Volume 5, Issue 7, July 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Software Component Retrieval Using GA and ACO

Iqbaldeep Kaur*
Associate Professor, CSE Department,
Chandigarh University, India

Satvir Kaur
Research Scholar, CSE Department,
Chandigarh University, India

Abstract— Software reuse is the process of using existing software components to develop new software product. Searching and retrieving reusable components for efficient software reuse are the major challenges faced during reuse. To reuse the any software component, there should be an efficient retrieval mechanism to retrieve the reusable component from repository.

In this paper, a brief introduction is given to the software reuse, reusable software components, process of component retrieval. Results of component retrieval using genetic algorithm and Ant colony optimization are discussed.

Keywords— Software reuse, reusable component, component retrieval, genetic algorithm, ant colony optimization.

I. INTRODUCTION

Software reuse means to develop any software applications by reusing the existing software components by simply embedding them as such or with slight modifications. Software component is an independent piece of software with a specific functionality. Software component can be used as individual or can be embedded with any other software to perform any particular function. Reusing software components reduces both the development time and cost.

But efficiency of software reuse depends on the efficient retrieval of the required software component from the repository. Component retrieval is the process of fetching the required software component from the large collection of components or software repository.

The basic architecture of component retrieval process is as shown in figure 1.

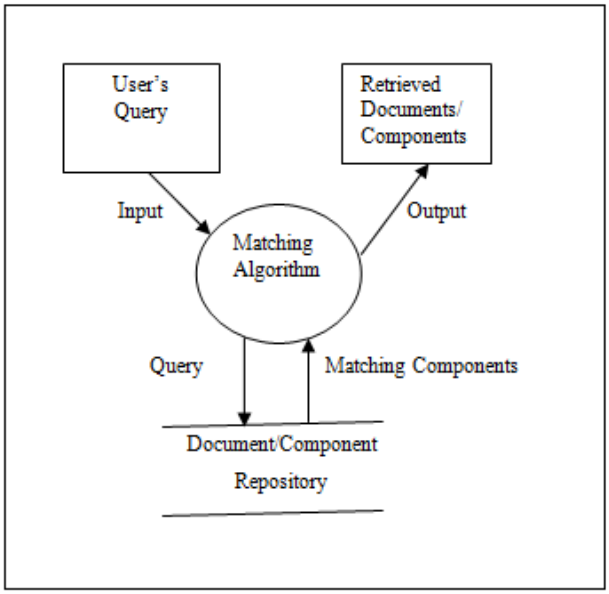


Fig 1: Component Retrieval Process

In component retrieval process, the user inputs his/her query to the retrieval system, this input is matched with the components that are stored in the repository on the basis of some matching algorithm. The components that match with user query are retrieved back to user.

There are various techniques that have been used for component retrieval from repository which includes traditional keyword based retrieval, faceted based retrieval technique, hypertext based retrieval, semantics based retrieval and knowledge based retrieval techniques.

In keyword based retrieval, software components are stored with some keywords. During retrieval, input query is matched with these stored keywords of components and the components whose keyword match with user input are retrieved from the repository. Keyword based search reduces the search space but it also results in ambiguity in retrieved components as all possible components are retrieved [11].

In faceted based retrieval, each component is classified into different facets i.e. attributes. Each facet is the important characteristics of the component Search and retrieval of components is improved by faceted classification and is also helpful in the development of a standard vocabulary for components and their attributes. Faceted classification divides the schema divided into several facets and each facet again has several terms [11, 13].

Semantic based retrieval technique retrieves the components on the basis of description of the component and its domain [11].This technique retrieves components on the basis of domain or features of the components. Components are retrieved by matching with the meaning given for every component in the repository. Ontology based knowledge base stores the semantic information of the components [5].

In hypertext technique the reusable components are organized in non-linear network of nodes and links. Hypertext technique is based on the concept of network of nodes and interconnection between them where each node is a reusable component or unit of textual information and interconnection between nodes are represented as links [1]. The node from which the link starts is parent node and where the link ends is child node. Components can be searched by traversing the links.

In knowledge based technique the components are retrieved using reasoning that relates the user query with the components or with an old query. The fuzzy rule set is described in knowledge base. The matching of the query with the component is done using fuzzy rules [3].

II. USE OF GENETIC ALGORITHM AND ACO IN COMPONENT RETRIEVAL

Genetic algorithm is based on evolution by natural selection. The approach is inspired by Darwin's theory of evolution, which is based on the survival of the fittest. A solution to a problem is considered an individual in a population of solutions. GAs update a population of individuals iteratively [2]. In every iteration of population, the individuals are evaluated using a fitness function. New population of individuals is generated by applying genetic operators such as Crossover and Mutation to create new offspring.

Pseudo code for genetic algorithm is as follows:

Algorithm: GA(n, χ, μ)
//**Initialize generation 0:**
k:=0;
p_k:= a population of n randomly- generated individuals;
//**Evaluate** p_k:
Compute *fitness*(i) for each i ϵ p_k
do
{
//**Create generation** $k+1$**:**
//**1. Copy:**
Select(1-χ)× n members of p_k and insert into p_{k+1};
//**2. Crossover**:
Select χ × n members of p_{k+1}; produce offspring; insert offspring into p_{k+1}
//**3. Mutate**:
Select μ× n members of p_{k+1}; invert a randomly selected bit in each;
//**Evaluate** p_{k+1}**:**

Compute *fitness*(i) for each i ϵ p_k;
// **Increment:**
$k = k + 1$**:**
}
while fitness of fittest individual in p_k is not high enough;
return the fittest individual from p_k

Genetic algorithm has been used in retrieving components. P.Pathak et.al (2000) used genetic algorithm to improve the performance of component retrieval by adapting the matching functions. It was assumed that components retrieved by adapting matching functions would give better retrieval results than a single matching function [2]. S. Andreou et.al (2006) used genetic algorithm for classification of software components into clusters. This classification reduces the search space to search components [4].

Ant colony optimization is a heuristic technique that is inspired from biological nature of ants. ACO technique is used to solve combinatorial optimization problem. Basic idea behind ACO is technique followed by ant colonies to find their food. Ant use pheromone trials to find shortest path between their nest and food. Ant colony optimization has been used by many researchers in component retrieval.

S.G.Khode et.al (2009) used ant colony optimization in retrieving software components. In their proposed work, first components are retrieved using keywords and the retrieve components from keyword based search are further refined using ant colony optimization that generates rules for matching the components with user query [6].

R.K.Bhatia (2010) used ant colony optimization to generate rules for representation and retrieval for reusable component retrieval [7].

Pseudocode for ant colony optimization is:

```
procedure ACO_MateHeuristic
  While(not_termination)
       generateSolutions ()
       daemonActions ()
       pheromoneUpdate ()
   end while
 end procedure
```

A. Comparison of genetic algorithm and ACO in component retrieval

P.Niranjan, and C.V.Rao (2013) proposed an methodology that uses genetic algorithm to optimize the retrieval results and to retrieve best components from the repository [10].

In their proposed classification and retrieval technique firstly keyword based technique is used to search all relevant components. After all the relevant components are retrieved by using keyword based search, these components are ranked by using a ranking technique. Finally these ranked components are optimized through genetic algorithm to find out best component. Genetic algorithm is used to get optimal solution of user query.

The optimization of the components using genetic algorithm results in better efficiency, precision and recall than keyword based retrieval.

Different search and retrieval experiments were performed by them to retrieve components from repository the efficiency results of which are shown in following table:

TABLE I RESULTS OF GA

Sr. No	Keyword/ Attributes	Efficiency
1.	Search, C	43.8
2.	Sort, C	53.7
3.	Java, Solaris	55.7

Table 1 shows efficiency results of three search operations performed in above discussed method.

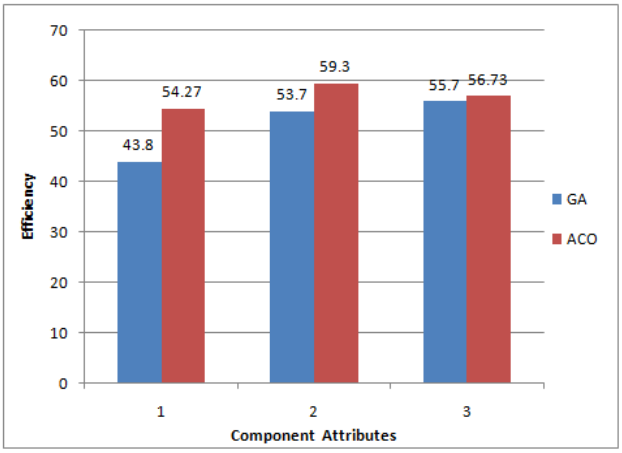
When in the above discussed technique, optimization using Genetic Algorithm is replaced by Ant Colony optimization; it shows better efficiency.

In proposed methodology, components are first searched by their features or characteristics that are stored in the repository. This results into all the relevant components to the user query. After this the components are ranked according to their download count. After this the components are optimized using ant colony optimization. The efficiency results of this proposed method are shown in following table:

TABLE I RESULTS OF ACO

Sr. No	Keyword/ Attributes	Efficiency
1.	Search, C	54.27
2.	Sort, php	59.3
3.	Java, Windows	56.73

Table 2 shows the results efficiency of ACO in retrieving components in proposed system. It is clear from above tables that efficiency has increased by using ACO for optimization of retrieved component with keyword based search.
Following given is the graph to compare results of both techniques:



In figure 1, the comparison of efficiency of GA and ACO based optimization is shown. The efficiency of genetic algorithm based optimization varies from 43.8 to 55.7 and efficiency of ant colony optimization based optimization varies from 54.27 to 59.3. It is clear that efficiency is increased by using ACO.

III. RESULTS AND CONCLUSION

In this paper, the efficiency scores are compared for component retrieval using GA and ACO. It is clear that efficiency using ACO is better than GA based optimization. Genetic algorithm shows efficiency ranges from 43.8 to 55.7 whereas efficiency range is increased upto 54.27 to 59.3. So, ACO increases efficiency of component retrieval.

REFERENCES

[1] S. Jacques, and D. Desbois, "Information retrieval in hypertext systems: an approach using Bayesian networks," *Electronic publishing* 4, no. 2 pp. 87-108, 1991.

[2] P. Pathak, M. Gordon, and W. Fan, "Effective information retrieval using genetic algorithms based matching functions adaptation," In*System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, pp. 1-8. IEEE, 2000.

[3] I. Sora, and D. Todinca, "Specification-based retrieval of software components through fuzzy inference," *Acta Polytechnica Hungarica* 3, no. 3 pp. 123-135, 2006.

[4] A. S. Andreou, D. G. Vogiatzis, and G. A. Papadopoulos, "Intelligent classification and retrieval of software components," In *Computer Software and Applications Conference, 2006. COMPSAC'06. 30th Annual International*, IEEE, pp. 37-40, 2006.

[5] C. Li, X. Liu, and J. Kennedy, "Semantics-Based Component Repository: Current State of Arts and a Calculation Rating Factor-based Framework," In *Computer Software and Applications, 2008. COMPSAC'08. 32nd Annual IEEE International*, pp. 751-756. IEEE, 2008.

[6] S. G. Khode, and R. Bhatia, "Improving Retrieval Effectiveness Using Ant Colony Optimization," In *Advances in Computing, Control, & Telecommunication Technologies, 2009. ACT'09. International Conference on*, IEEE, pp. 737-741, 2009.

[7] R.K.Bhatia., M. Dave, and R. C. Joshi, "Ant colony based rule generation for reusable software component retrieval," *ACM SIGSOFT Software Engineering Notes* 35, pp. 1-5, 2010.

[8] N. A. Saiyd, I. A. Said, and A. H. Takrori, "Semantic-Based Retrieving Model of Reuse Software Component," *IJCSNS* 10, pp. 154-161, 2010.

[9] J. K. Sudhakaran and R. Vasantha, "A mixed method approach for efficient component retrieval from a component repository," *Journal of Software Engineering and Applications*, pp. 442-445, 2011.

[10] P. Niranjan, and C.V.G Rao, "Dynamic grading of software reusable components for effective retrieval of components," In *Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), 2013 International Multi-Conference on*, IEEE, pp. 7-13, 2013.

[11] S. Yadav and K. K. Mangat, "Design of rank based reusable component retrieval algorithm," *International journal of advanced research in computer science and software engineering, Vol* 3, pp.850-857, 2013.

[12] A. Bakshi, "Development of a software repository for the precise search and exact retrieval of the components," *International Journal of Advanced Research in Computer Science and Software Engineering, Vol 3*, pp. 1116-1126, 2013.

[13] S. Singh, "An experiment in software component retrieval based on metadata and ontology repository," *International Journal of Computer Applications (0975–8887) Vol* 61, pp.33-40, 2013.

[14] A. Kumar, "Software Reuse Library Based Proposed Classification for Efficient Retrieval of Components," *International Journal of advanced research in computer science and software engineering," Vol 3,* pp.884-890, 2013.

[15] A. Bakshi, "Development of a software repository for the precise search and exact retrieval of the components," *International Journal of Advanced Research in Computer Science and Software Engineering, Vol 3*, pp. 1116-1126, 2013.

[16] V. Verma, M. Mehta, and N. Adlakha, "Augmentation of Component Retrieval Using Ceremonial Methods," *International Journal of Research in engineering,* vol.4, pp.26-35, 2014.

[17] C. Srinivas , V. Radhakrishna, and C.V.G. Rao, "Clustering and Classification of Software Component for Efficient Component Retrieval and Building Component Reuse Libraries," *Procedia Computer Science*, vol.31, pp.1044-1050, 2014.

[18] K. Sandhu, and T. Gaba, "A Novel Technique for Components Retrieval from Repositories," *An International Journal of Advanced Computer Technology*, vol.3, pp.912-920, 2014.

[19] N. Madaan, and J. Kaur, "A Survey on Selection Techniques of Component Based Software," *An International Journal of Advanced Computer Technology,* vol.4, pp.1245-1250, 2014.

[20] K. Sandhu, and T. Gaba, "A Novel Technique for Components Retrieval from Repositories," *An International Journal of Advanced Computer Technology*, vol.3, pp.912-920, 2014.

