Fuzzy-Based Algorithm for Resource Allocation

Gurpreet Singh Saini, Sanjay Kumar Dubey and Sunil Kumar Bharti

Abstract The algorithm presented in this paper deals with use of soft computing technique of Fuzzy logic applied with dynamic graph theory to create graphs which can be efficient in resource allocation process in varied environments, i.e., software project management, operating systems, construction models, etc. The algorithm implies one unique factor of dynamicity which makes graph of resource allocation evolving even after primary design due to chaotic nature of the afore mentioned nature of environments. The use of Fuzzy imparts a logical inference mechanism which rules out non-monotonous reasoning perspective of this dynamicity. The algorithm is robust and adaptive to varied environments. The proposed algorithm will be beneficial for more accurate Engineering in terms of reducing the failures and being more specific in answering the allocation of the resources and how the work has to be undertaken using those resources. It will also emphasize on devising a model which can be adhered to with the proper follow ups such that it could be referred to at the time of chaos or failures. "The development of the Algorithm will be much more product centric and will stick to developer's view of development along with customer's view of required functionalities."

Keywords Dynamic graph theory · Resource allocation · Fuzzy logic · Chaos theory · Soft computing

G.S. Saini (✉) · S.K. Dubey
Amity University Uttar Pradesh, Noida, India
e-mail: g.saini4888@live.com

S.K. Dubey
e-mail: skdubey1@amity.edu

S.K. Bharti
Central University of Haryana, Mahendergarh, India
e-mail: sunilbharti@cuh.ac.in

S.C. Satapathy et al. (eds.), *Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications*, Advances in Intelligent Systems and Computing 515, DOI 10.1007/978-981-10-3153-3_7

1 Introduction

Dynamic graph theory [1, 2] is quite new to computing environment and relatively well known in field of Engineering [3]. Through review of publications in renowned journals and few case studies available over the research channels the idea of implementation of dynamic graph theory in the domain of resource allocation struck.

In simple language dynamic graph [4] word is relative term in field of Mathematics and is applied to a situation, where occurs a lot of paths (which could be termed into anything ranging from man power, time, skills, etc.) but, most of them lead to failure. In such situations, a human being driven by his natural conscience works for minimizing the damages based upon his/her evaluation of current situation. But, if a person is unable to evaluate upon his/her current status of work; he/she can never determine or plan the series of events to be undertaken in future to minimize the damages. This very phenomenon gave rise to Chaos theory in Physics [5] which stated "When the present determines the future, but the approximate present does not approximately determine the future." However, with the increase in knowledge the Chaos theory saw evolution to many levels and the latest turning out to be Dynamic Graph Theory "facilitating construction of networks with multifactor-based path discovery from links within the network" [6].

There has been lot of proposals of using dynamic graph in the various fields [7, 8] and few of computer science specifically the field of software engineering is still not tapped to its full potential and still limited to testing domains [3, 6, 9, 10]. Software engineering is already predominated with few successful models of resource allocation Engineering which have been widely accepted in term of Development and rating Quality. But, with new innovations chaos has been successfully implied in development of dynamic graph theory [11]. Practically all the models are totally developer oriented in terms of their approaches and most of the times it leads to, non-satisfaction of the end user in terms of quality they expect out of the developed system. The Software quality is generally defined as "quality is a perceptual, conditional and somewhat subjective attribute and may be understood differently by different people" and is one such factor which determines the successful nature of the system developed through extreme hardwork of developer; but is totally a factor reliant over the end user or the vendor for which it is developed. The idea of dynamic graph-oriented resource allocation is quite simple in terms of understanding.

As per the reports by Richard Schmidt, Sirrush Corporation in October 2012 [12], around Two-third of the software projects developed in 2009 either "failed" or were "critically challenged" leading to their extensions in timeline by many years and hence incurring huge losses. To this very point, we present an efficient algorithm for resource allocations to be conducted.

The idea is simple and it states "Simply take the requirements providing very critical and uttermost required functionalities based on decisions taken through Dynamics network generated through the Stable marriage resource allocation based

on stated requirements during earliest phase of software cycle, and build a system around it. The leftover requirements should be imparted to the system as an update."

1.1 Literature Background

As mentioned earlier dynamic graph theory [11] was developed out of Chaos Strategy in view to answer the context relative to future events which are going to occur, based upon the current future. The official statement of Chaos theory states "When the present determines the future, but the approximate present does not approximately determine the future." This theory was highly used in answering the future states in operations of nonlinear, complex systems wherein uncertainty and predictability play a vital role.

Similarly, the field of engineering relies totally on uncertainty and prediction to drive upon and reach the goals stated during the initial phase of project development life cycle. Initial literature review lead to discovery of simple strategy of dividing the problem state into substantial sized individual and nonconflicting smaller problems each having its own state definition. The division of the problem is as follows [13–15]:

(a) A complex Module is an incomplete task with lot of dependencies.
(b) The most important fact is to divide modules on the basis of large, timely, and dependent issues.

    1. Large issues provide a functionality to a user and are basis of this development over which they will be imparted to the project.
    2. Timely issues are the ones which require immediate assistance or else they will delay the other work.
    3. Dependent modules are already tested components which are being reused as per the requisite of the project.

(c) The output is presented when the problem has been addressed completely and a state of stability is achieved.

However, so close the "Chaos Model and Graph model" came in answering many prominent questions related to "what is to be done?," it failed in answering most of the questions relative to "How it is to be done?." To this very point fuzzy algorithm solution using stable marriage resource allocation will be applicable. The Models had major backdrops in form of answers to the following:

(a) What requirement or functionality has to be designed at a given time and what resources, time, testing strategy (Graph Model had this feature) has to be employed over it?

(b) In defining, how one could break down the system and produce an early working model?
(c) Finally, to what extent we can minimize the risk of "failures"?

2 Algorithm

The work aims at implementation of soft computing techniques using dynamic graph for resource allocation strategy in prioritizing the requirements using specific weights allocated to them based upon their nature. The weights will be based upon probabilistic data which will be refined during the course of complete fuzzy process. Then, finally after finishing the of fuzzy process, the priorities will be given to requirements keeping into mind applied algorithm for predicting future steps based upon the current status. The complete process will take into account eradicating the drawbacks faced in chaos model of engineering such that the team gets an idea of putting appropriate resources on the given module/requirement(s) such that it does not leads to a failure. After each module completion an iteration will be made to early working model created at the beginning step. Hence, also giving this approach an evolutionary feature. The key steps involved in this approach are:

(a) The major problems should be broken down into smaller ones and a search must be conducted to convert each sub-problem into a question.
(b) Conduct an exhaustive search through various modus operandi into relevant text for answering the questions generated.
(c) Conduct tests of validity over all the answerable questions for finding impact of each question over the project and it is domain of applicability.
(d) Convert the findings into a practical evidence to support decisions and revise them as per the customer's values.
(e) Conduct performance evaluations over the set parameters and do evolutionary improvements as per the integrated modules into the project.

2.1 Advantages of Fuzzy Algorithm

There have been numerous constraints that evolved in past as well as while designing this algorithm. There were quite a few models designed and they did answer few of the backdrops but in the process, they moved away from the basic fact of dynamicity in development and hence, the requisite of efficient resource allocation came into existence which was totally based upon predictive nature of

future given the current statuses of the system. Each one of the models were clear in their approach but lacked in answering following few questions:

(a) Defining "what requirement or functionality has to be designed at a given time and what resources, time, testing strategy has to be employed over it."
(b) Defining "how one could break down the system and produce an early working model"; hence providing more productivity to the end-user at a very early phase and cutting down losses at both developer and end user site.
(c) Finally, "minimizing the risk of "failures" by minimizing the chances of facing "critical situations" leading to permanent shutdowns."

In general situations of Dynamicity, the software development team implies damage recovery procedures based upon any of the above available methods or some classical software development approaches.

As the problem is answering all nonpredictive models of development, the dynamic strategy works in chaotic manner while meeting deadlines. The project managers aim at finishing the tasks given the list of problems. This is usually done through a process of prioritizing them through an experienced professional of the same domain and accepting it as the only valid way of fixing all the issues.

This approach is implied at every Software Development Cycle and to overcome these fuzzy algorithm have been designed to answer all the above question which adds an advantage to the project planning team [16–18]. The algorithm has been successfully implied in answering the major other backdrops which are mostly human perspective based. The simplest possible perspective-based reason which could be quoted out based upon literature and observation could be stated as "How one can say that "someone" who prioritizes the tasks has done that efficiently as the software quality states "*quality is a perceptual, conditional and somewhat subjective.*""

This definition lays down the approach for Improvements. The last part of sentence marks that each individual has its own expectations. At the stage of "critically challenged," an end user cannot expect a standard working system, however, he does expects a system fulfilling the basic needs of its business. And that could only be provided in stipulated time if the prioritization of work has been done in efficient manner and is subjective to the functionalities expected out of the system.

2.2 Design

The idea is simple and it states *"Simply take the requirements providing very critical and uttermost required functionalities, based on decisions taken through Dynamic network. One must take into account that those functionalities are generated through the applied algorithm on stated requirements during earliest phase of Engineering cycle. Once identified start building a system around it. The leftover requirements should be imparted to the system as an update".*

The approach is as simple as its statement. It can be simply understood as following steps.

(a) Retrieve all the requirements. Divide them on the basis of priorities not set by an individual but through the collective weighing by end users and an experienced Team.
(b) Prepare a heuristic to determine weights or priorities of modules.
(c) Apply stable marriage approach for resource allocation at each of the requirements and start predicting the beginning to end prioritization of the work. The initial module or the iterations of module to be processed could be prioritized using fuzzy logic techniques [17–19].
(d) Start working on the modules and prepare working models implemented in live environment, and add more and more functionalities with new Iterations or updates to the older module.

Identify the Membership Function over which requirements could be fuzzified
Creation of weights for each Membership Function and preparing rules to prioritize the requirements based upon them
Refining the weights while Fuzzy Inference Engine is ON
Repeating the process for each individual which is part of prioritization
Accommodating each requirement dynamically by creating a union between dynamic graph strategy and fuzzy output
Theoretical verification through Fractal Identification for future projects
Repeating the complete process at each new iteration to be introduced into the first evolution

This strategy can eradicate following Problems:

(a) Only an individual taking up the task of prioritization.
(b) Transparency in Development cycle for the end user.
(c) Early working system.
(d) Expanded timeline for complete Development of system.
(e) Independence to explore new methods for solution.
(f) Independent modules, resulting into reusability in terms of software or civil constructions.

2.3 Backdrops

The process of algorithm design, however reflected few important points which are hard to neglect and may not be reformed to some extent:

(a) *History is not always the best support to build future upon*: Historical data can teach us how to proceed but can never inform of efficient use of resources. This is the dynamic nature of projects. This dynamic aspect voids all the constants and results into new complex equations.

(b) *The issue of nonpredictable values*: When one discusses and plans one complex system, one cannot ensure all the constraints have been defined and all the variable shave been assigned some values. There will always be few unpredicted values which may pop up due to dynamic nature of the problem.
(c) *Self-defeat of algorithm*: After a practice has been generalized into mode of operations and people have understood its protocol or algorithm; they may use it as per their personal modifications and this beats its essence.

2.4 Algorithmic Outcomes

Along with above-mentioned important outcome of literature survey, research will also take into consideration many factors:

(a) Time period of every iteration
(b) The man power required
(c) Testing required
(d) Future functionality considerations
(e) Coupling of new updates into older versions
(f) Training of end users

The strategy can be implemented at the beginning itself or could be used whenever during any phase of project cycle the development team faces a Challenged situation.

The algorithm has answered almost all the backdrops with efficient deployment of Fuzzy for computing the relevant no. of resources and the allocation process is efficiently done using the stable marriage algorithm. The fusion will result in efficient planning of projects and the deployments may be done even outside the IT projects with few customizations relevant to the domain of implementation.

3 Conclusion

The proposed research work will be beneficial for more accurate Engineering in terms of reducing the failures and being more specific in answering the allocation of the resources and how the work has to be undertaken using those resources. The results of the algorithm will also emphasize on devising a model which can be adhered to with the proper follow ups such that it could be referred to at the time of chaos or failures. "*The development of the model will be much more product centric and will stick to developer's view of development along with customer's view of required functionalities*."

4 Future Scope

The design of algorithm started with developing a generalized framework which could be employed in varied environments and hence a few backdrops did evolve as stated in sections above. The future actions may comprise undertaking this generalized framework and turning it to environment specific framework which could turn out to be a real-time project management solution to the specific nature of the problem.

References

1. Konig, M.D., Battiston, S., Napoletano, M. And Schweitzer, F., "On Algebraic Graph Theory and the Dynamics of Innovation Networks", Networks and Heterogeneous Media, Volume 3, Number 2, June (2008)
2. Shai, O., Preiss, K., "Graph theory representations of engineering systems and their embedded knowledge", Artificial Intelligence in Engineering, Elsevier, Vol 13(1999)
3. Toffetti, G., Pezze, M., "Graph transformations and software engineering: Success stories and lost chances", Journal of Visual Languages and Computing, Elsevier, Vol 24(2013)
4. Konig, M.D., "Dynamic R&D Networks - The Efficiency and Evolution of Interfirm Collaboration Networks", Dissertation – 18182, ETH ZURICH (2010)
5. Boccaletti, S., Grebogi, C., Lai, Y.C., Mancini, H., Maza, D., "The Control of Chaos: Theory and Applications", S. Boccaletti et al. / Physics Reports 329, Elsevier (2000)
6. Chong, C.Y., Lee, S.P., "Analyzing maintainability and reliability of object oriented software using weighted complex network", The Journal of Systems and Software, Elsevier, Vol 110 (2015)
7. Attri, R., Grover, S., Dev, N., "A graph theoretic approach to evaluate the intensity of barriers in the implementation of total productive maintenance (TPM)", International Journal of Production Research, Taylor and Francis (2013)
8. Schweitzer, F., Fagiolo, G., Sornette, D., Redondo, F.V., White, D.R., "Economic Networks: What do we know and what do we need to know?", ACS - Advances in Complex Systems, Vol 12, Number 4(2009)
9. Robinson, H., "Graph Theory Techniques in Model-Based Testing", International Conference on Testing Computer Software (1999)
10. Lane, P.C.R., Gobet, F., "A theory-driven testing methodology for developing scientific software", Journal of Experimental & Theoretical Artificial Intelligence, Taylor & Francis (2012)
11. Saini, D. K., Ahmad, M., "Software Failures and Chaos Theory", WCE -London, Vol II (2012)
12. Schmidt, R., "Software engineering: Architecture-driven Development", NDIA 15th Annual Systems Engineering Conference, October (2012)
13. Jifeng, H., Li, X., Liu, Z., "Component-Based Software Engineering-The Need to Link Methods and their Theories", 973 project 2002CB312001 of the Ministry of Science and Technology of China
14. Boehm, B., "Value-Based Software Engineering: Overview and Agenda", USC-CSE-2005-504, February (2005)
15. Dybå, T., Kitchenham, B.A., Jørgensen, M., "Evidence-Based Software Engineering for Practitioners", IEEE Software, Published by the IEEE Computer Society, January-February (2005)

16. Kelly, D., “Scientific software development viewed as knowledge acquisition: Towards understanding the development of risk-averse scientific software” The Journal of Systems and Software, Elsevier, Vol 109(2015)
17. Kumar, G., Bhatia, P.K., “Neuro-Fuzzy Model to Estimate & Optimize Quality and Performance of Component Based Software Engineering”, ACM SIGSOFT Software Engineering Notes, Vol 40(2015)
18. Mishra, S., Sharma, A., “Maintainability Prediction of Object Oriented Software by using Adaptive Network based Fuzzy System Technique”, International Journal of Computer Applications, Vol 119(2015)
19. Tyagi, K., Sharma, A., “A rule-based approach for estimating the reliability of component-based systems”, Advances in Engineering Software, Elsevier, Vol 54 (2012)