TECHNIUM
Technium Vol. 12, pp.58-64 (2023)
ISSN: 2668-778X
www.techniumscience.com

A Solution to the Next Release Problem by Swarm Intelligence

Deema Yahya Qassem deema.20csp33@student.uomosul.edu.iq
Dr.Najla Akram Al_saati dr.najla_alsaati@uomosul.edu.iq

University of Mosul, College of Computer Science and Mathematics
Software Department

Abstract: First of all, in this research, we solve the problem of the next release ((NRP) (Next Release Problem)), which is classified as a multi-objective difficult problem (NP_ hard problem) using swarm intelligence, since the programs are spread in all areas of our life and process The development on it is constantly ongoing and the selection of the optimal requirements to satisfy customers for the following versions is a very important process, as the requirements that have been dealt with are complicated due to interdependence and other limitations. Therefore, we will highlight it in our research to solve it, as the problem of the next release (NRP) is defined as a multi-objective improvement problem with two conflicting goals, which are customer satisfaction and development cost, and since it is a multi-objective problem, we chose swarm intelligence to solve it, where we solved This problem using the Multi_objective Mayfly Algorithm is derived from the behavior of the swarms of the Mayfly in nature.
Keywords: Artificial e Intelligence, Next Release Problem, Multi-objective, Swarm intelligence

1. Introduction

Many software companies these days focus on creating, maintaining, and improving large, sophisticated systems. Customers are typically interested in purchasing and using these systems, and each has specific requirements or preferences. Deciding what to include in the next release of their products in order to satisfy the largest number of consumers while incurring the lowest cost to the company is therefore one of the first considerations that software companies must make, and this issue is referred to as the Next Release Problem (NRP). In engineering, a requirement is a specific and documented need for what a particular good or service should be or be able to perform.It is a declaration that defines the basic feature of the system in order for it to be useful and of value to users, so (NRP) entails choosing a subset of the requirements of the software package by assigning Doing so with the least amount of money or resources while ensuring the highest level of customer satisfaction [4].

2. Literature Review

In 2019, Carlos and colleagues proposed to solve the next release problem by applying the Fuzzy Bi-Objective Particle Swarm optimization algorithm. A smart algorithm was needed to explore a larger extension of the Pareto ideal front to provide a larger set of possible solutions. The results show that this algorithm is suitable for finding Pareto fronts. [6].

In 2020, Humayun and his colleagues proposed to solve the next release problem by means of a system of fuzzy reasoning. An algorithm based on the system of fuzzy reasoning was presented to determine the suitability of each requirement for development in the next release problem. It is the responsibility of the proposed algorithm, rather than the developer team, to determine the optimal subset of requirements for the development of the next release problem.

Then the experimental results of the proposed algorithm are balanced by the results of the genetic algorithm. The subset selected by the proposed algorithm provides much greater satiation than the genetic algorithm[7].

In 2021, Alaa Omar suggested studying the issue of the next release problem about the control parameters and contrast factors, as she balanced several algorithms in terms of the two quality indicators (Hypervolume HV and S SPRED). and classical, and then study the effect of adjusting the parameters on the most prominent algorithm among the five algorithms. This study shows that controlling the crossover and mutation probability of MOcell values while the algorithm is running to increase exploitation and decrease exploration over time does not lead to an increase in MOCell performance counter balanced by tuning the parameter to the algorithm itself. [1].

In 2022, researcher Fang Chen and colleagues benchmarked the performance of nineteen modern, multi-objective evolutionary algorithms that could address the issue of the next release problem. The matter is designed to maximize customer satisfaction and minimize the total cost required. Three indicators, namely (Hyper volume) (HV), (Spread) (S), and running time, were examined to balance the algorithms. Two types of data sets, i.e. classical and real data, were also examined from small to large scale to check the applicability of the results. [5].

3. Next Release Problem

Next Release Problem is process of defining specifications and features (requirements) must be added to the software for each release. However, determining what requirements will be in the next release is a difficult task, given the many constraints involved, such as budget, interdependence, and technical issues in this context [2].

As software company seeks to create a set of requirements while being affected by a number of factors including:

1- Customers are evaluated at different levels of importance to the developer company.

2- The huge volume of customer requirements that must be complied with.

3- Requirements are varied and different that require a great deal of time and effort.

4- Reliability between requirements, as some requirements may be linked to other requirements.

Solving next release problem is critical to the success of any company:.

1- Assist stakeholders in choosing the basic requirements of the system by guiding them in their decisions.

2- Support and assist in planning, organizing and defining the optimal set of requirements for the software project, which will be included in subsequent iterations.

3- Achieving balance by taking advantage of the requirements and their cost.

4- Strive to balance the effects of requirements on cost reduction and customer satisfaction in the software development architecture.

5- Dealing with interactions or constraints between requirements, which helps resolve disagreement between stakeholders and customers on the best way require for selecting requirements.

6- The importance of each requirement must be weighed in order to provide the highest level of satisfaction at the lowest reasonable prices. [10].

4. Requirement Interaction

A number of interactions or constraints between two or more potential software product development requirements may arise during the software development process. These interactions or limitations can also be described as a set of guidelines that must be adhered to and taken into consideration when selecting requirements. As failure to do so will lead to conflict when selecting new requirements for the next versions of the program.

As it comes the functional interactions:

1- Precedence or Implication ($r_i \Rightarrow r_j$) That is, the requirement (r_j) must be selected and executed first, so that the requirement (r_i) can be selected and executed.

2- Combination or Coupling ($r_i \otimes r_j$) That is, each of the requirements (r_i) and (r_j) cannot be selected separately and must be selected together.

3- Exclusion ($r_i \oplus r_j$) That is, it is not possible to choose both requirements together, because if we choose the requirement (r_i), we cannot choose the requirement (r_j), and vice versa. [3].

- The precedence relationship between requirements is represented by edge ($r_i \rightarrow r_j$) in the graphic.
- The coupling relationship between the requirements is represented by the edge ($r_i \leftrightarrow r_j$) in the figure
- The exception relationship between requirements is represented by the edge ($r_i \not\leftrightarrow r_j$) in the drawing.

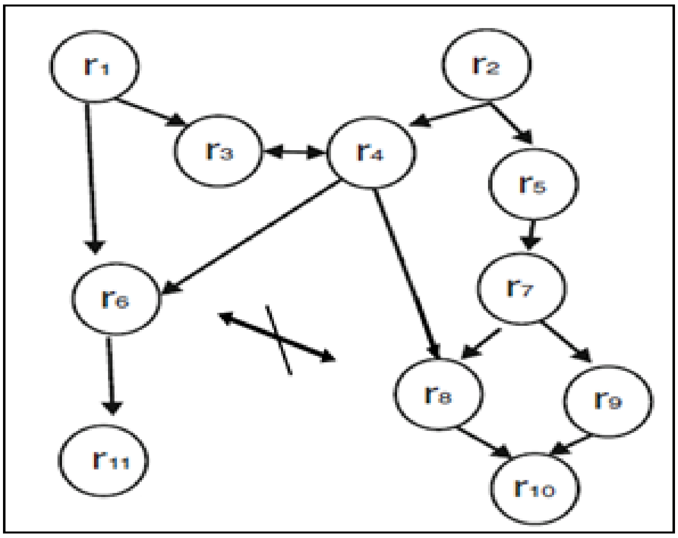

Figure I: Functional interactions between requirements

5. Applying algorithm interactions and constraints (precedence and coupling) between requirements:

The algorithm for applying interactions and constraints includes several steps as follows:

First, reading data comes from the set of requirements and the constraints between requirements

Second: Determine the length of the requirements row in the counter (len) and set the main counter to start with length (0).

Third: Bring the requirements row

Fourth: Determine whether the value of the main counter matches the length of the value of the requirements row

a_ Repeat steps 5 and 6 if the value is less than or equal to the length of the requirements row

b_ If not, go to step 7.

Fifth: Check the requirements, is it a built-in requirement, as follows:

a_ A value (1) is set for the relevant built-in requirements if there is precedence between the requirements.

b_ If not, set a value (0) for the relevant non-inline requirements

Sixth: Examine the requirements constraints to see if there is an associated constraint:

a_ A value of (1) is placed if there is an association, for the requirements that are associated with each other.

b- Otherwise, the main counter must be increased by a value of (1) and move to the fourth step.

Seventh: Store new requirements resulting from the process of precedence and pairing

Eighth: End of the process

6.Multi-objective Mayfly Algorithm (MMA)

The Mayfly algorithm (MA) is based on the social behavior of mayflies, and it is one of the swarm intelligence algorithms, taking into account that the fly is already an adult and that the fittest flies live in the algorithm, regardless of how long they live. The location of each fly in the search space represents a possible solution to the problem. In the beginning, two groups of mayflies are randomly generated for the male and female population, i.e. each fly is randomly placed in the problem space as a candidate solution. The area of the solution is represented by (d, dimensional), and the location of the mayfly by vector (x=$x1$, ,…,xd) and its performance is evaluated. Velocity for a mayfly (v=$v1$,…,vd,) is defined as a change in its position, and the direction of flight for each mayfly is a dynamic interaction of both individual and social flight experiences [8].

Each fly specifically adjusts its trajectory towards its personal best position so far (pbest), as well as the best position reached by any fly in the flock so far (gbest), which represents the best position among the best fly locations achieved so far (Global Best) [8].

7. Optimization Schema

The next release problem was initially approached as a single-purpose issue and based on satisfaction value, as the complexity of the issue was greatly reduced. However, the issue was later treated as multi-objective because it contained two main and conflicting objectives at the same time, which made it difficult to choose the best requirements for the next stage[12].

Maximize f(x)

Subjected to $gj(x) \leq B$, j=1,2,….., J

x□X

As:

- f(x) represents the objective function.
- gj(x) is the vector of constraints of inequality.
- j: the number of entries for the inequality.
- B: Budget limits.Single Objective Optimization

A. Single Objective Optimization

Single-objective optimization, when balanced against multi-objective optimization, has the advantage of dealing with a single objective and arriving at a solution that is at least the best solution found. In this case, the NRP issue is seen as a single-purpose issue that provides for meeting most requirements while staying within budget. The following equations provide a mathematical formula for the single-objective optimization problem. [8].

B. Multi-Objective Optimization

Many optimization issues in many different settings require solutions that take into account a number of objectives that are equally important to each other. Similar to the single-objective optimization problem, the multi-objective optimization problem entails a number of functions that must be made smaller or larger. Any possible solution (including all optimal solutions) to a multi-objective problem may have a number of constraints that it must satisfy. The following equation can be used mathematically and in general to formulate the multi-objective optimization problem. . [11].

Minimize, Maximize $f_m(x)$, m =1,2,….., M

Subjected to $g_j(x) \geq 0$, j =1,2,….., J

$h_k(x) = 0$, k =1,2,….., K

$xi^{(L)} \leq xi \leq xi^{(U)}$, i =1,2,… …, n

As:

$f_m(x)$: represents the multiple objectives function.

m the number of targets of the issue.

$g_j(x)$ vector constraint inequality.

j number of entries for the inequality.

$h_k(x)$ vector equality constraints.

K the number of equality entries.

x decision variables.

n the number of decision variables.
the same. In this proposed system, logical errors will be achieved, and a slight change will be made to the algorithm of the application that will be under test

4. Design decision-making for Pareto goals

The set of solutions (Pareto X) is a matrix containing a number of (n) rows and (m) columns, and the absolute value of each solution in (X) can be counted as one of the decision-making indexes, or as the product of the unit solution to the (Pareto) terms, as shown below . [9].

$$\boldsymbol{X'(i,j) = |X(i,j)|, i = 1, 2, \ldots, n, j = 1, 2, \ldots., N}$$

5. Used data
In this work, the data set for solving the multi-objective next release problem issue was studied, as mentioned in the previous paragraph, which included a set consisting of (100) requirements with (5) customers) as well as an effort to develop each requirement from (1-100) and the level of importance For every requirement for every customer. The database also contained the importance of each customer to the company requesting the software product. The data set also contained a number of interactions or restrictions that must be followed during the implementation of the requirements, which included (42) restrictions represented by (38) precedences and (4) coupling.

6.Conclusion

The results showed a gradual increase in the percentage of customer satisfaction during the continuous increase in iterations, as if we look at the iterations, for example, from 10_20, we will see that they contain a value or more higher than the first ten iterations of the percentage of customer satisfaction, until the highest percentage of satisfaction is reached while maintaining the required budget limit, The research continues to ensure that the percentage of satisfaction obtained is the highest, up to the last iteration.

The The multi-objective mayfly optimization algorithm was used in this thesis in order to adapt the algorithm strategy to the nature of the problem, and by applying this algorithm and observing the outputs of the quality and performance measures on a set of data related to the next release problem .

This work indicates that the multi-objective Mayfly Algorithm is able to solve the next release problem, which is of the (NP-hard) type.

7. Acknowledgements

To my father, who helped me through hardships throughout my life.My dear mother, to your kind heart and gentle chest.To the best sisters and brother I have whom were my best supporters I dedicate my scientific research to all of my friends and to everyone who helped the thesis be completed successfully. I also thank my supervisor, Professor Dr.Najla Akram Al_saati , who was The best guide and gave me directions in the course of my research.

8. References

[1] Bandello Alaa Omar,” A study on variation operators' parameters Tuning: Next-Release Problem as a case study”, Birzeit University (pp. 1-3).

[2] Bagnall, A.J, Rayward-Smith, V.J, Whittley, I, (2001),"The next release problem". Inform. Software. Technol. 43 (14) 883–890.

[3] Carlshamre, P., Sandahl, K., Lindvall, M., Regnell, B., Och Dag, JN., (2001), "An industrial survey of requirements interdependencies in software product release planning". In: Proceedings of 5th IEEE International Symposium on Requirements Engineering (RE 2001), Toronto, Canada, IEEE Computer society

[4] Durillo, J.J., Zhang, Y., Alba, E., Nebro, A.J., (2009), "A study of the multi-objective next release problem". In: SSBSE ’09: proceedings of the 2009 1st international symposium on search based software engineering. IEEE Computer Society, Washington, pp 49–58.

[5] Fang Chen, Iman Rahimi, Amir H Gandomi, Mohammad Reza Nikoo,” A Comparative Study on Evolutionary Multi-objectiveAlgorithms for Next Release Problem”,2022, University of Technology Sydney.

[6] Gorski, J.,(2018),"Multiple objective optimization and implications for single objective optimization" (Doctoral dissertation, University of Wuppertal, Faculty of Mathematics and Natural Sciences .Mathematics and Computer Science dissertations.

[7] Homayun Motameni & Hamidreza Alrezaamiri , Ali Ebrahimnejad (2020) “Solving the next release problem by means of the fuzzy logic inference system with respect to the competitive market” ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/teta20, Journal of Experimental & Theoretical Artificial Intelligence.

[8] Konstantinos Zervoudakis, Stelios Tsafarakis ,2020" ,A mayfly optimization algorithm ," Technical University of Crete, School of Production Engineering and Management.

[9] S. Liu and L. N. Vicente, ‘Convergence rates of the stochastic alternating algorithm for bi-objective optimization’, Lehigh University. ISE Technical Report 22T-004

[10] Wang, S., Lu, Z., Wei, L., Ji, G., & Yang, J., (2016), "Fitness-scaling adaptive genetic algorithm with local search for solving the Multiple Depot Vehicle Routing Problem". Simulation, 92(7), pp.601-616.

[11] Yang, X. S., (2014), "Nature-inspired optimization algorithms", Elsevier.pp.1-263.

[12] Zhang, Y., Harman, M., Ochoa, G., Ruhe, G., and Brinkkemper, S., (2018), "An Empirical Study of Meta- and Hyper-Heuristic Search for Multi-Objective Release Planning". ACM Transactions on Software Engineering and Methodology (TOSEM).