International Journal of Information and Computation Technology.
ISSN 0974-2239 Volume 4, Number 4 (2014), pp. 403-408

http://www. irphouse.com /ijict.htm

Clustering - A New Perspective

K. Sarojini

Department of Information Technology, SIES College, University of Mumbai, Sion (West), Mumbai, Maharastra, India.

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, after the system had gone through many changes during the maintenance, remembering the system's structure is less possible one. Software architecture is a model of the software system expressed at a high level of abstraction. The architectural view of a system raises the level of abstraction and concentrating on only ‘black box’ elements. Software module clustering technique is a key to create a clear view about those abstractions. It follows the emergent of Multi-objective search-based optimization techniques which yield accurate objective based clustering successfully. In this paper, I am going to propose two algorithms, one as a search optimization technique and another as a multi-objective fitness evaluation function of that search technique. As fitness function is a component of search based Genetic algorithms, I have embedded one algorithm within another.(Abstract)

Keywords: Optimization;intra-cluster similarityt; metaheuristic; modularization quality; cohesion; coupling.

1. Introduction

Clustering is a process of partitioning a set of datainto a set of meaningful sub-classes, called clusters. It helps the users to understand the natural grouping structure in data set.The goal of clustering is to maximize the intra-cluster similarity and minimize the inter-cluster similarity.Optimization technique is a key to optimize thefittest solutions(highly similar solutions) in each cluster.Optimization techniques should consider many complicated factors to optimize the solutions.One such factor ismultiple decision variables (multiple objectives).Optimizing x with respect to a single objective often results in unacceptable solutions with respect to other objectives.So optimization algorithms should adopt multi-objectives. As simply says success of product will not rely on one confined factor instead define more factors such as good quality, low cost, etc. Another factor that may add to the difficulty of solving a problem is the complex nature of the relationships between the decision variables and the associatedoutcome. For example,though increasing the quality of a product and decreasing the cost a product are inversely proportional to each other; both should be attained for profit-maximization. A third complicating factor is the possible existence of one or more complex constraints on the decision variables. So aperfect multi-objective solution that simultaneously optimizeseach objective function is almost impossible. Areasonable solution to a multi-objective problem is toinvestigate a set ofsolutions, each of which satisfies theobjectives at an acceptable level without being dominatedby any other solution. I have proposed such a type of algorithm which is elaborated in the rest of the sections.

2. Research Elaborations

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 system for educational purposes written in the Turing Language) to explain the execution of the algorithm.

Similarity based Encoded-Emergent Algorithm and Metaheuristic weighted ranking algorithm

Similarity based Encoded-Emergent Algorithm

1. Let total number of clusters is t and labeled as C1 to Ct. Total number of modules is tm. Each cluster has n modules. Each module is uniquely identified by Mi; i=1 to tm;
2. Mi = Vi .Bi where Vi isthe module number and Bi is binary value of the module number.
3. Number of digits d in binary value depends on tm in the following way.
4. If 22>= tm >=20, d= 2; If 23>=tm >= 22+1, d= 3; If 24>=tm >= 23+1, d=4….
5. Metaheuristic Weighted Ranking Algorithm
6. Metaheuristic Search to find the existence of high level objectivein each Bi.

Step 1: Select Bi.ofMi (i=1 to tm) and check number of 1’s in Bi. Let it beobjj(j=1 to t) .

Step 2: Assign j as an index of Ct.

Step 3: Label the Mi in the form of Vi. Bi and put it under the cluster index Ct. (Here‘t’ denotes number of 1’s in the Binary value Bi of Module Mi.

Step 4: Repeat step 1 to step 3 till i> tm.

Multiple objectives Weighted-Ranking of each index.

Step 1: Clusters are termed as Ct where t =1 to Total number of clusters.
Objectives are Oj where j = 1 to number of objectives.
Modules are Mnwhere n= 1 to number of modules in a cluster
Weight of each module is Wn
Calculate the weight (Wn) of each module (Mn) in cluster Ct with respect to objective Oj .

Step 2: Rank the modules in Ct based on Wn. The module which has high weightage value gets lower ranking number which means that is the fittest Mn to stay in its native Ct .

Step 3: Total Ranks of each Mnwith respect to all the objectives are sum of all the ranks.

Step 4: Modules which have Worst ranking number should be shifted from its native cluster to new cluster
which islabeled as Mnn. The Range ofworst ranking number is defined by explicit function.

Step 5: Increment the Ct. by 1(select the next cluster) and repeat Step 1 to Step 4 till Ct.>t.

Table 1 shows initial module clusters ofmtunis software. Ithad 5 clusterslabeled in decimal numbers, the order in which it was created. I have used metaheuristicsearch as a higher-level proceduredesigned to find a lower-level multiple objectives which speed up the process of finding a satisfactory solution via mental shortcuts to ease the cognitive load of making a decision. Our algorithm starts with encoding of each module. The encoded string hastwo sections V_i andB_i . V_i denotes module number in decimal and B_i is the binary equivalentof the module number.

For example in Table 2, M2 is encoded as 2.00010(V_i.B_i). In this string 2 denotes module number and 00010 is a binary equivalent to the module number. Number of digits in B_i is based on total number of modules ‘tm’. For example if tm is 30, we need minimum 5 digits to assign binary value of 30. In this caseB_i ofall modules will be encoded with 5 digits binary number.

Table 1: Initial clusters of mtunis software.

	CLUSTERS				
	C1	**C2**	**C3**	**C4**	**C5**
MODULES	M2	M1	M4	M8	M16
	M5	M6	M3	M9	M10
	M12	M17	M18	M20	M24
	M7	M11	M13	M14	M19
	M21	M22	M25	M26	M28
	M15	M23	M27	M29	M30

Table 2: Modules are in encoded form.

	CLUSTERS				
	C1	**C2**	**C3**	**C4**	**C5**
MODULES	2.00010	1.00001	4.00100	8.01000	16.10000
	5.00101	6.00110	3.00011	9.01001	10.01010
	12.01100	17.10001	18.10010	20.10100	24.11000
	7.00111	11.01011	13.01101	14.01110	19.10011
	21.10101	22.10110	25.11001	26.11010	28.11100
	15.01111	23.10111	27.11011	29.11101	30.11110

In table 3, columns C_1, C_2,C_3, C_4shows the result of the metaheuristic search by keeping the number of I's in B_i as a high level abstraction. This gives index to each cluster and the index value is based on number 1's (O_j) in $B_{i.}$ In this algorithm objectives are dynamic which can be separately derived based on some constraints or denotesdirectly in the encoded string. I explained one of the objectives which express the similarity in the form of position of 1's in the encoded string. Most significant bit of B_i is assigned lowest weight (here 2) and it will increase by 2 towards least significant bit position. Least significant bit will have highest weight. Weight is calculated by adding up theweight of the position of 1. For example in Table 3, index 2 has module 5.00101 which is in the form of V_i .B_i.Weight assigned for each bit is in 2+4+6+8+10 sequence. The bit only which has value '1'gets weight based on its respective position. So $W_n$of 00101 is calculated as 0+0+6+0+10=16.

Table 3: Columns C_1 to C_4 showsthe result of Similarity based Encoded-Emergent Column $W_n$andR_n shows the result of weightage based Ranking with respect to single objective inMetaheuristic weighted-Rankingalgorithm.

C1 (Index 1)	WN	RN	C2(Index 2)	WN	RN	C3(Index 3)	WN	RN	C4(Index4)	WN	RN
2.00010	8	2	5.00101	16	2	7.00111	24	1	15.01111	28	1
1.00001	10	1	6.00110	14	3	11.01011	22	2	23.10111	26	2
4.00100	6	3	3.00011	18	1	13.01101	20	3	27.11011	24	3
8.01000	4	4	9.01001	14	3	14.01110	18	4	29.11101	22	4
16.10000	2	5	10.01010	12	4	19.10011	20	3	30.11110	20	5
			12.01100	10	5	21.10101	18	4			
			17.10001	12	4	22.10110	16	5			
			18.10010	10	5	25.11001	16	5			
			20.10100	8	6	26.11010	14	6			
			24.11000	4	7	28.11100	12	7			

Table 4 shows ranking of each module individually with respect to each objective (here 3 objectives are O_1O_2 and O_3) and total ranking value (TR_1 to TR_4)of each module.The worst ranking number is >=13 which is calculated by explicit function. The modules which have ranking number greater than 12 are worst ranking modules and those modules are not fit (because of having less similarity with rest of the modules in the cluster) to stay in the same cluster.

Table 5 shows the new such clusters C_{22} andC_{33}. Modules in these clusters have good cohesion ratio among themselves but poor cohesion ratio with the modules in cluster $c_2c_3$respectively.

Table 4: Total ranking based on multi-objectives inMetaheuristic weighted-ranking optimization algorithm.

C1					**C2**					**C3**					**C4**				
MN	**O1**	**O2**	**O3**	**TR1**	**MN**	**O1**	**O2**	**O3**	**TR2**	**MN**	**O1**	**O2**	**O3**	**TR3**	**MN**	**O1**	**O2**	**O3**	**TR4**
2.00010	2	5	2	9	5.00101	2	5	1	8	7.00111	1	8	3	12	15.01111	1	2	5	8
1.00001	1	2	4	7	6.00110	3	7	5	15	11.01011	2	5	8	15	23.10111	2	1	2	5
4.00100	3	1	3	7	3.00011	1	8	6	15	13.01101	3	1	4	8	27.11011	3	3	4	10
8.01000	4	4	1	9	9.01001	3	1	4	8	14.01110	4	2	8	14	29.11101	4	3	1	8
16.10000	5	3	2	10	10.01010	4	3	3	10	19.10011	3	6	1	10	30.11110	5	4	3	12
					12.01100	5	5	2	12	21.10101	4	4	3	11					
					17.10001	4	6	3	13	22.10110	5	7	6	18					
					18.10010	5	2	7	14	25.11001	5	8	1	14					
					20.10100	6	2	6	14	26.11010	6	2	3	11					
					24.11000	7	4	6	17	28.11100	7	3	7	17					

Table 5: Resultant Table derived from algorithms.

	CLUSTERS					
	C1	**C2**	**C22**	**C3**	**C33**	**C4**
MODULES	M2	M5	M18	M13	M14	M23
	M1	M9	M3	M19	M25	M15
	M4	M10	M20	M21	M11	M29
	M8	M12	M6	M26	M28	M27
	M16	M17	M24	M7	M22	M30

3. Conclusions

This paper has presentedtwo 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 clusters the modules in maximum possible minimal search. In clustering algorithm, finding modularization quality is based on factors like coupling between clusters, cohesion among the modules which are in same cluster and number of clusters, etc.The above elaborated algorithms finds a reasonable solution to a multi-objective problem by investigating a set ofsolutions, each of which satisfies theobjectives at an acceptable level without being dominatedby any other solution.

References

[1] K. Praditwong, M. Harman, Xin Yao(2011), Article in Software engineeringconference, "*Software module clustering as a multi-objective search problem*," vol. 37, pp. 264-282.

[2] Ying Zhao, George Karypis, Usama Fayyad(2005), Article in Journal, Data mining and Knowledge Discovery "*Hierarchical clustering algorithms for document datasets*,", vol. 10, pp.141-168.

[3] Horn J, Nafpliotis N, D.E. Goldberg (1994), Article in conference, *Proc.IEEEIntl. Conf.* on evolutionary computation, IEEE world congress on computational intelligence, "*A niched Pareto genetic algorithm for multiobjective optimization*".

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