

Machine Learning Based Software Projects Allocation using Flower Pollination Optimization Algorithm

¹R. Mamatha, ^{2*}Dr. P. Lalitha Surya Kumari, ³Dr.A. Sharada

¹Research Scholar, Department of CSE, Koneru Lakshmaiah Education Foundation, Hyderabad-500075, Telangana, India. mamatha.racharla@gmail.com

²Professor, Department of CSE, Koneru Lakshmaiah Education Foundation, Hyderabad-500075, Telangana, India. vlalithanagesh@gmail.com

³Professor, Department of CSE, G. Narayanamma Institute of Technology & Science, Hyderabad, Telangana, India. sharada@gnits.ac.in

Abstract

Resources are often scarce, and projects will constantly need them. The act of allocating and scheduling the available resources in the most effective and efficient way is known as resource allocation[24]. Therefore, a project manager's responsibility is to choose the best time for those who must adhere to the project's timetable. Allocating resources to a project in accordance with the business plan and the resources at hand, management is a continual activity that must be performed effectively. This study employs the Flower Pollination Algorithm to try to solve the resource allocation problem (FPA). First, the historical project data from the organization is taken into account to extract the necessary features using FPA and input them into the machine learning algorithm for decision-making. This approach benefits the project leads and management. This method helps the project managers/leads to save their time in allocating the task to the right person and it does very efficiently without any bias in allocation.

Keywords: Scheduling, Flower Pollination, Resource allocation, Machine learning.

1. Introduction

Because human resources must be distributed manually, poor management performance and inefficient resource allocation result [23]. In addition, as software resources predominate over expensive equipment and labor in the production of software Funding for projects is often more openly allocated than to individuals engaged in industrial or construction initiatives. The majority of the resources needed for software development are humans because it is a labor-intensive activity [24]. Different software project jobs necessitate different skill sets, and employee skill competency has a significant impact on project execution performance. As a result, selecting the best employees Because it can be difficult to match tasks to project managers, allocating human resources has become an essential part of software project preparation. Models for resource allocation and scheduling, such as the RCPSP, are ignored by techniques such as PERT and CPM and do not account for the assignment of workers with varying skill sets.

The knowledge obtained from historical project data sets can be used to develop predictive models using either a mathematical approach, such as linear regression and the study of associations, or machine learning (ML) techniques, such as Artificial Neural Networks (ANN) and Support Vector Machines (SVM). Predictive approaches provide a method that is focused on both historical and current project information in order to forecast the project's future. Because there are so many ML algorithms, one should investigate the best algorithm for the project data before making a decision.

In terms of seeking improvements in model performance, the most important step in the creation of models for tabular datasets is the pre-processing and manipulation of the input features. This includes combining, modifying, cleaning, and filtering data in a feature engineering procedure in order to build new features based on existing features. According to the theory, such manufactured traits enable a model to discover interactions between characteristics, allowing for more precise prediction on a specific body of knowledge. Feature engineering is a time-consuming process with no set recipe to follow, making it difficult to implement successfully without the assistance of a specialist.

2. Preliminary study

A. Human resource allocation: The Project Management Body of Knowledge (PMBOK) lists ten project management areas, with this paper focusing on the human resource area. To effectively manage resources and avoid under or overutilization of the workforce, resource allocation enables the selection of the best assets for a variety of projects. Unfortunately, not all project managers utilize it.

A.ML (Machine Learning): Machine learning (ML) employs artificial intelligence to enable computers to learn from experience and advance without explicit programming. In other words, the main goal of machine learning is to allow computers to learn autonomously, without human intervention, and then modify their behaviour accordingly. Furthermore, ML allows for the processing of massive amounts of data.

B.ML for Software Project Management: Project managers can use ML to train the model with available historical data in order to solve resource allocation problems that will efficiently allocate the right person to the right task without bias.

3. Background Study

Resource-constrained scheduling issues have been successfully resolved by search-based methods like Evolutionary Algorithms (EA) [17]. However, these resource-constrained scheduling issues [18] lack some characteristics in software projects, including the fact that employees can divide their focus across many tasks concurrently.

The resource allocation problem was defined by Di Penta et al. [19] as the issue of dividing workers into teams and selecting the sequence in which work packages should be completed system of queues. Then, assignments are made using the queue system bundles of work to teams. Various search-based methods investigated were single- and multi-objective Ea. The goals are to reduce completion time and/or rather than completion time and expense, or fragmentation. The total number of unproductive person months is known as fragmentation due to the restrictions on precedence imposed by the to-be-processed work items.

Chang et al. [20] created a timeline to divide a task into smaller, time-sliced actions. Once a task has begun, it can be interrupted in this manner, and employees are not required to work on a specific task from start to finish. Other features included the distinction between an employee and a contractor, varying payment amounts based on whether there is excessive effort, varying levels of ability, the possibility of training throughout the project, and so on. Combining all of these results in the formulation of a problem, as well as the introduction of a large number of algorithmic input values that must be subjective.

[21] describes collaborative methods for allocating and specializing in Software Engineering (SE) work. They conclude that behavioral studies, such as those conducted by HCI (Human Computer Interactions), are required to improve collaborative SE processes. Altmann et al. describe a process model and a product model for the creation of collaborative software [22], demonstrating that both productivity and technical quality should be improved. The authors provide a brief overview of group-supportive work, and Jastroch et al. discuss the importance of the environment for CSD of the core as well as inter-organizational collaboration activities in software production.

Although the works above address the problem of systematically assessing the domain of software development tasks so that one can think about them in terms of abilities, they do not consider the project's internal possibility for allocation. In this paper, we solve this by considering employees' bug-resolving capability to identify their expertise and then mapping with the skill to decide on task allocation.

4. Methodology

We used a dataset from the Git-Hub repository to solve the resource allocation problem, which has over 10,000 records and more than 18 attributes. We are now attempting to process the Summary field, which consists of a text-based description of a bug related to the product. The objective of this paper is to attempt to assign the resource that can best solve the bugs raised by the customer; however, in order to do so, the corpus's text must first be transformed into a format that machine-learning algorithms can understand. This resource allocation problem is solved in the literature using a Genetic algorithm, Integer programming, Clustering analysis, and so on. When gradient information is unavailable, these algorithms were among the first to provide significant advantages for determining global optimality in large, complicated search spaces. The implementation of these algorithms is based on simple ideas, and they have relatively high accuracy and convergence rate. They are also fairly adaptable because the parameters can be easily adjusted for improved performance. There were numerous hybrid algorithms or variations on several evolutionary algorithms. Nonetheless, significant issues such as sluggish convergence or early convergence persist. Furthermore, for extremely complex situations, these techniques may be computationally demanding and require multiple iterations.

To address the aforementioned drawbacks, modern algorithms frequently have heuristic and metaheuristic characteristics. Metaheuristic algorithms are higher-level heuristics that generate new solutions using memory, solution history, and other forms of "learning" strategies rather than the trial-and-error method used by heuristic algorithms. The majority of metaheuristic algorithms are currently inspired by nature, and this type of algorithm is based on natural swarm intelligence. [1][9][13].

4.1.1 Problem definition:

- There are X employees in the organization.
- Each employee X has P projects.
- There are C candidates, and a number of them are selected for projects.
- Each candidate can be assigned to more than one project at a time.
- Different scenarios are defined to assign the task to the employee/team member.

- The amount of time each selected team member in each project should be more than the minimum required time for that project.
- For each project, a competency profile based on bug resolving capacity and skill set is described.
- The manager assesses the team member based on the competency profile of that particular employee.
- Two objective functions are considered in the model
 - Minimizing the HR cost of team members.
 - Maximizing the competency of team members.

4.2.2 Background: Flower Pollination Algorithm

A meta-heuristic optimization algorithm is being used to solve the above problem definition. In response to flowering plant pollination traits, a population-based algorithm known as the Flower Pollination Algorithm (FPA) [16] was developed. The goal of FPA is to mimic some of the most important aspects of both biotic and abiotic pollination, as well as the co-evolutionary flower constancy between specific flower species and pollinators such as insects and mammals. The pollination process in plants is mimicked by FPA.

All algorithms use each agent in a population of many agents, such as particles, ants, bats, cuckoos, fireflies, bees, and so on, each representing a solution vector. When it comes to physical fitness, the ideal response is frequently found among the populace. The various solutions of a population reflect its diversity and fitness. Certain operations (such as mutation and crossover) frequently facilitate population evolution, which is frequently expressed in terms of formulas or equations derived from algorithms. Iterative evolution is typical, resulting in the evolution of solutions with varying properties. When all solutions have sufficiently converged, the system is said to have converged.

An optimization algorithm works by first randomly generating the initial population, then computing fitness values for each solution, combining, moving, or evolving the initial population over a predetermined number of iterations, and repeating this process until the best solution is obtained. Fig 4.1 shows the FPA process.

The following equation represents a mathematical model for global search:

$$x_i^{t+1} = x_i^t + L (g^* - x_i^t)$$

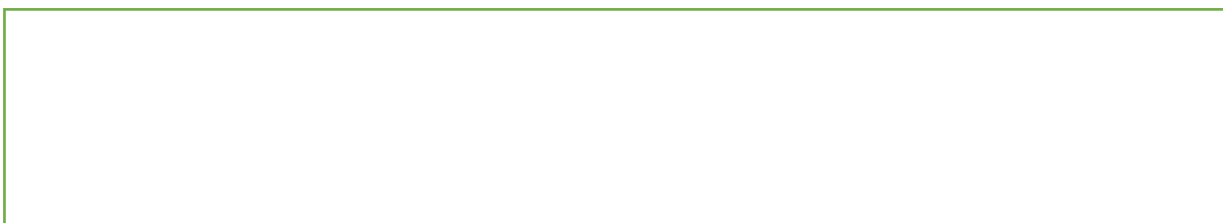
Where x_i^{t+1} represents the new solution

x_i^t is an old solution

L is /levy's distribution

g^* best solution

4.1.3 FPA process:



Step 01: Take N population size

Where $N=1,2,3,\dots$ that represents population size and initial position for N flowers is represented as

$X_i (i=1,2,3,\dots,N)$

Step 02: Using the fitness function evaluate performance of each agent in the initial population which can be given as $f(x_i)$

Step 03: Finding the best agent

Step 04: Defining switch probability $p \in [0,1]$ to decide whether to go with local or global pollination.

Step 05: Initialize current iteration=0

Step 06: check for stopping criteria

while (current iteration \leq max iteration)

Step 07: flower pollination

while (current iteration \leq maxT)

for $i=1$ to N

if rand < p

 draw(d-dimensional) step vector L from Levy distribution

 global pollination

else

 draw from a uniform distribution in [0,1]

 local pollination

end if

evaluate new solution

update position if the new solution is better than the old solution

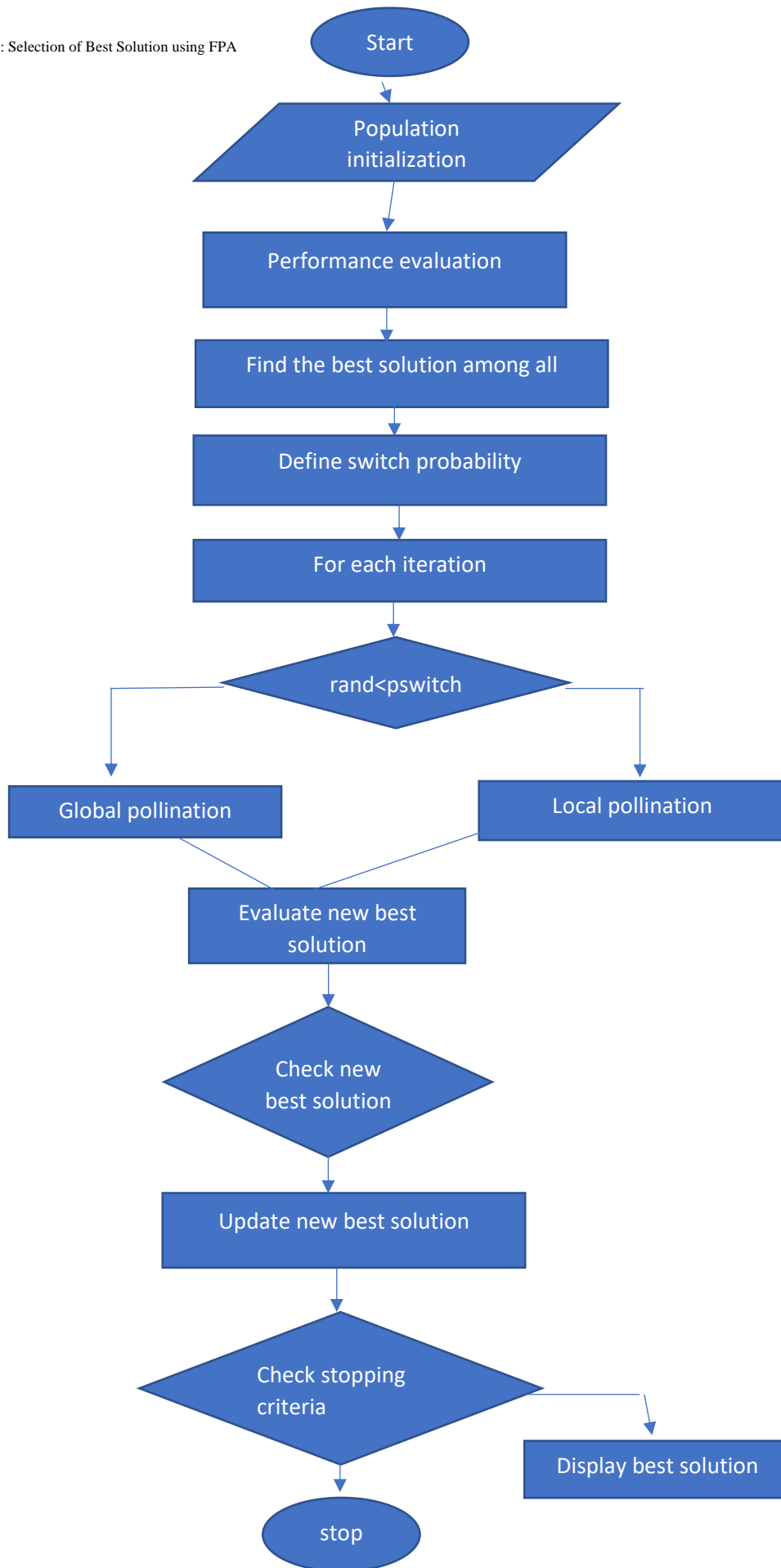
end for

 find the current best solution

end while

display the best solution

Fig:4.1: Selection of Best Solution using FPA



5. Conclusion

When developing the competency model, position assessment, staff expectations, and a human multi-objective criteria mathematical model resource distribution are considered to determine the profit of some employees on specific posts. This technique exemplifies the dual concepts of matching staff skills to bug-solving capacity, as well as the personnel's expectations and the request, which is consistent with the current state. However, a variety of factors influence employee performance. In recent studies, staff competency and expectations are used to gauge employee performance, which will result in some errors, but there are still some issues left. to use FPA to solve a multi-objective problem. Later a machine learning model can be trained to automate the resource allocation task based on historical project data.

References:

- [1] Eswari, R., Nickolas, S.: Modified multi-objective firefly algorithm for task scheduling problem on heterogeneous systems. *Int. J. Bio-Inspired Comput.* 8(6), 379–393 (2016)
- [2] Hou, Y.C., & Chang, Y.H. (2004). A new efficient encoding mode of genetic algorithms for the generalized plant allocation problem. *Journal of Information Science and Engineering*, 20, 1019–1034.
- [3] Luss, H., & Gupta, S.K. (1975). Allocation of effort resource among competing activities. *Operations Research*, 23, 360–366.
- [4] Dai, Yuan-Shun, & Wang, Xiao-Long (2006). Optimal resource allocation on grid systems for maximizing service reliability using a genetic algorithm. *Reliability Engineering and System Safety*, 91, 1071–1082.
- [5] Dai, Y.S., Xie, M., Poh, K.L., Yang, B. (2003). Optimal testing-resource allocation with genetic algorithm for modular software systems. *The Journal of Systems and Software*, 66, 47–55.
- [6] Johannesson, M., & Weinstein, M.C. (1993). On the decision rules of cost effectiveness analysis. *Journal of Health Economics*, 12, 459–467.
- [7] Gil, Alvaro E., & Passino, Kevin M. (2006). Stability analysis of network based cooperative resource allocation strategies. *Automatica*, 42, 245–250.
- [8] Srinivas, N., & Deb, K. (1999). Multi-objective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3), 221–248.
- [9] He, L.F., Huang, S.W.: Modified firefly algorithm based multilevel thresholding for color image segmentation. *Neurocomputing* 240(1), 152–174 (2017)
- [10] Zhang Li-biao, Zhou Chun-guang, Liu Xiao-hua, Xu Xiang-li, Sun Cai-tang “A multiple subswarms evolutionary algorithm for multiobjective optimization problems” ‘Control and Decision’ Vol.22 No.11 Nov, 2007, pp.1312-1320
- [11] Li Ning, Zou Tong, Sun Debao, Qin Yuanqing. “Multi- objective optimization Utilizing Particle Swarm “ ‘Application and Engineering of Computer ’ 2005, Vol.23, pp.43-45
- [12] M. Dorigo and L.M. Gambardella, “Ant Colony System: A Cooperative Learning Approach to TSP,” *IEEE Trans. Evolutionary Computation*, vol. 1, no. 1, pp. 53-66, Apr. 1997.

- [13] Kougianos, E., Mohanty, S.P.: A nature-inspired firefly algorithm-based approach for nanoscale leakage optimal RTL structure. *Integr. VLSI J.* 51, 46–60 (2015)
- [14] D. Merkle, M. Middendorf, and H. Schmeck, “Ant Colony Optimization for Resource-Constrained Project Scheduling,” *IEEE Trans. Evolutionary Computation*, vol. 6, no. 4, pp. 333-346, Aug. 2002.
- [15] W.-N. Chen and J. Zhang, “An Ant Colony Optimization Approach to a Grid Workflow Scheduling Problem with Various QoS Requirements,” *IEEE Trans. System, Man, and Cybernetics Part C*, vol. 39, no. 1, pp. 29-43, Jan. 2009
- [16] Marichelvam, M.K., Geetha, M.: A hybrid discrete firefly algorithm to solve flow shop scheduling problems to minimise total flow time. *Int. J. Bio-Inspired Comput.* 8(5), 318–325 (2016)
- [17] A.E. Eiben and J.E. Smith, *Introduction to Evolutionary Computing*. Springer-Verlag, 2003.
- [18] R. Kolisch and S. Hartmann, “Experimental Investigation of Heuristics for Resource-Constrained Project Scheduling: An Update,” *European J. Operational Research*, vol. 174, no. 1, pp. 23-37, 2006.
- [19] M. Di Penta, M. Harman, and G. Antoniol, “The Use of SearchBased Optimization Techniques to Schedule and Staff Software Projects: An Approach and an Empirical Study,” *Software: Practice and Experience*, vol. 41, no. 5, pp. 495-519, 2011.
- [20] C.K. Chang, H. Jiang, Y. Di, D. Zhu, and Y. Ge, “Time-Line Based Model for Software Project Scheduling with Genetic Algorithms,” *Information and Software Technology*, vol. 50, no. 11, pp. 1142-1154, 2008.
- [21] Hildenbrand T. et al., “Approaches to Collaborative software development,” *Intl. Conf. on Complex, Intelligent and Software Intensive Systems, CISIS- 2008, Mannheim, 2008*, pp 523 – 528.
- [22] Altmann J. and Pomberger G., “Cooperative software development: Concepts, model and tools,” in *Procs. of the TOOLS-30 Conf., (TOOLS-30’99)*. IEEE Computer Society, pp. 1999, pp 194–277.
- [23] C.K. Chang, M.J. Christensen, and T. Zhang, “Genetic Algorithms for Project Management,” *Annals of Software Eng.*, vol. 11, pp. 107- 139, 2001.
- [24] A. Barreto, M. deO. Barros, C.M.L. Werner, “Staffing a Software Project: A Constraint Satisfaction and Optimization-Based approach,” *Computers & Operations Research*, vol. 35, pp. 3073-3089, 2008.