ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 1, Issue 10, December 2012

SOFTWARE MODULE CLUSTERING USING SINGLE AND MULTI-OBJECTIVE APPROACHES

CHANDRAKANTH P[1] **ANUSHA G[2]** **KISHORE C[3]**
[1,3]Department of Computer Science & Engineering, YITS, India
[2]Department of Computer Science & Engineering, AITS, India

Abstract--- *Most of the interesting software systems are very large and complex which is difficult to understand their structure. Complexity occurs due to having entities that depend on each other in intricate ways in source code. During the maintenance, the structure of the software system tends to change. Due to this reason the software engineer once understands a system's structure; it is difficult to preserve this understanding. Research into the software clustering problem has proposed several approaches to deal with the above issue by defining techniques that partition the structure of a software system into subsystems (clusters). In previous, Single-Objective approach was developed to solve the above technique. But, it was not produced optimal results. So, we proposed Multi-Objective approach for producing optimal results than the single-objective approach.*

Index terms--- *Clustering, Single-Objective, Multi-Objective, Pareto optimality.*

1. INTRODUCTION

Software Module Clustering is a one of the challenging tasks in software engineering. We all know that a well modularized software system is easy to develop and easy to maintain. A good module software structure is regarded as one that has a high cohesion and low coupling. There are different approaches are there for software module clustering.

In previous work on software module clustering we used single objective approach. Single objective is taken as, integrating the twin objectives of high cohesion and low coupling i.e., Modularization Quality (MQ). They used a search based optimization algorithm for single objective approach is hill-climbing algorithm [3]. There is somewhat difficult to achieving high cohesion and low coupling when defining module boundaries. Therefore, any attempt to conflate cohesion and coupling into a single objective may yield suboptimal results.

This paper introduces multi-objective approach for software module clustering, presenting results that show how this approach can yield superior results to those obtained by the single-objective formulation.

The rest of this paper is organized as follows: Section 2 describes how the software modules clustering process will be done. Section 3 describes single-objective search i.e., hill-climbing algorithm. Section 4 introduces multi-objective approach for software module clustering. Section 5 presents the results, while section 6 concludes.

2. SOFTWARE MODULE CLUSTERING

Software module clustering process is shown in figure 1. The first step in the software clustering process is to extract the module-level dependencies from the source code and store the resultant information in a database by using source code analysis tool. After all of the module level dependencies have been stored in a database, a script is executed to query the database, filter the query results, and produce, as output, a textual representation of the Module Dependency Graph (MDG). We define MDGs formally, but for now, consider the MDG as a graph that represents the modules (Classes) in the system as nodes, and the relations between modules as weighted directed graph.

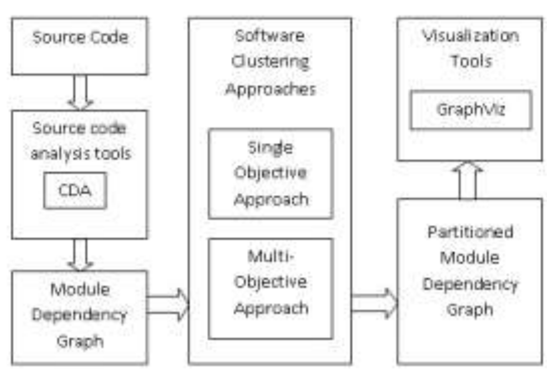

Figure 1: Software Module Clustering Process

Once the MDG is created, apply Multi-Objective approach to the MDG and generates partitioned MDG. The clusters in the partitioned MDG represent subsystems, where each subsystem contains one or more modules from the source code. After the partitioned MDG is created, we use graph drawing tools such as Graphviz (dotty) to visualize the results.

In previous single-objective approach was used. In that only one objective was used i.e., Modularization Quality (MQ). It is the combination of both high cohesion and low coupling. But single-objective approach produces suboptimal results. So we go for multi-objective approach, here multiple objectives are used for clustering the software modules for getting superior results than that of the single-objective approach. The multi-objective approach produces maximum number of clusters.

3. Single-Objective Approach

Hill-Climbing clustering algorithm [3] is the single-objective search algorithm. It starts with a random partition of the MDG. Modules from this partition are then systematically rearranged in an attempt to find an “improved" partition with a higher MQ. If a better partition is found, the process iterates, using the improved partition as the basis for finding even better partitions. This hill-climbing approach eventually converges when no additional partitions can be found with a higher MQ.

Hill-Climbing algorithms move modules between the clusters of a partition in an attempt to improve MQ. This task is accomplished by generating a set of neighboring partitions (NP). A partition NP and P are neighbor with each other if N only if NP and P are same except the single cluster of a partition P is in a different cluster in partition NP. If partition P contains m nodes and k clusters, the total number of neighbors is O(n. k). It should be noted that for many partitions of an MDG the number of neighbors is exactly n.k. However, if a partition contains clusters with 1 or 2 nodes, the total number of distinct neighbors is slightly less. There is a limitation in Hill-Climbing clustering algorithm i.e., it is not practical to use with the systems that have more than 15 modules.

4. Multi-Objective Approach

Here we proposed the first pareto optimal multi-objective formulation of automated software module clustering, presenting results that show how this approach can yield superior results to those obtained by the single objective formulation.

A. *Multi-Objective Approach*

The aim of Multi-Objective Approach is to capture the attributes of a good clustering. It will have maximum possible cohesion and minimal possible coupling. But it should not put all the modules into a single cluster and not produce a series of isolated clusters.

The objectives of MCA are as follows:

- Cohesion (Maximizing)
- Coupling (Minimizing)
- Number of Clusters (Maximizing)
- Modularization Quality (Maximizing)
- Number of Isolated Clusters (Minimized)

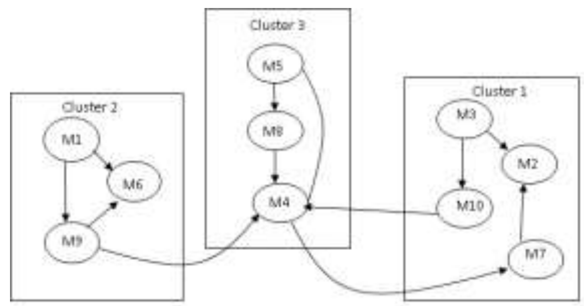

Figure 2: The module dependency graph after clustering by Multi-Objective Approach

To illustrate the Multi-Objective Approach [1], consider the MDG in fig. 2. The objective values are as follows:

- The sum of intra-edges of all clusters: 9
- The sum of inter-edges of all clusters: -6
- The number of clusters: 3
- Modularization Quality: 2.014836
- The number of isolated clusters: 0

5. Results

In experimental study, the algorithm is applied to 17 different Module Dependency Graphs (MDGs). The modules in MDGs are varying from 20 to 100. There are two types of MDGs are taken for experiment. The first type is weighted MDGs and second type is unweighted MDGs shown in table 1. In unweighted MDG graphs, an edge denotes a unidirectional method or a variable passed between two modules, where as weighted MDG graph is assigned by considering the number of unidirectional method or variables passed between two modules; the greater the weight, the more dependency between two modules. Table 1 presents details of the subject MDGs studied. These systems are not necessarily “degraded” systems in terms of their modular structure, but they have been studied widely by other researches to evaluate their algorithms for module clustering and so they denote reasonable choices for comparison.

Table 1: The systems studied

	Name	Nodes	Edges	Description
unweighted	mutunis	20	57	An operating system for educational purposes written in the turning language
	ispell	24	103	Software for spelling and typographical error correction in files
	rcs	29	163	Revision Control System used to manages multiple revisions of files
	bison	37	179	General-purpose parser generator for converting grammar description into c programs
	grappa	86	295	Genome rearrangement analyzer under parsimony and other phylogenetic algorithms
	bunch	116	365	Software Clustering tool (Essential java classes only)
	incl	174	360	Graph drawing tool
weighted	icecast	60	650	Streaming media server based on the MP3 audio codec
	gnupg	88	601	Complete implementation of the OpenPGP Internet standard
	inn	90	624	Unix news group software
	bitchx	23	729	Open source IRC client
	xntp	111	729	Time synchronization tool
	exim	23	1255	Message transfer agent for use on Unix systems connected to the Internet
	mod_ssl	135	1095	Apache SSL/TLS Interface
	ncurses	138	682	Software for display and update of text on text-only terminals
	lynx	23	1745	Web browser for users on UNIX and VMS platforms
	nmh	198	3262	Mail client software

Table 2: Comparison of algorithms by taking MQ as assessment criterion.

	Name	Single-Objective Approach	Multi-Objective Aproach
unweighted	mutunis	2.249	2.314
	ispell	2.337	2.339
	rcs	2.218	2.239
	bison	2.639	2.648
	grappa	12.676	12.578
	bunch	13.536	13.455
	incl	13.568	13.511
weighted	icecast	1.779	2.654
	gnupg	4.869	6.905
	inn	6.720	7.876
	bitchx	2.465	4.267
	xntp	6.655	8.168
	exim	5.199	6.361
	mod_ssl	7.906	9.749
	ncurses	9.836	11.297
	lynx	3.488	4.694
	nmh	7.012	8.592

This section presents the results of the experiment that compare the MQ value obtained for two approaches, Single-Objective Approach and Multi-Objective Approach described earlier in previous section. i.e., the results assess how well the Multi-Objective Approach performs when compared with Single-Objective Approach using MQ value.

Table 2 presents the results comparing Multi-Objective Approach with Single-Objective Approach. There is strong evidence to suggest that our approach is outperformed the Single-Objective approach for weighted and unweighted MDG problems. For unweighted MDG problem, Multi-Objective approach gives higher values for MQ in 4 from 7 problems. However for weighted MDG problem, Multi-Objective approach gives higher values for MQ in 10 out of 10 problems.

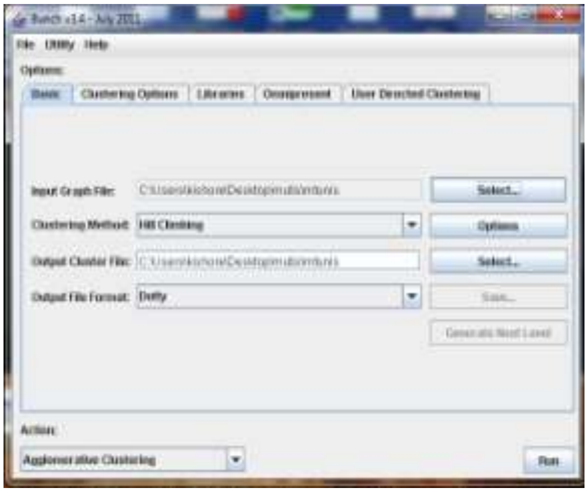

Figure 3: Applying "Hill-Climbing" as Clustering Method to mtunis system

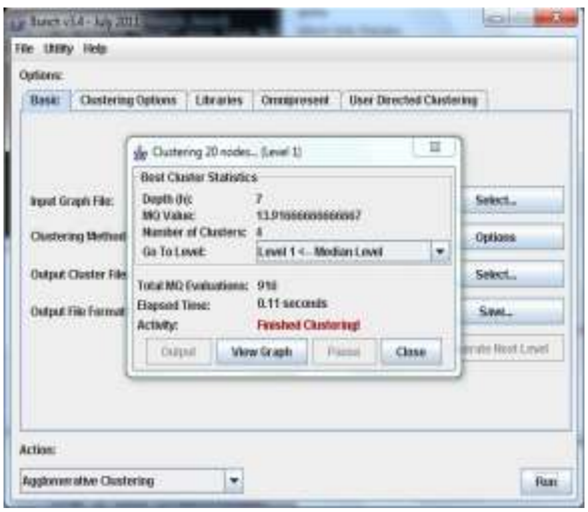

Figure 4: Finished status of hill-climbing algorithm

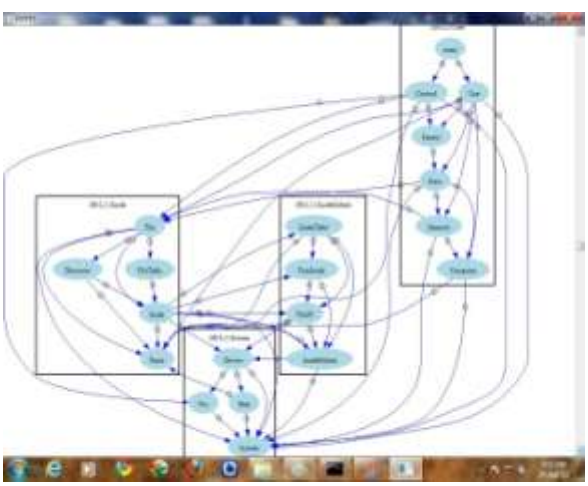

Figure 5: Clustered MDG of mtunis system using Single Objective Approach

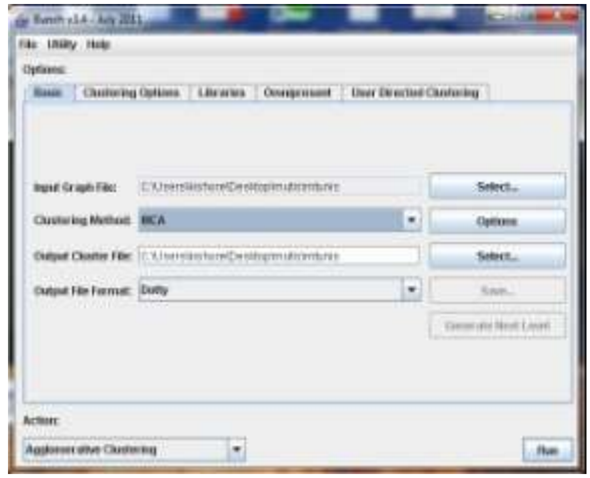

Figure 6: Apply "MCA" as Clustering Method to mtunis system

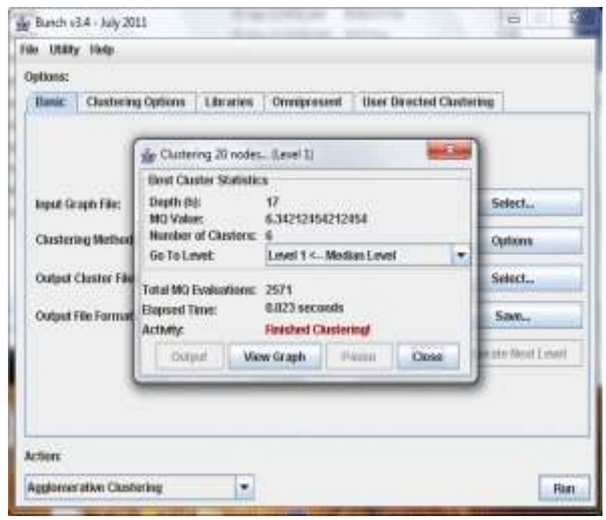

Figure 7: Finished status of Multi-Objective Approach

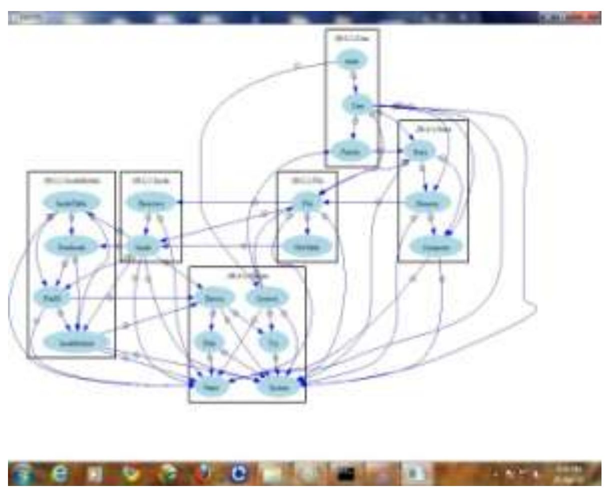

Figure 8: Mtunis system applying Multi-Objective Approach

6. CONCLUSION

This paper introduces multi-objective approach to software module clustering. The results obtained by applying this approach on 17 real-world clustering problems are compared with single objective approach. The results indicate that multi-objective approach is able to produce better solutions than the existing single-objective approach.

The multi-objective approach lends itself to extensions by considering other possible objectives with respect to which modularization takes place. Future work could consider such additional objectives for better modularization.

REFERENCES

[1] Kata Praditwong, Mark Harman, and Xin Yao, "Software Module Clustering as a Multi-Objective Search Problem", IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 37, NO. 2, MARCH/APRIL 2011

[2] S. Mancoridis, B.S. Mitchell, C. Rorres, Y.-F. Chen, and E.R. Gansner, "Using Automatic Clustering to Produce High-Level System Organizations of Source Code," Proc. Int'l Workshop Program Comprehension, pp. 45-53, 1998.

[3] L.L. Constantine and E. Yourdon, Structured Design. Prentice Hall, 1979.

[4] K. Mahdavi, M. Harman, and R.M. Hierons, "A Multiple Hill Climbing Approach to Software Module Clustering," Proc. IEEE Int'l Conf. Software Maintenance, pp. 315-324, Sept. 2003.

[5] S. Mancoridis, B.S. Mitchell, Y.-F. Chen, and E.R. Gansner, "Bunch: A Clustering Tool for the Recovery and Maintenance of Software System Structures," Proc. IEEE Int'l Conf. Software Maintenance, pp. 50-59, 1999.

[6] R. Pressman, Software Engineering: A Practitioner's Approach, third ed. (European adaptation (1994), adapted by Darrel Ince). McGraw-Hill, 1992.

[7] B.S. Mitchell and S. Mancoridis, "Using Heuristic Search Techniques to Extract Design Abstractions from Source Code," Proc. Genetic and Evolutionary Computation Conf., pp. 1375-1382, July 2002.

[8] Sommerville, Software Engineering, sixth ed. Addison-Wesley, 2001.

[9] D. Doval, S. Mancoridis, and B.S. Mitchell, "Automatic Clustering of Software Systems Using a Genetic Algorithm," Proc. Int'l Conf. Software Tools and Eng. Practice, Aug.-Sept. 1999.

[10] B.S. Mitchell and S. Mancoridis, "On the Automatic Modularization of Software Systems Using the Bunch Tool," IEEE Trans. Software Eng., vol. 32, no. 3, pp. 193-208, Mar. 2006.

[11] M. Harman, S. Swift, and K. Mahdavi, "An Empirical Study of the Robustness of Two Module Clustering Fitness Functions," Proc. Genetic and Evolutionary Computation Conf., pp. 1029-1036, June 2005.

[12] Kishore C, Srinivasulu Asadi, Anusha G, "Comparative Study of Software Module Clustering Algorithms: Hill-Climbing, MCA And ECA", IJARCET, Volume 1, Issue 3, 189-195, May, 2012.

[13] Kishore C, Asadi Srinivasulu, "Multi-Objective Approach for Software Module Clustering", IJARCSSE, Vol 2(3), 2012, 266-271, March, 2012.

[14] Kishore C, Srinivasulu Asadi, "Comparative Study of Single and Multi-Objective Approaches of Software Module Clustering" Proc. Nati Conf. Computational Intelligence and Networking Technologies, May 2012.

AUTHORS BIOGRAPHY

Chandrakanth P received the B.Tech Computer Science & Engineering from JNTUA, Anantapur, India in 2009 and received Master's degree in Computer Science & Engineering from the JNTUA, Anantapur, India in 2011. His research areas are Software Engineering, Data warehousing and Data Mining, Database Management Systems, Computer Networks and Cloud Computing.

Anusha G received the B.Tech Information Technology from JNTUA, Anantapur, India in 2010 and received the Master's degree in Software Engineering from the JNTUA, Anantapur, India in 2012. Her research areas are Software Engineering, Data warehousing and Data Mining, Database Management Systems and Cloud Computing. she has published 5 papers in International Journals and Conferences. Some of her publications appear in IJARCET and IJARCSSE digital libraries.

Kishore C received the B.Tech Information Technology from JNTUA, Anantapur, India in 2010 and received his Master's degree in Software Engineering from the JNTUA, Anantapur, India in 2012. His research areas are Software Engineering, Data warehousing and Data Mining, Database Management Systems and Cloud Computing. He has published 7 papers in International Journals and Conferences. Some of his publications appear in IJCSIT, IJARCET and IJARCSSE digital libraries.