Indian Journal of Science and Technology, *Vol 9(48), DOI: 10.17485/ijst/2016/v9i48/98093, December 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Rule based Classification of Graph Theory Concepts by Use Case Analysis

Anuja Bokhare[1]*, P. S. Metkewar[1] and R. S. Walse[2]

[1]Symbiosis Institute of Computer Studies and Research (SICSR), A Constituent of Symbiosis International University (SIU), Atur Centre, Gokhale Cross Road, Model Colony, Pune - 411016, Maharashtra, India; anuja.bokhare@sicsr.ac.in

[2]College of Dairy technology, Pusad, Affiliated to MAFSU, Nagpur - 440006, Maharashtra, India; pravin.metkewar@sicsr.ac.in, rajeshwalse@gmail.com

Abstract

Objective: Nature of the problem is always solvable, partially solvable or unsolvable but by using certain techniques we may resolve uncertainty up to some extent. This study evaluates Graph Theory (GT) concepts, in order to resolve its complexity by applying use case analysis method and helps to classify them. **Methods/Statistical Analysis:** Experiment has been formulated on 38 GT concepts. For each GT concept identification of use cases and its corresponding activities is performed. Further, proposed method helps in classifying the problem. **Findings**: In this paper rule based random sampling technique for use case analysis is being proposed. It helps to compute required number of use cases for solving graph theory related problems and to categorize them into simple, moderate or complex classes. In order to achieve this, proposed work deals with identifying use cases, activities in each use case, classification of activities in terms of simple, moderate and complex classes. **Novelty/Improvement:** Computation of problem length (PL) through proposed rule based random sampling helps in classification of problem. Classifying the problem helps to reduce its complexity. Proposed classification method/process achieves the same.

Keywords: Graph Theory (GT) Concepts, Rule Based Random Sampling, Use Cases

1. Introduction

A Graph is formed by vertices and edges connecting the vertices. Graph theory originates in 1736 to find a solution for the problem of Königsberg Bridge. The solution was given by Eular as a concept of Eulerian Graph. Afterwards, the idea of complete graph and bipartite graph given by A.F Mobius. These graphs are planar has been proved by Kuratowski[1]. In 1845 Gustav Kirchhoff implemented tree which is a kind of graph with no cycle. Electric current calculation is one of the application of this concept. William R. Hamilton and Thomas. P. Kirkman has invented the Hamiltonian graph in 1856, to visit certain sites exactly once revising the trips. The four color problem was identified by, Thomas Gutherie in 1852. Wolfgang Haken and Kenneth Appel have resolved the problem after a century. From there on wards time was considered as a start of Graph Theory concept. Heinrich solved four color problems in 1969 using computers[2].

A use case consists of description of sequence of actions. It consists of different sequences as well error sequences, which generally occurred when system or subsystem interact with external actor. A use case approach is a means of documenting functional requirements

*Author for correspondence

which further guide how to discover what classes will be required for implementing that functionality. We can use textual description along with diagram for explaining the functional requirement. A function is a procedure or routine defined to perform a specific task. It is a group of statements that together perform a task.

Use cases are used for identifying activity or task of functionality or scenario. Therefore, the use case model may be used for predicting problem size at early stage of development. This paper focuses on use case analysis of Graph Theory concepts by categorizing them into simple, moderate and complex.

The structure of this paper is as follows. First some previous work in the area of Use Case analysis, its related metrics and their improvement are reviewed (Section 2). Then in the further section Use case analysis of GT concepts is explained from identification of use cases, activities in each identified use cases and their classification through rule based sampling (Section 3). Section 4 is devoted to conclusion and finally, Section 5 talks about future scope

2. Previous Work

Authors[3] have observed that, requirements of different participants in object-oriented software development are often compiled with the help of use case models which is based on dynamic and functional requirements. An unwavering requirements specification in view of high level design has been prepared by analyzing and integrating these requirements. After reviewing higher level design, team has come up with lower level design specifications. They have proposed a prescribed clarification related to use case representation based on unified modeling language (UML), use case, collaboration and activity and state transition diagrams. This formulation is developed using graph theory concept. Authors have presented[4] an extension of the traditional use case analysis i.e. a concept of role analysis. Based on given use cases and technical constraints the role analysis algorithm creates required images of actors. Users, activities and its relationships in other use cases are been determined with the help of algorithm. It can be useful in analysis and design of any application software, web enabled system, utility packages, plug-ins, frameworks, tools etc. Authors have focused on Structured analysis[5] is an extensively used as a system modeling technique which helps to or understand actual systems before they are constructed. In structural analysis user has to concentrate on ER modeling, context level diagram but as far as object oriented analysis is concerned, they need to focus on use case modeling. They have proposed for to structured analysis a use case driven approach. The functionality of a system is identified easily for model builders by using this approach

Researcher have focused on the disadvantages of Service Oriented Analysis and Design i.e SOAD methodologies[6,7]. A new approach presented is related to SOAD method that has been formed by the concept of use case. Service recognition from use case writing is facilitated through a service model. This method helps to attain better quality service recognition standard to an analyst work. Finally, they had concluded that the technique is best as compared to the one based on business process modeling. Software engineering approaches[8] may give a good support for closing the gap between the problem domain and solution and due to this it decreases the failure risk in software projects. During requirement phase, the most appropriate technique for analyzing, classifying the software development related problem is the, making use of problem Frames (PF). However, PF still don't have good tools and technique which help in integration of various phases in software development. Researchers have presented a technique of converting these problem frames in mostly used UML diagrams. The technique is being verified and explained through the case study approach. In this paper, authors have initially, examine the use of the function point analysis[9] in the software measurement; later explain the information detailed illustration about method of production rules. At last, proposes the data model of the function point analysis for software size measurement. It also investigates the technical complexity factors and their weights. Knowledge representation has been carried on function point analysis. It has been concluded that human thinking can be simulated for problem solving through knowledge representation.

Authors have focused on various software development effort estimation models[10] used for identifying relationship between software effort and size. Data used during project analysis is incomplete, inconsistent most of the time. Software schedule effect and software effort affect the computation of budget. Incomplete estimations about project give a lot of problem, mainly the delay in project completion. Finally, research conclusion shows that the proposed model gives good results than the existing models. Albrecht[11,12] has developed a method of function point analysis for measuring the size of a computerized system. Author has focused on the areas where partial alternative solution can be given for problems identified. Based on the validity and applicability of function point analysis conclusions are been derived.

Author has focused on lines of code (LOC)[13] or size of LOC, a simple method for measuring effectiveness of a function of input effort and output size. Generally, productivity about software is attached with a good approximation of software size.

However, the final release doesn't include modified code[14]. Practical calculation[15] to the EVI evaluation model is done through quantification. It helps to determine the detailed EVI quantized value. This paper[16] explains an approach for quantifying reliability and tracking, growth and prediction (TGP) model. The presentation of performance parameter is explained in 2 ways; using past knowledge or by individual evaluation when data is not available. It is observed that a updated TGP model provides an approximate of the risk caught up in attaining good reliability. It is based on failure rate parameters and the performance parameter. There is significant use of Graph theory[17] in multiple fields like biochemistry, Electrical Engineering, Mechanical Engineering, Computer Science and operation research. A review on theory of graphs is presented to focus the methodology used in representing graph. In the article[18], proposed ratio estimator under rank set sampling found to be more proficient than simple random sampling which uses the linear combination of Median and Co-efficient of skewness. Modes like COCOMO, FPA, LOC etc are widely used for effort estimation. But these techniques have some limitations i.e., experts are required for counting function points.

3. Use Case Analysis of GT Concepts

In this section Use case analysis of the GT concepts have been carried out. Use Case based analysis may require some understanding of knowledge of domain, system size etc. First an identification of use case is explained, and then rules are defined to classify the activities. Identification of number of activities in identified use cases is brought in (Subsection 3-b). Further, classification of activities (Subsection 3-c) is explained. Rule base sampling (Subsection 3-e) is applied after performing experimental analysis (Subsection 3-d) and results are shown in Tables 1, 2 and 3 for simple, moderate and complex classes of GT concepts.

Table 1. Weight factor

Type of class	Weight
Simple (w_1)	1
Moderate (w_2)	2
Complex (w_3)	3

3.1 Identification of use cases

For each Graph Theory concepts, we have drawn use case diagram along with its description. With the help of this strategy it is easy to identify the use cases. Here uses case would be a smaller activity or activities within activity, or a smaller task or an event or a functio nality etc.

3.2 Activities in Each Identified Use Cases

A use case may have number of activities. Typical model of Bipartite GT concept has been discussed here. Following set represents use case number, activities in each use case number {1-3, 2-5, 3-5 and 4-8} respectively.

3.3 Classification of Activities

As far as analysis of total number of GT concepts have been considered, it has been observed that there are not more than 9 activities in respective use case. On the basis of above analysis, following rules have been formulated. They are

R1: Simple class means which consists of <= 4 activities and its ranges.
R2: Moderate class means which consists of 5 to 7 activities and its ranges.
R3: Complex class means >= 8 activities and its ranges.

These classes are defined as follows:

i. *Simple class (SC)* is defined as a smaller activity or activities within activity, or a smaller task or an event which is less than or equal to 4 and whose weight factor assumed to be 1 in order to solve GT concepts.
ii. *Moderate class (MC)* is defined as number of activities associated with particular use case if it falls in between

Table 2. Computation of problem length

Sr. No	Graph parameter	Simple Σa_i	Moderate Σb_i	Complex Σc_i	Problem length PL	Sr. No	Graph parameter	Simple Σa_i	Moderate Σb_i	Complex Σc_i	Problem length PL
1	Indegree / outdegree	2+2+2	0	0	6	20	Spanning tree	2+4	7+6	0	32
2	Ecentricity of tree	2+2+4+2	0	0	10	21	Decomposition of Graph	2+2+4+2	0	8	34
3	Union	4+2	5	0	16	22	Cut vertex	2+2+2+2	0	9	35
4	Bridge	2+2	6	0	16	23	Isomorphism	2	5	8	36
5	Path	2+2+2	5	0	16	24	Product of graph	6	0	10	36
6	Directed path	2	10	0	22	25	Eulerial digraph	0	5	9	37
7	Hamiltonian path	2+2	5+5	0	24	26	Hamiltonian digraph	0	5	9	37
8	Simple digraph	2	0	8	26	27	Adjacency matrix	2+2	5	8	38
9	Asymmetric digraph	2	0	8	26	28	Directed circuit	2+2	5	8	38
10	Complete digraph	2	0	8	26	29	Edge connectivity	2+2	5	9	41
11	Balanced digraph	2	0	8	26	30	Fundamental circuit	2+2+4	5	8	42
12	Incidence matrix	3+2	5+6	0	27	31	Subgraph	2+3+4	6	8	45
13	Simple asymmetric	4	0	8	28	32	Vertex connectivity	2+2	7	9	45
14	Symmetric digraph	4	0	8	28	33	Shortest path	2+2+4	6+7+6	0	46
15	Eulerian trail	2+2	0	8	28	34	Bipartite	3	5+5	8	47
16	Intersection	4+2	0	8	30	35	Complete Bipartite	3+2	5+5	8	49
17	Hamiltonian circuit	2+4+2	5+6	0	30	36	Dijkstra's algorithm	2+2	6+6	8	52
18	Hamiltonian cycle	2+4+2	5+6	0	30	37	Eulerian circuit	2+4	0	8+8	54
19	Hamiltonian graph	2+4+2+2	5+6	0	32	38	Eulerian graph	4+4	0	8+8	56

the range of 5 to 7 activities and whose weight factor assumed to be 2 in order to solve GT concepts.

iii. *Complex class (CC)* is defined as number of activities associated with particular use case if it is greater than or equal to 8 and whose weight factor assumed to be 3 in order to solve GT concepts.

Table 3. Simple, moderate, complex class and its ranges

Class	LR	MR	HR	Min	Max
Simple (SC)	**0-3**	**4-7**	**8-10**	**0**	**10**
Moderate(MC)	0-7	8-14	15-19	0	19
Complex(CC)	0-5	6-11	12-16	0	16

3.4 Experiment Analysis

Graph theory consists of various concepts that are applicable in various domains. Graph partitioning is one of the concepts associated with Graph theory. It deals with partitioning the graph vertices into specified sizes such that minimum edges cross between sets. Multilevel Graph Partitioning (MGP) approach gives good quality partitions[12]. Other concepts are listed in Table 2.

Experiment has been formulated on 38 GT concepts by identifying use cases and its corresponding activities respectively. This process has been adopted for each GT concept independently. By applying the rules, use cases has been categorized into one of the class that may be simple, moderate or complex. There might be possibility that one or more use cases may fall in simple, moderate or complex class. Table 1 gives details about the weight factor which is to be assumed for simple, moderate or complex class. It helps to calculate the sum of number of activities in each defined class. Further, problem length has been calculated and its formula is given in equation 1. Its final computation results have been given in Table 2.

Mathematical Model of problem length formulation:
Problem length based on UC = ($a_i * w_1$) + ($b_i * w_2$) + ($c_i * w_3$)

$$PL_j = w_1 * \sum_{i=1}^{n}(a_i) + w_2 * \sum_{i=1}^{n}(b_i) + w_3 * \sum_{i=1}^{n}(c_i) \quad (1)$$

for each j, i varies from 1 to 3 (where j=1 to 38)

where, a_i=sum of activities in simple class
b_i=sum of activities in moderate class
c_i=sum of activities in complex class
PL_i=problem length for GT concepts
For $w_1 w_2$ w_3 refer Table 1.

3.5 Rule based Random Sampling of Problem Length

Table 2 has been already sorted on problem length and on the basis of that we have obtained minimum and maximum values of each simple, moderate and complex class; it is given in Table 3. Activities of each class are divided into low (LR), medium (MR) and high (HR) ranges and its corresponding range is depends upon minimum and maximum values of each class.

Here instead of using traditional random sampling method, we have proposed rule base sampling method. Rule based random sampling method is depicted in order to pick up the particular GT concept in one of the categories and which is based on set of defined rules. Rule based collection sets are identified and given in Tables 4, 5, 6 respectively.

Table 4. SC collection set of GT concepts and its PL

Sr.No	GT concept	PL
1	**Simple Digraph**	**26**
2	Asymmetric Digraph	26
3	Complete Digraph	26
4	**Balanced Digraph**	**26**
5	Isomorphism	36
6	Eulerian Digraph	37
7	**Hamiltonian Digraph**	**37**

Table 5. MC collection set of GT concepts and its PL

Sr.No	GT concept	PL
1	Shortest path	46

Table 6. CC collection set of GT concepts and its PL

Sr.No	GT concept	PL
1	Complete Bipartite	49
2	Dijkstra's algorithm	52

Set of rules are defined for rule based sampling:

Rule 1: If (SC=LR) or (MC=LR) or (CC=LR) then "SC collection set of GT concepts and itsPL".

Rule 2: If (SC=MR) or (MC=MR) or (CC=MR) then "MC collection set of GT concepts and its PL".

Rule 3: If (SC=HR) or (MC=HR) or (CC=HR) then "HR collection set of GT concepts and its PL".

Based on results obtained in Table 4, 5 and 6, GT concepts have been chosen from SC, MC and CC collection set as a sample for further quantification process and to study its influencing factors. Samples are simple digraph, balanced digraph, Hamiltonian digraph, shortest path, complete bipartite, and Dijkstra's algorithm.

4. Conclusion

In this paper, authors have worked on GT concepts in order to resolve its complexity by applying use case analysis method. Use case analysis approach help us to categorize the problem nature into simple, moderate and complex classes which is further useful to calculate the problem length. Rule base random sampling helps to categorize the problem in low, moderate and complex category, which is not yet presented anywhere. Rules are defined w.r.to problem length and defined ranges. Rule 1, 2 and 3 gives collection set of 3, 1 and 2 GT concepts for simple, moderate and complex class respectively which is being used for further processing of quantification as a part of future scope.

Identifying the number of use cases is one of the novel techniques for estimating the size of the problem and comparatively it is better than lines of code (LOC) and Function Point estimation Analysis (FPA).

5. Future Scope

GT concepts which are considered during study can be weighted through, throughput. Separate throughput model need to be built for such simple, moderate and complex categories of GT concepts. Further research can be extended to calculate the throughput of different GT concepts. Quantification technique can be used to categories the calculated throughput into different subcategories.

6. References

1. Deo N. Graph theory with applications to engineering and computer science. Prentice Hall of India; 1990.
2. Shirinivas SG, Vetrivel S, Elango NM. Applications of graph theory in computer science an overview. International Journal of Engineering Science and Technology. 2010; 2(9):4610–21.
3. Hausmann JH, Heckel R, Taentzer G. Detection of conflicting functional requirements in a use case-driven approach. Proceedings of the 24th International Conference on Software Engineering, ICSE, USA; 2002. p. 105–15.
4. Zotov V, Gao X. Role analysis: The use case analysis extension for in-depth investigation of actor properties. 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), USA; 2010. p. 649–54.
5. Mirza M, Cha SH, Jung J, Lee KJ, Youn C. Use case driven structured analysis. Proceedings of 20th International Computer Software and Applications Conference, COMPSAC. South Korea; 1996. p. 510–15.
6. Liu W, Fu M, Luo S, Zou D. Use case-based service-oriented analysis and modeling. International Conference on Internet Computing and Information Services (ICICIS), China; 2011. p. 94–6.
7. Huayou S, Yulin N, Lian Y, Zhong C. A service-oriented analysis and modeling using use case approach. International Conference on Computational Intelligence and Software Engineering, CiSE, China; 2009. p. 1–6.
8. Imam AA, Hamza HS, Moneim RA. Automated generation of use case diagrams from problem frames using formal concept analysis. Tenth International Conference on Information Technology: New Generations (ITNG); 2010. p. 233–40.
9. Ren Y, Xing T, Quan Q. Function point analysis to knowledge representation of software size measurement. International Conference on Information Science and Technology (ICIST), China; 2011. p. 122–5.
10. Kaur M, Sehra SK. Function Point analysis. International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT); 2014. p. 140–5.
11. Symons CR. Function point analysis: difficulties and improvements. IEEE Transactions on Software Engineering. 1988; 14(1):2–11.
12. Bokhare A, Metkewar PS. Typical model–amalgamation of multilevel graph partitioning. Advances in Theoretical and Applied Mathematics. 2016; 11(2):115–29.
13. Morozoff EP. Using a line of code metric to understand software rework. IEEE Software. 2010; 27(1):72–7.
14. Bhatia S, Malhotra J. A survey on impact of lines of code on software complexity. International Conference on Advances in Engineering and Technology Research (ICAETR), India; 2014. p. 1–4.
15. Yao L, Hong-Wen Z. An approach for quantifying Enterprise Value of Information (EVI). International Conference on Management Science and Engineering, ICMSE, China; 2009. p. 594–600.
16. Farquar KJ, Mosleh A. An approach to quantifying reliability-growth effectiveness. Reliability and Maintainability Symposium, Annual, USA; 1995. p. 166–73.
17. Geetha NK, Sekar P. Graph theory matrix approach – a review. Indian Journal of Science and Technology. 2016 Apr; 9(16):1–4.
18. Jeelani MI, Mir SA, Nazir N, Jeelani F. Modified ratio estimators using linear combination of co-efficient of skewness and median of auxiliary variable under rank set sampling and simple random sampling. Indian Journal of Science and Technology. 2014 Jan; 7(5):1–6.