ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)
On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

Multi-objective Ranking based Non-Dominant Module Clustering

K.Sarojini[1]
Department of Information Technology, SIES College, University of Mumbai, Sion (west), Maharashtra, India. [1]

Abstract—Although there has been a demarcation between development and evolution (maintenance) of software, this is increasingly irrelevant as fewer and very fewer systems are completely new. Additionally, once a software engineer understands a system's structure, it is difficult to preserve this understanding, because the structure tends to change during maintenance. So Software engineers greatly emphasize on good modular structures as well modularized software is easier to develop and maintain. In this paper, I propose two algorithms, one as a search optimization technique and another as a multi-objective fitness evaluation function of the search technique. The former one, I have labeled as Modified Pareto Optimal Genetic Algorithm (Modified par-op GA) and the later one as Non-Dominance Module Ranking Algorithm (Non-DMR). As the Fitness function is a component of GA, I have embedded the Non-DMR within the Modified par-Op GA.

Keywords—Non-hierarchical clustering optimization, cohesion, coupling, reverse engineering, Agglomerative hierarchical clustering, Pareto optimality.

I. INTRODUCTION

Software architecture is a model of the software system expressed at a high level of abstraction [1]. The architectural view of a system raises the level of abstraction and concentrating on only 'black box' elements. Normally, architecture will not be explicitly represented in the code, but it has to be documented. Often, the available documentation is incomplete, outdated or is completely missing, the designers having only the code available, sometimes only in compiled form. So the reverse engineering community has developed lot of techniques and tools to create appropriate abstraction of software structure which makes them more understandable and easier to maintain. Software clustering technique is a key to create these types of abstractions. It follows the emergent of search based approaches to search the similar modules in different clusters. In the search based approach [2] the attributes of a good modular decomposition are formulated as objective, the evaluation of which as a 'fitness function' guides a search based optimization.

Bunch uses a single objective [3] called Modularization quality (MQ) as a fitness function. It combines the twin objectives of high cohesion and low coupling. Cohesion is a measure of the number of intra edges (edges that lie inside a cluster) while coupling is measured by the number of inter edges (The edges that connect two clusters) in the modularization [4]. For example if functions that access the same variables are placed into the same module, the module would tend to have communicational cohesion. Low coupling and high cohesion as a single objective may yield suboptimal results. This led the researches to find Pareto optimality for automated module clustering within multi-objective [5] paradigm. Two formulation of the multiple objective approach are Maximizing clustering Approach (MCA) and Equal size cluster approach (ECA). In MCA modules will not be put in one cluster and series of isolated clusters will not be produced. The ECA aims to minimize the difference between the maximum and minimum number of modules in a cluster through which we can create equal size clusters. In multi- objective paradigm Genetic algorithm is used as a search algorithm. The above will be discussed detail in section 3 and 4. Although the search based approaches do not guarantee to provide optimal solutions, yet, they can obtain near optimal solutions in reasonable amount of computational time.

The rest of the paper is organized as follows. Section 2 describes the background and related study of software module clustering. Section 3 explains the software module clustering as a multi-objective search problem. Section 4 presents Multi-objective genetic algorithm. Section 5 and 6 explains my proposed approach and experimental results with discussions respectively.

II. BACKGROUND AND RELATED STUDY

Optimization algorithms take an initial solution(module) as a parent and tries to improve this solution by iterative adaptations according to some heuristic. The optimization

ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)
On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

method has been used to produce both hierarchical [6] and nonhierarchical [7] clustering. A typical nonhierarchical clustering optimization method starts with an initial partition derived based on some heuristic. Then, modules are moved to other clusters in order to improve the partition according to some criteria. This relocating goes on until no further improvement of this criterion takes place. Nonhierarchical method creates one level of clustering. Since only one set of clusters is output, the user normally has to input the desired number of clusters. In hierarchical approach clusters are created in levels actually creating sets of clusters in each level. Hierarchical methods can be subdivided into Agglomerative Hierarchical clustering and Divisive hierarchical clustering. In the former type, each object initially represents a cluster of its own. Then clusters are successively merged until the desired cluster structure is obtained. In the later, all objects initially belong to one cluster. Then the cluster is divided into sub-clusters, which are successively divided into their own sub-clusters. This process continues until the desired cluster structure is obtained. I have used Divisive hierarchical clustering method in outer algorithm for initial clustering.

Genetic clustering algorithms are randomized global optimal search heuristic that mimics the process of natural evolution, having large amount of implicit parallelism. Genetic algorithms are characterized by attributes, such as the objective function, the encoding of input data, the genetic operators, such as crossover and mutation, and population size. The first multi-objective GA, called vector evaluated GA (or VEGA), was proposed by Schaffer [8]. Afterwards, several multi-objective evolutionary algorithms were developed including Multi-objective Genetic Algorithm (MOGA) [9], Niched Pareto Genetic Algorithm (NPGA) [10], Weight-based Genetic Algorithm (WBGA) [11], Random Weighted Genetic Algorithm (RWGA)[12], Strength Pareto Evolutionary Algorithm (SPEA) [13], Pareto-Archived Evolution Strategy (PAES) [14], Region-based Selection in Evolutionary Multi-objective Optimization (PESA-II) [15], Fast Non-dominated Sorting Genetic Algorithm (NSGA-II) [16], Multi-objective Evolutionary Algorithm (MEA) [17], and Dynamic Multi-objective Evolutionary Algorithm (DMOEA) [18]. Note that although there are many variations of multi-objective GA in the literature, these cited GA are well-known and credible algorithms that have been used in many applications and their performances were tested in several comparative studies.

III. MULTI-OBJECTIVE APPROACH FOR SOFTWARE MODULE CLUSTERING

Complex structure of the software is typically depicted by one or more directed graphs called Module Dependency Graphs (MDGs). These graphs can be used to describe the modules (or classes) of a system and their static interrelationship using nodes and directed edges, respectively (see Fig 1).

Thus clustering can be visualized as a graph partitioning problem, which is known to be NP hard. It cannot be solved efficiently by traditional optimization techniques and in many real-life problems, objectives (see the bulleted list below) under consideration conflict with each other. Hence, optimizing x (a solution) with respect to a single objective often results in unacceptable results with respect to the other objectives. Therefore, a perfect multi-objective solution that simultaneously optimizes each objective function is almost impossible. A reasonable solution to a multi-objective problem is to investigate a set of solutions, each of which satisfies the objectives at an acceptable level without being dominated by any other solution. In multi-objective software module clustering, multiple objectives will become the following:

- The sum of intra-edges of all clusters(maximizing)
- The sum of inter-edges of all clusters(minimizing)
- Modularization Quantity called as MQ (maximizing)
- The number of isolated clusters(Minimizing)

The intra-edges are those in which the source and target of the edge lie inside the same cluster. The inter-edges are those for which the source and target lie in distinct clusters [19]. MQ is the sum of Modularization Factor (MFk) which is the ratio of intra-edges and intra-edges in each cluster. Modularization Factor (MFk) for cluster k can be defined as follows [5]:

$$MF_K = \begin{cases} 0, & if\ i = 0 \\ \frac{i}{i+\frac{j}{2}}, & if\ i > 0 \end{cases}$$

Where 'i' is a sum of the weights of all intra-edges in cluster k and j is the sum of edge weights for all edges that originate or terminate in cluster k. The reason for the occurrence of the term j/2 in the above equation (rather than merely j) is to split the penalty of the inter-edge across the two clusters that connected by that edge.

MQ can be calculated in terms of MF_K as

$$MQ = \sum_{k=1}^{n} MF_k$$

ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)
On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

Where n is the number of clusters.
To illustrate the objectives value in the Multi-Objective approach, consider the MDG in Fig.1.

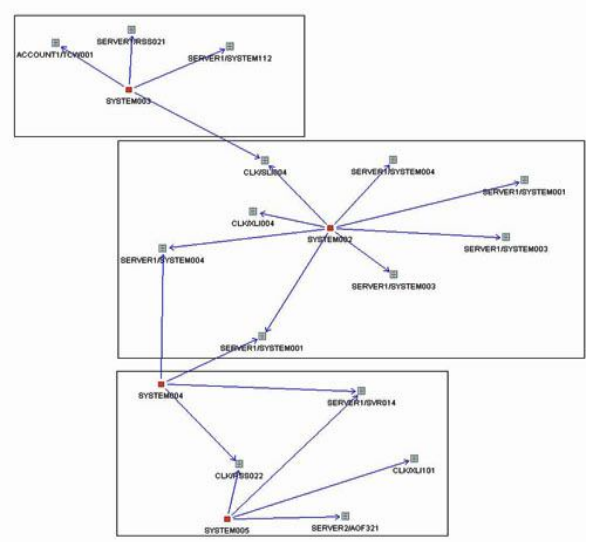

Fig 1. Module Dependency Graph (MDG) in multi-objective approach.

The objective values of Fig 1 are as follows:

- The number of clusters: 3
- The sum of intra-edges of all clusters:17
- The sum of inter-edges of all clusters: 6
- Modularization Quality : 2.5563
- The number of isolated clusters: 0
- The difference between the maximum and minimum number of modules in clusters: 4

One general approach to multiple-objective optimization is Pareto optimal search. A Pareto optimal set is a set of solutions that are non-dominated with respect to each other. While moving from one Pareto solution to another, there is always a certain amount of sacrifice in one objective(s) to achieve a certain amount of gain in the other(s).

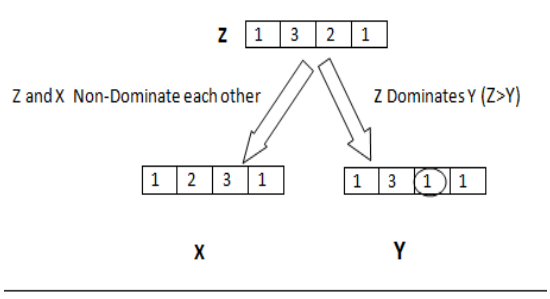

Fig. 2. Non-Dominance solutions selection using Pareto Optimal search.

In the Fig 2, I have taken 3 chromosomes (X, Y And Z) from population N. The genomes of each chromosome are encoded using numerals. Z is compared with X and Y taking the similarities as fitness functions. Similarity is expressed using non-dominance property. Pareto optimality determines solution X is better than solution Y to cluster with solution Z. We can observe that, solution X has more similarities with solution Z than solution Y. So Solution Z and X are non-dominant to each other whereas Z dominates Y.

IV. MULTI-OBJECTIVE GENETIC ALGORITHMS

Genetic algorithms use concepts of population and of recombination inspired by Darwinian evaluation. The concept of GA was developed by Holland and his colleagues in the 1960s and 1970s [20]. GA are inspired by the evolutionist theory explaining the origin of species in nature, weak and unfit species within their environment are faced with extinction by natural selection. The strong ones have greater opportunity to pass their genes to future generations via reproduction. In the long run, species carrying the correct combination in their genes become dominant in their population. Sometimes, during the slow process of evolution, random changes may occur in genes. If these changes provide additional advantages in the challenge for survival, new species evolve from the old ones. Unsuccessful changes are eliminated by natural selection [21].
Genetic algorithms operate with a collection of chromosomes called a population. The population is normally randomly initialized. As the search evolves, the population includes fitter and fitter solutions, and eventually it converges, meaning that it is dominated by a single solution. Being a population based approach, GA are well suited to solve multi-objective optimization problems. A generic single-objective GA can be modified to find a set of multiple non-dominated solutions in a single run. The ability of GA to simultaneously search different regions of a solution space makes it possible to find a diverse set of solutions for difficult problems with different objectives.

V. PROPOSED APPROACH

This section describes our proposed approach. In generic multi-objective optimization approach, all the objectives should be either the minimization objectives or maximization objectives to get optimal solution in single run. But Non-Dominance module ranking algorithm (Non-DMR) does software module clustering without forcefully change minimization objective constraints to maximization or vice versa. A general frame work of

ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)
On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

Modified Par-Op GA and Non-DMR algorithms is given in Fig.3.

Algorithm: Non- Dominance module ranking in GA

1. Set t = 1. Randomly generate N solutions to form the first population, Pt. (P1).
2. Evaluate the fitness of solutions which are in Pt (P1). (To find fitness of individuals in P1 Non- Dominance module ranking method is proposed.)
   - 2.1 Set i =1 and TP =Pt.
   - 2.2 Identify non-dominated solutions in TP and assigned them to set Fi.
   - 2.3 Set TP = TPFi. If TP ≠ 0 go to 2.4, else set i=i +1 And goto 2.2.
   - 2.4 For every solution x Є Pt at generation t, assign Rank r1(x,t)=i. if x Є Fi
3. Crossover: Generate an offspring population Qt as follows:
   - 3.1. Choose two solutions x and y from Pt based on the fitness values.
   - 3.2. Using a crossover operator, generate offspring and add them to Qt. (Qt will also have N number of solutions)
4. Mutation: Mutate each solution x Є Qt with a predefined mutation rate.
5. Fitness assignment: Evaluate and assign a fitness value to each solution x Є Qt based on its objective function value.
6. Selection: Select N solutions from Qt based on their fitness and copy them to Pt+1.
7. If the stopping criterion is satisfied, terminate the search and return to the current population, else, set t= t+1 go to Step 2.

Fig. 3. Modified Par-Op GA and Non-DMR algorithm.

Our proposed algorithm has two different algorithms in which one will be executed within another. The outer algorithm is a modified Par-Op GA which finds non-dominant solutions (modules/clusters) using Pareto optimality as a base. This finds whether one solution is better than another (Not how much better) using non-dominance property through which initial clustering made. In this, I have used Divisive hierarchical clustering in which all modules initially belong to one cluster. By applying Modified Par-Op GA on that cluster, initial clustering partition has made. The inner algorithm, termed as Non-DMR, is applied separately on each resultant clusters of Modified Par-Op GA. This algorithm finds that, each solution is how much better than the other solution by ranking the each solution with respect to multiple objectives. Highest ranked solution is the worst one to satisfy the multiple objectives. Through ranking, decision maker can exactly group similar modules in one cluster. Additionally, ranking helps to increase the number of clusters. Increasing the number of clusters will also increase the MQ (Modularity Quality) which is one of the maximization objectives in module clustering. Non-DMR can be directly applied on the initial cluster without using Modified Par-Op GA. But it will increase search and computational time in considerable amount.

VI. EXPERIMENTAL RESULTS AND DISCUSSIONS

The effectiveness of the algorithm has been studied on 4 different size real world module clusters and among them i have taken one (mtunis - an operating for educational purposes written in the Turing Language) to explain the execution. Initial clusters (see Table.1) are labelled as C1, C2, C3, C4 and C5. Solutions (modules) are labelled as 1.X, 2.X, 3.X, 4.X. and 5.X. All the modules which are termed as 1.X are non-dominant to each other. Likewise modules 2.X, 3.X, 4.X and 5.X. are non-dominant to each other within them. To apply Modified Par-Op GA initially all the clusters are merged together and considered as one cluster called Pt. Pt has randomly generated N solutions (total number of modules in all clusters).

TABLE I
SAMPLE MODULE CLUSTERS TAKEN FROM TUNIS SOFTWARE

Clusters	C1	C2	C3	C4	C5
Modules/solutions in respective clusters	2.2	3.3	3.9	5.9	1.13
	4.8	3.11	2.12	1.3	3.8
	1.8	4.4	5.1	4.6	2.4
	1.5	1.1	4.7	1.16	4.3
	2.8	5.7	3.2	3.14	5.2
	5.6	4.5	2.9	2.1	5.4

Then randomly selected module from Pt is compared (non-dominancy) with its remaining modules. For example if we select solution 3.2 to compare with all other solutions, the resultant solution set will be 3.3, 3.11, 3.9, 3.2, 3.14, 3.8 given in Table 2.. These solutions will be moved to one separate cluster called C3. Though it seems finding optimal solutions by comparing one solution with all other solution is a tiring search, we yield a cluster of solutions (not a single solution) in a single run. Now we have two clusters, C3 which has only 3.X solutions and Pt which has other than 3.X.

In a multi-objective problem a single fitness value can only be obtained if the objectives are grouped into a single utility function. This is not appropriate since a specific set of weight values drives the GA towards specific solutions. After completed one iteration of Modified Par-Op GA, the inner algorithm, Non-DMR starts to execute taking the result of first iteration (here C3)

ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)
On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

as an input. The individuals in the cluster C3 indicated by 3.3…3.8 are ranked for each objective. Here we determine the worst individual(s) (modules) in the group (cluster) by ranking the individuals for each objective (such as high cohesion and low coupling) and selecting the individual(s) with the worst total ranking (see Table 2, below). The lower rank number, here 5 of solution 3.2, indicates a better value for the objective being considered. Same rank numbers are assigned to individuals with equal or similar objective values. The total ranking for each individual is then determined by the sum of all the ranking numbers. The individual(s) with the highest total rank, here 3.11 and 3.14, are replaced by the offspring in the population and put in another cluster called **C33**. Using ranking as a way to determine the worst individual in a group allows good individuals for specific objectives to evolve as well as individuals that are average in two or more objectives. Table 2 shows the result derived from first iteration of Non-DMR.

TABLE 2
NON-DMR APPLIED ON C3

Individual Solutions in C1	Multiple objectives (Obj_1 to Obj_4)				
	Obj_1	Obj_2	Obj_3	Obj_4	Total Ranking
3.3	3	1	2	1	7
3.11	4	4	1	2	11
3.9	2	2	4	2	10
3.2	1	2	1	1	5
3.14	3	4	2	2	11
3.8	1	3	3	1	8

Table 3 shows the result of all the iterations of Modified Par-Op GA. Each cluster has the solutions which are non-dominant to each other in the same cluster.

TABLE 3
RESULT OF MODIFIED PAR-OP GA

Clusters	C1	C2	C3	C4	C5
Modules of clusters C1 to C5	1.8	2.2	3.3	4.8	5.6
	1.5	2.8	3.11	4.4	5.7
	1.1	2.12	3.9	4.5	5.1
	1.3	2.9	3.2	4.7	5.9
	1.16	2.1	3.14	4.6	5.2
	1.13	2.4	3.8	4.3	5.4

In Table.4, Q1,Q_2, Q_3 and Q_4 are multiple objectives of each cluster. I rank the individuals (modules) for each objective. To find the fitness value of each module, individual objective ranking is calculated and showed below. Table 5 depicts the total ranking of each individual for multiple objectives. It is determined by the sum of all the objectives (here O_1 to O_4) ranking numbers in which we get single ranking number for each module. In table 5 ranking are not in sequence.

In Table 6, since the ranking numbers are ordered in ascending, the worst individuals comes bottom of the search space which reduces the search to select those individuals. Fitness of individuals is implicitly given by their objective values and their rank against others. There is an explicit fitness function in this problem to replace the worst individuals from its native cluster. Individual(s) (modules) which have the ranking numbers greater than O_n*3 (O_n is the total number of objectives (here O_n = 4 and O_n*3 = 12)) are worst individual(s) to be removed and replaced to new cluster. The C1r in the Table 6, modules 1.13, 1.16 (refer Table 3 to identify corresponding module names) which have been ranked as 14 and 15(greater than 12) are worst modules. They move to new cluster termed $C_{11,}$ given in Table 7.

TABLE 4
RESULT OF MULTI- OBJECTIVE RANKING IN NON-DMR

C1				C2				C3				C4				C5			
O_1	O_2	O_3	O_4	O_1	O_2	O_3	O_4	O_1	O_2	O_3	O_4	O_1	O_2	O_3	O_4	O_1	O_2	O_3	O_4
1	3	2	5	2	1	2	3	3	1	2	1	3	2	3	3	3	5	3	1
4	3	1	1	4	1	5	1	4	4	1	2	1	3	1	1	2	2	6	2
2	1	1	2	5	2	4	5	2	2	4	2	2	3	1	1	1	2	4	1
1	3	2	2	3	3	4	3	1	2	1	1	2	1	2	4	4	5	3	2
2	4	4	5	1	2	1	2	3	4	2	2	4	2	1	1	1	3	3	2
2	4	5	3	4	1	3	1	1	3	3	1	1	1	1	1	5	4	1	1

TABLE 5
RESULT OF TOTAL MULTI-OBJECTIVE RANKING IN NON-DMR

Ranking of Clusters	C1r	C2r	C3r	C4r	C5r
Total ranking of all objectives	11	8	7	11	12
	9	11	11	6	12
	6	16	10	7	8
	8	13	5	9	14
	15	6	11	8	9
	14	9	8	4	11

TABLE 6
RESULT OF ORDERED TOTAL MULTI-OBJECTIVE RANKING IN NON-DMR

Ranking of Clusters	C1r	C2r	C3r	C4r	C5r
Ordered Total ranking of all objectives	6	6	5	4	8
	8	8	7	6	9
	9	9	8	7	11
	11	11	10	8	12
	14	13	11	9	12
	15	16	11	11	14

ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)
On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

Table 7 shows, the final clusters represented in following terms:

- MC_1, MC_2, MC_3, MC4, MC5 - Modules in C1,C2,C3,C4 and C5
- C1r , C2r , C3r , C4r, C5r - Ranking of modules in C1, C2,C3,C4 and C5
- WC_1, WC_2, WC_5 - Worst modules removed from C1, C2, C5 and put it in new clusters C_{11}, C_{22} and C_{55} respectively.

TABLE 7
CLUSTERS AND ITS MODULE RANKING

MC_1	C1r	WC_1 (C_{11})	MC_2	C2r	WC_2 (C_{22})	MC_3	C3r	MC_4	C4r	MC_5	C5r	WC_5 (C_{55})
1.1	6		2.1	6		3.2	5	4.3	4	5.1	8	
1.3	8		2.2	8		3.3	7	4.4	6	5.2	9	
1.5	9		2.4	9		3.8	8	4.5	7	5.4	11	
1.8	11		2.8	11		3.9	10	4.6	8	5.6	12	5.6
1.13	14	1.13	2.9	13	2.9	3.11	11	4.7	9	5.7	12	5.7
1.16	15	1.16	2.12	16	2.12	3.14	11	4.8	11	5.9	14	5.9

Clusters C3 and C4 do not have worst modules because of all modules have ranking below 12.Finally, we have 8 clusters C_1 to C_5, C_{11}, C_{22}, and C_{55} which have lowest possible coupling, highest possible cohesion within modules, good MQ ratio and zero isolated clusters (Taken 4 objectives) within the reasonable number of search and acceptable amount of computational time. Additionally, since the objectives are not specified it is flexible to add more number of objectives as per decision maker's requirements.

VII. CONCLUSIONS AND FUTURE WORK

This paper presents two algorithms for the solution of Multi-objective software module clustering which is more flexible and adoptive in nature even if the number of objectives increases. Additionally, it makes the clusters in maximum possible minimal search. In Genetic algorithm based clustering, finding the fitness value with the consideration of satisfying multiple objectives in efficient way is a challenging task. In future, I intend to continue this to reduce the difference between maximum and minimum number of modules in clusters using effective algorithm.

REFERENCES

[1] B. H. Liskov, "A design methodology for reliable software systems," in Proc. AFIPS '72, 1972, P.191-199.

[2] M. Harman, RM. Hierons, M. Proctor, "A new representation and crossover operator for search-based optimization of software modularization," in *Proc*. GECCO, 2002.

[3] S. Mancoridis, B.S. Mitchell, Y-F. Chen, and E.R. Gansner, "Bunch: A clustering tool for recovery and maintenance of software system structures," in Proc. IEEE Int'I conf. Software maintenance, 1999, p.50-59.

[4] A. Mohammed, A. Basel, S.Hamzeh Eyal Salman, Imad Salah, "*Using Genetic algorithm as test data generator for stored PL/SQL program units,*" Journal of software engineering and applications, vol. 6, pp. 65-73, Feb. 2013.

[5] K. Praditwong, M. Harman, Xin Yao, "*Software module clustering as a multi-objective search problem,*" Software engineering, vol. 37, pp. 264-282, Mar-Apr. 2011.

[6] Ying Zhao, George Karypis, Usama Fayyad, "*Hierarchical clustering algorithms for document datasets*," Data mining and Knowledge Discovery, vol. 10, pp.141-168, Mar. 2005.

[7] T.A. Wiggerts, "Using clustering algorithms in legacy system remodularization," in *Proc*. Fourth working conference on Reverse Engineering, 1997, p. 33-43.

[8] JD. Schaffer, "Multiple objective optimizations with vector evaluated genetic algorithm," in *Proc*. International conference on Genetic algorithm and their applications, 1985.

[9] CM. Fonseca, PJ .Fleming, " Multiobjective genetic algorithms," in IEE colloquium, Genetic Algorithms for Control Systems Engineering, 1993, Digest No. 1993/130.

[10] Horn J, Nafpliotis N, DE. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," in *Proc*. The first IEEE conference on evolutionary computation, IEEE world congress on computational intelligence, 1994.

[11] P. Hajela, lin C-y, "Genetic search strategies in multicriterion optimal design," in *Proc*. Struct Optimization, 1992, 4(2) p. 99–107.

[12] T .Murata, H. Ishibuchi, "MOGA: multi-objective genetic algorithms,"
in *Proc*. IEEE International conference on evolutionary computation, 1995.

[13] E. Zitzler, L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," in *Proc*. IEEE Trans Evol Comput, 1999, 3(4), p.257–71.

[14] JD. Knowles, DW. Corne, "Approximating the nondominated front using the Pareto archived evolution strategy," in *Proc*. Evol Comput, 2000, 8(2), p.149– 72.

[15] D. Corne, NR. Jerram, J. Knowles, J. Oates, "*PESA-II*: region-based selection in evolutionary multiobjective optimization," in *Proc*. Genetic and evolutionary computation conference (GECCO-2001), 2001.

[16] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, "A fast and elitist multiobjective genetic algorithm *NSGA-II*," in *Proc*. IEEE Trans Evol Comput , 2002, 6(2)p. 182–97.

[17] R. Sarker, K-H. Liang, C. Newton, "A new multiobjective evolutionary algorithm," in *Proc*. Eur J Oper Res, 2002, 140(1), p. 12–23.

[18] GG. Yen, H. Lu, "Dynamic multiobjective evolutionary algorithm: adaptive cell-based rank and density estimation," in *Proc*. IEEE Trans Evol Comput, 2003, 7(3), p. 253–74.

[19] Urlik Brandes, Macro Gaertler, Dorothea Wagner, "*Experiments on graph clustering algorithms,*" Lecture notes in computer science, vol 2832, pp.568-579, 2003.

[20] D. E. Goldberg, *Genetic Algorithms: in search, optimization & Machine learning*, Addition Wesley, 1989.

[21] L. Davis (Ed.), *Handbook of Genetic algorithms*, Van Nostrand Reinhold, New York, 1991.