are portrayed as segment engineering in theoretical level utilizing reusable segments and connectors. The interfaces of every segment carry on as an interface procedure and setups for part engineering. Figure 1 speaks to the reusable part storehouse that displayed to characterize interconnection among reusable segments. It is characterized and refined to speak to interconnecting part in detail.

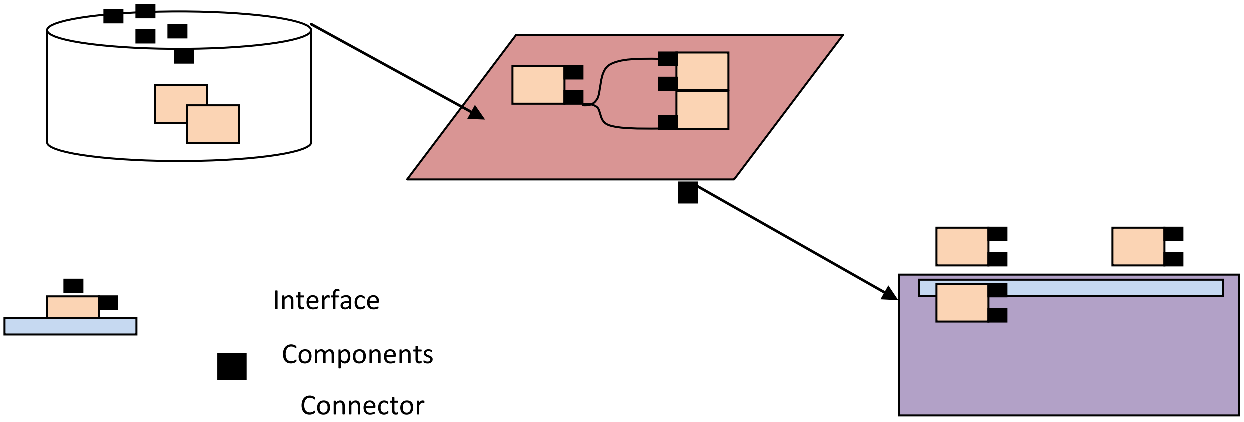

Figure 1 Component Model with Reusable Components

The principle point of the software reuse is to misuses the significant disappointment of the product. The emergency of the reuse is given underneath:

- It keeps running the extension.
- It keeps running after some time.
- Software wastefulness
- Software low quality
- Project is unmanageable.

The rest of this paper is illustrated as follows. Section 2 describes related works about the genetic algorithm in software reusability; section 3 explains the proposed algorithm with flowchart; section 4 we have the experimental results and section 5 concludes and explains the future scope.

2. LITERATURE SURVEY

Rajani Devi et al(2017) Reuse of programming part benefits in the development of new applications by utilizing the effectively open segments. Reusable segments can be utilized in the most fundamental and viable route for the improvement of efficiency and nature of programming ventures. To reuse segments from the vault, we need to concentrate on choosing the suitable recovery process. The Choice of choosing the best or best fit segment and fantastic conceivable determination of the Component which we have recovered is trickier than the straightforward recovery of the part. Greater improvement is done on programming reuse activities to locate the reusable parts in an efficacious manner. In any case, looking and recovery of such data from the archive is a noteworthy worry in writing, once there is a space present between what is put away in the store and what the s/w specialist is eager to recover. In this paper, first we are ordering the parts and second, we are executing operational Semantic-based methodologies and Genetic calculation based methodology for hunt and recovery of the segments.

Jihyun Lee et al(2003)Identifying a reusable segment and its usefulness is imperative to elevate profitability of part based programming. A segment archive is required to encourage putting away and keeping up reusable programming segments productively. This paper depicts the segment storehouse, which supports to encourage reuse of programming segment in part based programming advancement. Our part archive gives usefulness to segment reuse procedure, for example, particular review, adjusting, testing, and sending. Through the part store with these capacities, segments can be proficiently reused in segment reuse process.

P. Niranjan et al(2011)The substance of programming reuse is the utilization of designing information or antiques from existing programming parts to construct another framework. Programming reuse can altogether improve the nature of programming items and diminishes the general advancement cost. Programming reuse storehouse must be structured and created so that they can without much of a stretch find the parts dependent on the prerequisites of the engineers. This work proposes another approach for proficient characterization and recovery of mixed media programming parts dependent on client necessities by utilizing quality grouping plan with hereditary calculation. In this savvy characterization, we utilize Genetic calculation that plays out the arrangement of reusable programming parts in a clever way and recovers the segments dependent on the prerequisites of the engineers.

Ramu Vankudoth et al(2017) In the current situation of the product advancement the principle elements like time and cost for the part determination is excessively short and is additionally extremely hard to distinguish the best-coordinating segment from the segment store. Different research and strategies for the best ideal part choice models exist with significant to the information approval and check to make an interpretation of the commented on models into segment execution. The choices are frequently taken on the transitory reason for the segment choice from the part storehouse. The prior strategies and procedures for the segment assessment and choice do not meet the attributes and assessment of the practical and non-useful prerequisites. A powerful answer for the assessment and choice of the reusable programming segment from the part vault is an exceptionally hot issue for the product improvement network. In this paper, the exploration will focus on the issue that has a place with the assessment and choice procedure to locate the best ideal part of the segment storehouse by utilizing the hereditary calculations.

Kwong et al. (2010) Described that the main stage in the part based programming building is to choose the product segments from the segment vault. In the wake of finding the best programming segments, the following stage is to choose the best ideal part for every product segments. This advancement model has two noteworthy stages, in the principal stage, the useful execution of the part based programming framework expanded and in the second stage the union usefulness is expanded however diminished the coupling usefulness. The system gives the product improvement condition in which attempt to diminish programming advancement time, cost and human exertion for autonomously programming improvement. All these autonomous programming segments are collected together to grew new huge and complex frameworks.

Kaur and Goel (2011) author described the significant advantages of the part vault that store an enormous number of the segment in the composed and methodical way. Segment archives ought to be skilled in addresses the changing and developing of the product association and enterprises. To recover the best ideal part of the huge segment archive is the most significant component of the reusable programming segments. The creator additionally portrayed the various parts of the segment vaults, for example, segment looking through philosophies and order like Free Text, Enumerated, Attribute esteem and faceted orders. The end clients and programming engineers can configuration altered and abnormal state questions to recover the powerful part from the segment vault as indicated by the prerequisite of the end clients.

B. Abdullah et al (2009) The reason for the work portrayed in this paper is to give an interruption recognition framework (IDS), by applying hereditary calculation (GA) to organize interruption identification framework. Parameters and development process for GA are talked about in detail and actualized. This methodology utilizes data hypothesis to channel the traffic information and in this way lessen the intricacy. We utilize a straight structure standard to group the system practices into typical and strange practices. This methodology connected to the KDD99 benchmark dataset and acquired high discovery rate up to 99.87% just as a low false-positive rate of 0.003%. At last the consequences of this methodology contrasted and accessible AI procedures.

3. PROPOSED METHODOLOGY

The proposed flowchart of programming reusability based part programming store utilizing a genetic algorithm is appeared underneath as figure 2. The repository components are created, put away and recovered. It is critical to choose the best possible recovery strategies. In this paper connected a genetic algorithm for compelling inquiry and recovery of parts. Every single part is put away in the store with characterized traits. It is choosing the important parts just as gives awards to pick the most alluring programming reusable segment with the assistance of a genetic algorithm.

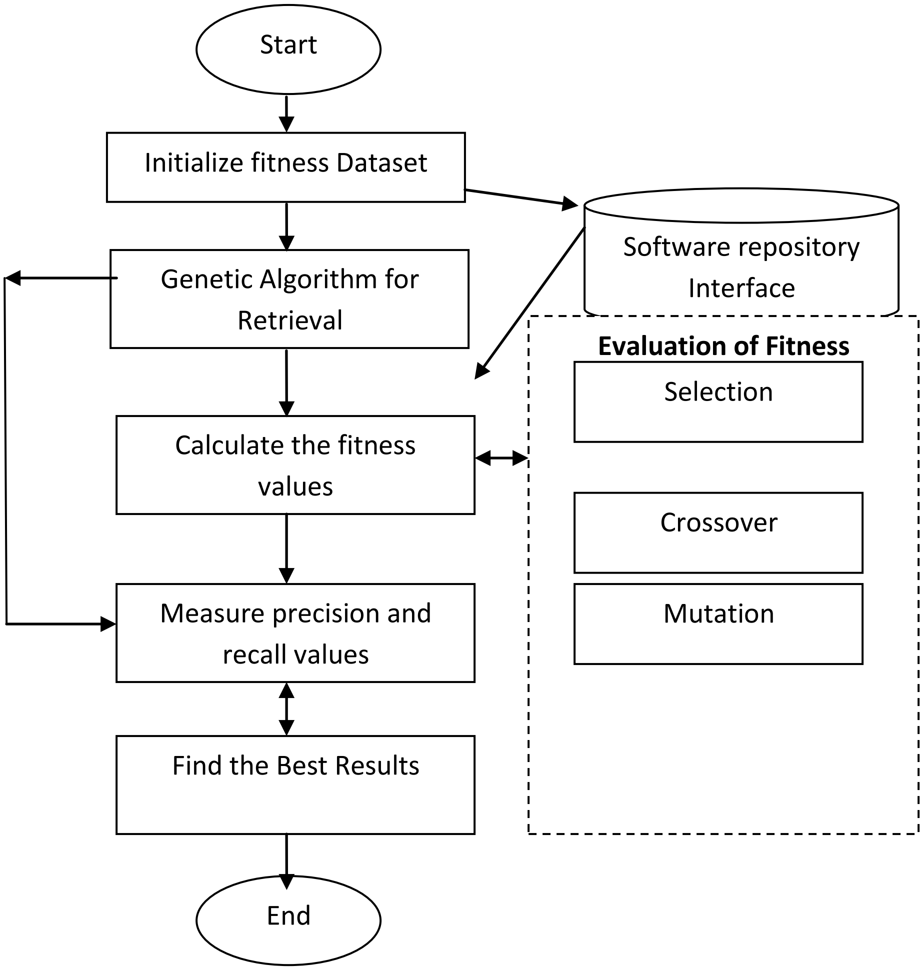
Start
Initialize fitness Dataset
Software repository Interface
Genetic Algorithm for Retrieval
Evaluation of Fitness
Selection
Calculate the fitness values
Crossover
Measure precision and recall values
Mutation
Find the Best Results
End

Figure 2 Flowchart of Proposed Methodology

4. GENETIC ALGORITHMS

Genetic algorithms are versatile heuristic quest methods for hereditary determination. It is non-deterministic inquiry calculations dependent on the instruments of choice. Genetic algorithms are powerful in numerous regions and search a colossal issue space while misusing verifiable data on new pursuit focuses. The Genetic algorithm locates a fitting answer for the issue by hereditarily the populaces. The Genetic algorithm is experiencing determination within the sight of variety by inciting administrators, for example, change and recombination.

GA advances a populace of initial to a populace of excellent people, where every individual represents to an answer of the issue to be unraveled. Every indidual is known as a chromosome and is made out of a foreordained number of qualities. The nature of each standard is estimated by a wellness work as the quantitative representation of each standard's adjustment to a specific situation. The technique begins with an initial populace of arbitrarily created individulas. At that point, the populace is developed for various ages while step by step improving the characteristics of the people in the feeling of expanding the wellness esteem as the proportion of value. During every age, three fundamental genetic operators are successively connected to every person with specific probabilities, for example, choice, hybrid, and transformation. The calculation stream is introduced in Fig. 3.

Received: 10 Jul 2020| Revised: 20 Aug 2020| Accepted: 02 Sep 2020

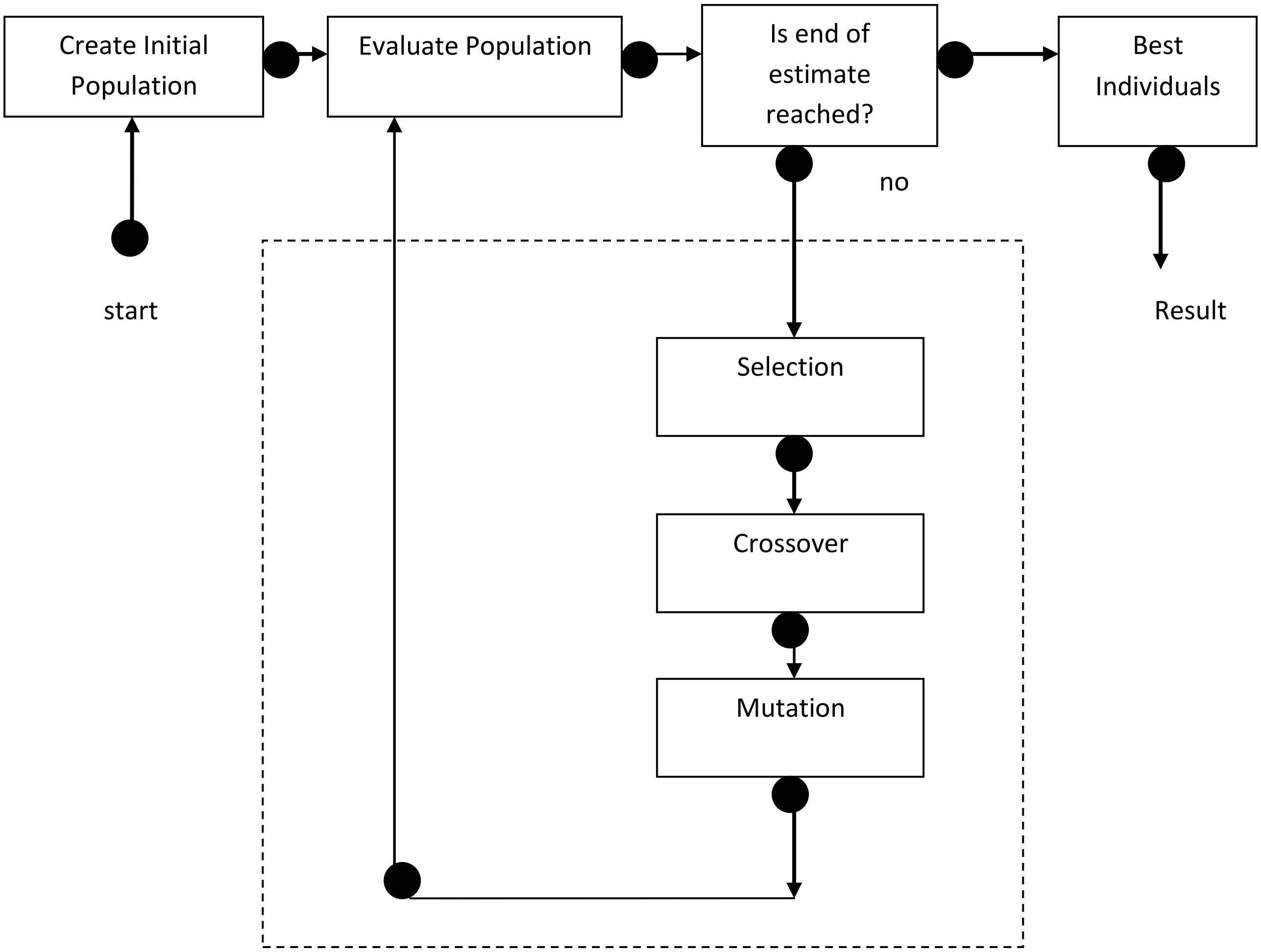

Figure 3 Genetic Algorithm Flow Chart

WORKING PRINCIPLE OF GENETIC ALGORITHMS

The Working principle of genetic algorithm is contains a set of population, fitness values, mutation and selection. To give better motivation of old population than the other one. The solutions are selected according to their fitness to form new solutions. The process will continue until the some condition is satisfied.

Algorithm of GA

Step 1: Represent the size of the population N,the crossover probability pc and the mutation probability pm.

Step 2: Describe a fitness methods to calculate the performance or fitness values.

Step 3: Randomly choose an initial population of size N:x1,x2,…,xN.

Step 4: Evaluate the fitness of each individual methods: f(x1),f(x2),….,f(xN)

Step 5: Select a pair of the current population. The initial population values are selected with a probability related to their fitness.

Step 6: By applying the genetic operators of crossover and mutation. The pair of offspring is measured.

Step 7: Placed the created offspring in the new population.

Step 8: Repeat Step 5 until the size of the population becomes equal to the size of the initial population, N.

Step 9: Replace the initial population with the new population.

Step 10: Go to step 4, and repeat the process until the termination criterion is satisfied.

Step 11: The genetic algorithms performance is influenced by crossover and mutation operators. The below figure 4 represents the genetic algorithm is shown

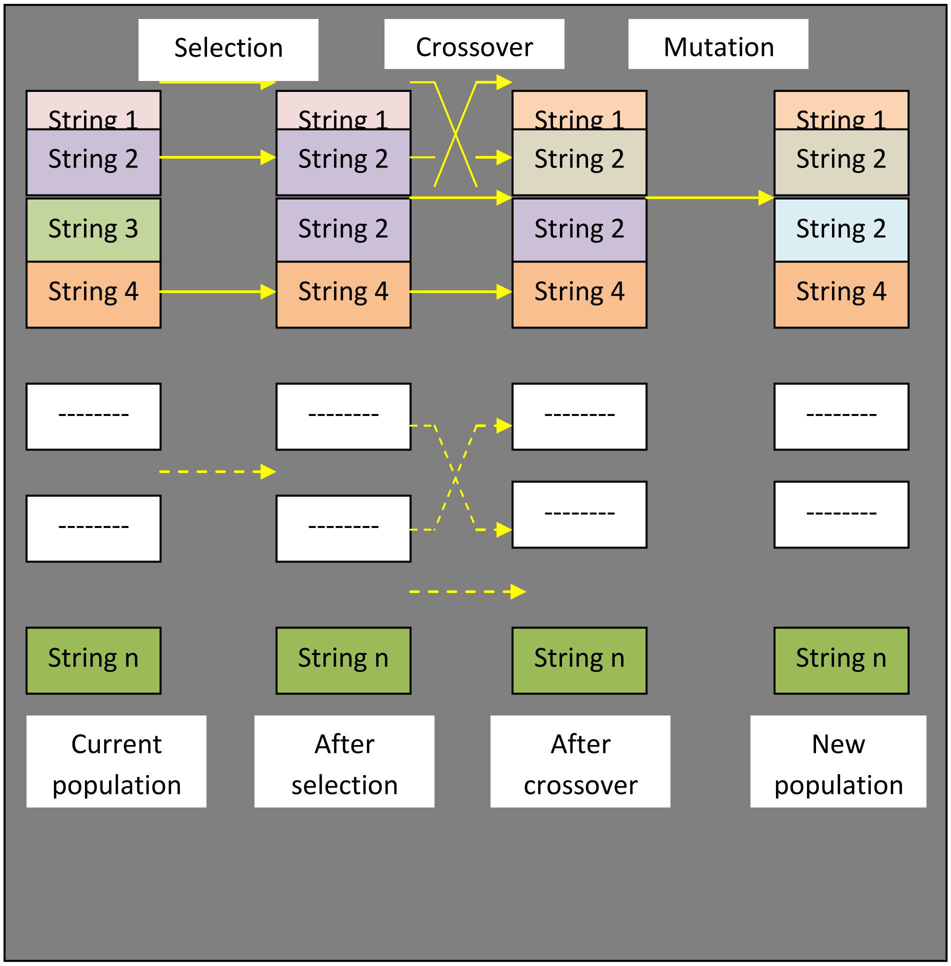

Figure 4 Representation of Genetic Algorithm

Pseudo code of Genetic Algorithm

```
G=0;
Generation Counter
Initiate_Population POP(G);
        //Initial population generation
Fitness_Evaluation POP(G);
        //Evaluates Individuals Fitness
While G<Max_Gen
Terminating Criterion
Begin
        G=G+1
Increase Generation Counter
Select Initial
```

Received: 10 Jul 2020| Revised: 20 Aug 2020| Accepted: 02 Sep 2020

```
        //Select Initial value from the population
Crossover;
        //Apply crossover to selected parents
Mutation
        //Apply mutation to new population
Evaluate_Fitness;
        //Evaluate Fitness of New population
Select best population for next
        //Select survivor Individuals
Generation
End
```

Start and end of “Genetic Algorithm” is the other optimization techniques. It is working with the assertions of optimization variables, their costs and end up with an optimum solution. However, the complete process from beginning to ending is simple different from other optimization techniques.

One of the objectives of “Genetic algorithm” is the best possible solution from the static and finite sized population in a minimum amount of time. The working process of genetic algorithm has two phases. The first phase is performed from the selection process and the second phase after selection is manipulated to create new generation. The main advantages is purely non-heuristic search throughput the solution is performed. Therefore, no specific knowledge of the problem is required in advance. These are flexible and robust in solving the complex problems.

GENETIC ALGORITHM BASED RETRIEVAL

Genetic algorithm was developed to progress classifiers and chooses the selectable result in provision of a number of components in the corresponding classes that facilitates in number of steps as follows:

- Randomly choose the 100 classifiers for each generation of genetic algorithm.
- Apply crossover operation to each pair of classifiers and hand-picked consistent with the crossover chance.
- Apply mutation to randomly selected classifier according to mutation probability,
- Perform component classification for each of the 100 classifiers,
- It compares classifiers values of characteristics with each and every component. If a component is close enough to classifiers, then assigned to the class and represented as classifier.
- Choose the top 20 classifiers in the provision of the number of assigned components. Then find the average of the components from the 20 classifiers.
- The average fitness values of current generation is bigger than the before generation, then the new population is created by selection according to their fitness and repeat step 3.otherwise do not produce new population and repeat step 2.

The above algorithm process is continues till the state reaches to termination. The algorithm is terminated suppose the average fitness is not done properly.

OPTIMIZATION USING GENETIC ALGORITHM

The genetic algorithm is connected in different fields of use and sort out the mind-boggling issue to locate the best arrangement with improved execution. The primary objective of the proposed framework is to look and recover the ideal segment structure from the component repository using genetic algorithm.

Crossover

A solitary Site Crossover continues in three stages. First is the choice of two individual string sets indiscriminately for mating in proliferation administrator. Second is the determination of cross-site is picked aimlessly and third, is the String esteems are traded among two strings.

Mutation

Change bits join combined characteristics 0 and 1 and exchange these characteristics with little change probability Pm. The transformation likelihood is the likelihood where various bits are determined for change. It is additionally significant in transformation search that the assorted variety between the populaces is kept up. The total number of bits changed in the populace is depending upon the three parameters, change probability and populace size and populace length of the general population in the populace.

Various changed bits = Probability transformation * Population size * String lengths.

End Criteria

Stopping the procedure end criteria in hereditary calculations is exceptionally essential. Hereditary procedures comprise of an enormous number of emphases. An end standard gives significant data when to stop the hereditary procedure. After the foreordained number of period hereditary method end the hereditary figuring's and tests the character of the best individual from the populace other than the portrayed issue. There are two pieces of the end criteria.

Fitness Convergence

A Termination condition is a technique that stops the development when the fitness gets met. Fitness is considered to combine when the distinction between normal wellness over the present populace and past populace is not exactly the worth indicated.

Generation Number:

In specific cases, the Stopping criteria or end criteria aren't accomplished even after an enormous number of cycles, all things considered, Generation Number is utilized. It is an end strategy that stops the development when the predefined most extreme number of emphases has been run. The end state of the proposed calculation isn't static as it changes according to the client inquiry. The most significant parts can be accomplished uniquely based on the client question. The GA merges when the normal fitness estimation of the populace (AvgF) ends up equivalent to the client question fitness esteem. Each time the client enters a question, the end state of GA changes progressively. It is characterized as

$$AvgF(P) = Q(fv)$$

The proposed algorithm is defined as

Assign predefined values and encoded values to the attributes in the database.

The objective function to be used by the GA process

$$F = \sum_{i=1}^{n} w_i$$

Add the components into the repository and evaluate the fitness of each components

Fitness value=Sum of all the attribute weights.

$$C(fv) = \sum_{i=1}^{n} w_i$$

To code the search components to some finite values to binary 1s and 0s.

1.calculate the fitness value for the component to be searched

Received: 10 Jul 2020| Revised: 20 Aug 2020| Accepted: 02 Sep 2020

$$Q(fv) = \sum_{i=1}^{n} w_i$$

2.Apply GA for retrieval of the relevant component

Start the random population P.For each iteration doSearch the average fitness valueSelectionCrossoverMutationUntil the termination conditionThe termination condition for the GA is the average fitness value must be equal to the fitness value of the user query.

5. EXPERIMENTAL RESULTS

The experimental result used WEKA tool and is measures two retrieval mechanism such as recall and precision. The existing hybrid approach uses two phases,first phases is keyword search based components retrieval and later passes to GA and select the best component among the retrieved components. Due to some limitation in keyword search and in second phases is possibility of retrieving non relevant components. So GA method is applied to reduce the relevant component choosing.

The recall used to retrieve the relevant components and ratio of the relevant retrieved component to the total number of relevant components in repository.

$$Recall = \frac{Ratio\ of\ the\ number\ of\ relevant\ retrieved\ documents}{Total\ \ number\ of\ relevant\ documents\ in\ document\ collection}$$

Precision means that all the retrieved components are exact as per the user query and ratio of the relevant retrieved components to the total number of retrieved components.

$$Precision = \frac{Ratio\ of\ the\ number\ of\ retrieved\ documents}{Total\ \ number\ of\ retrieved\ documents}$$

These are two measures that are extensively used in retrieval performance.Genetic Algorithm's typically need a single valued measure for fitness evaluation of an individual in the population Table 1 belows shows the calculated measures of some retrieval components for our proposed work

Table 1 Retrieval Components of our Proposed Work

S.No	Name of component retrieved	Recall	Precision
1	Binary search	0.65	0.6
2	Sort	0.75	0.8

Received: 10 Jul 2020| Revised: 20 Aug 2020| Accepted: 02 Sep 2020

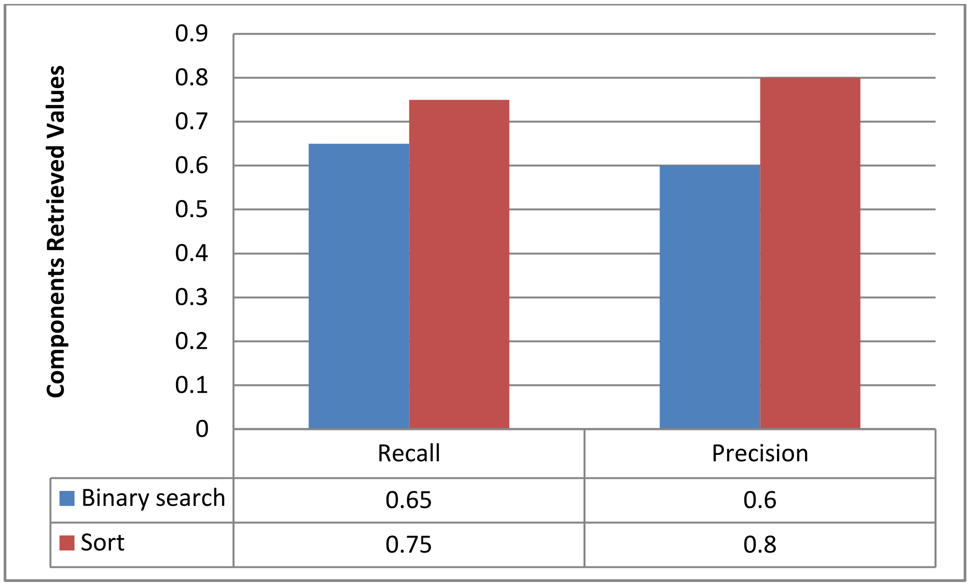

Figure 5 Graphs of Components Retrieval Values of Recall and Precision

Table 2 Components Retrieved using Genetic Algorithm

Component ID	Name of the Component	Fitness Value
100	Quick Sort	148
101	Linear Search	143
102	Binary Search	136
103	DES encryption	123

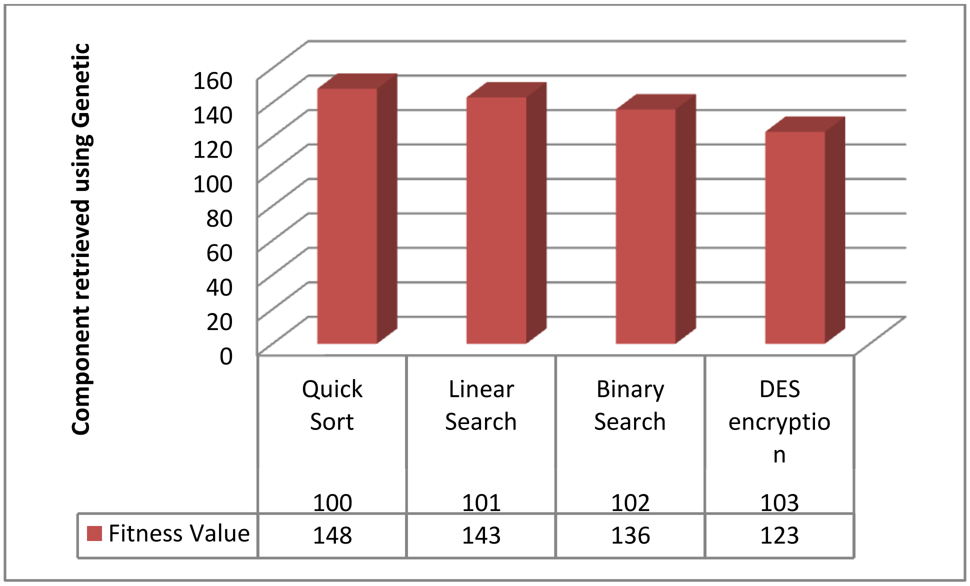

Figure 6 Graph of Fitness Value of Component Retrieved using Genetic Algorithm

The figure 5 shows the graphical representation of recall and precision calculation result of the components-based retrieval and Table 2 and figure 6 shows the graphical representation of fitness results of proposed algorithm.

6. CONCLUSION

This paper is concludes that reuse of software is based on the effective search and retrieval of information. The software reuse is ineffective without search and retrieval mechanisms. Here components are classified and for effective retrieval of the components that implemented the genetic algorithm and additionally precision and recall concepts are evaluate the method of retrieving software components.The proposed algorithm is worked on the principle of the genetic algorithm. The mainly focused on the termination condition.The GA finds the average fitness value of population that becomes equal to the fitness value of user queries.The other benefit of GA is fast calculation of the fitness value because in the insertion process the fitness value is calculated and stored in the database. In future can be extended to initialize weights dynamically without storing in the predefined values. The selection and crossover operators can be even changed to get more optimized results.

7. REFERENCES

1. T. Rajani Devi,B. Rama "Designing Software Reuse Repository Through Intelligent Classification for Effective Search and Retrieval Mechanism" International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB17) IEEE,2017.
2. Jihyun Lee, Jinsam Kim, and Gyu-Sang Shin "Facilitating Reuse of Software Components using Repository Technology" Proceedings of the Tenth Asia-Pacific Software Engineering Conference (APSEC'03) IEEE,2003.
3. B. Abdullah, I. Abd-alghafar, Gouda I. Salama and A. Abd-alhafez "Performance Evaluation of a Genetic Algorithm Based Approach to Network Intrusion Detection System" AEROSPACE SCIENCES & AVIATION TECHNOLOGY, pp:1-17,2009.
4. Ramu Vankudoth,Dr. P. Shireesha,T. Rajani Devi "A Model of System Software Components Using Genetic Algorithm and Techniques" International Journal of Advanced Research in Computer Science and Software Engineering Volume 6,Issue 9,2016.
5. Kaur, V. and Goel, S "Facets of Software Component Repository" International Journal on Computer Science and Engineering, Volume 3,Issue 6, pp.2473-2476. 2011
6. Kwong, C.K., Mu, L.F., Tang, J.F. and Luo, X.G., "Optimization of software components selection for component-based software system development". Computers & Industrial Engineering, Volume 58,Issue 4, pp.618-624.,2010.
7. Jiang Guo, Luqi "A Survey of Software Reuse Repositories" Seventh IEEE International Conference and Workshop on the engineering of computer based systems,2000.
8. Swathy Vodithala, Suresh Pabboju "A Dynamic Approach for Retrieval of Software Components Using Genetic Algorithm" IEEE International conference on software engineering and service science,2015.
9. Faisal Islam "An Effective Approach for Evaluation and Selection of Component" Computer Engineering and Intelligent Systems Vol.8, No.1, 2017,pp-1-6.
10. Kamna Mahajan, Mandeep Kaur "Component Retrieval Using Genetic Algorithm Based Optimization Technique" IJCST Vol. 4, Issue 2, 2013,pp-585-587.
11. Reena, Pradeep Kumar Bhatia "Application of Genetic Algorithm in Software Engineering: A Review" International Refereed Journal of Engineering and Science (IRJES) Volume 6, Issue 2, 2017, pp. 63-69.
12. Ted J.Biggerstaff "An Assessment and Analysis of Software Reuse" Advances in ComputersVolume 34, 1992, pp-1-57.