Analysis on Software Project Staffing and Scheduling Using Ant Colony Optimization

Foram Mehta
Computer Engineering
K.J.Somaiya College Of Engineering
Mumbai, India
foram.m@somaiya.edu

Ojus Sangoi
Computer Engineering
K.J.Somaiya College Of Engineering
Mumbai, India
ojus.sangoi@somaiya.edu

Tanvi Chopda
Computer Engineering
K.J.Somaiya College Of Engineering
Mumbai, India
tanvi.chopda@somaiya.edu

Manthan Shukla
Computer Engineering
K.J.Somaiya College Of Engineering
Mumbai, India
manthan.s@somaiya.edu

***Abstract*—** Developing computer compatible and efficient techniques for software project planning are very much required and challenging in various multinational software companies. Most of the software projects are based on mainly human resources and it consists of people intensive activities. Thus proper scheduling of projects and human resource allocation is very much needed for rapid growth and development of software companies in a competitive world. In this paper a prototype based on project management system for project scheduling and human allocation is discussed. The paper discusses about the Ant Colony Optimization Algorithm (ACO) that takes into account both project scheduling and human allocation. For this the algorithm uses a task list and an employee allocation matrix. The paper consists of use of Gantt chart which displays the project schedule thus providing the project manager with the necessary project details (e.g. start date, end date, resource allocated).Thus providing the project managers the ease in functioning. The purpose of developing such a system is that it provides the software companies the flexibility in software planning.

Keywords- *Project management system, project scheduling, human resource allocation, Ant Colony Optimization (ACO)*

*****

I. INTRODUCTION

With rapid growth and development in the software industry it is very challenging for the software companies to succeed in such a competitive world. In order to succeed proper software project planning is needed. The project plans must be planned in such a way that it reduces the overall cost of construction. However in medium to large companies' software planning is very complex. They mainly face two major problems, which are:

1. EMPLOYEE ALLOCATION:

Different projects require employees with different skills and efficiency of a project largely depends on employee skill proficiency. Thus allocating the right employee for the right project becomes an uphill task for the project managers in the software companies.

2. PROJECT SCHEDULING:

This problem occurs due to improper scheduling of projects. The projects here are not scheduled properly in terms of start date and end date .There may be occasions when there is derail in the completion of project.

Thus employee allocation and project scheduling are the important aspect of the project management system and the above two problems must be addressed for efficient functioning of a company.

There are various tools and techniques for addressing the above two problems. Techniques like Evaluation of human Competences and Human resource Management (ECHRM) [1] and Evolutionary Technique based on Run Time Analysis (ERTA) [3] were used. However both were not useful as the former ECHRM lacked consideration for scheduling, it just evaluated the competences of the employee and based on their competency allocated the job to the employee and scheduling of the projects was also not possible using ECHRM [1] technique and the later ERTA technique [3] did not perform efficient allocation and it had certain constraints for e.g. Team of employees assigned to a task must have all the skills. It is not possible that all the employees in a company may have good command on all the skills. Also in large projects the later technique was a challenging problem as it used enormous amount of space for the allocation of employees and in turn did not perform optimal allocation of employees to a particular task.

Thus these techniques were not able to perform both scheduling and allocation together. This in turn degrades the performance of the system and results inefficient allocation. However a more effective approach named Ant Colony Optimization (ACO) [2] solved the issues of project scheduling and human allocation. It maintains an employee allocation matrix and task list for scheduling and allocation purpose. Also it takes into consideration the various parameters such as project cost, workload, schedule, resources.

The main advantages of this approach are:

ISSN: 2321-8169
1724 - 1727

1. Provides flexibility.
2. Reduces the search space and speeds up the search process
3. It provides a practical way of solving the problem.
4. It gives a optimal solution.

The detailed information of the algorithm is discussed in the next section.

II. ANT COLONY OPTIMIZATION (ACO)

In this paper, we develop a practical and effective approach for human resource allocation problem in software project planning with an Ant Colony Optimization (ACO) [2] algorithm. The underlying idea of ACO is to stimulate the foraging behavior of ants. It has been observed that the ants deposit a certain amount of pheromone on their path while travelling from their nest to the food. Pheromone serves as a medium for ants to communicate with each other. By sensing the concentration of pheromone, other ants can follow the path to find the food. Inspired by the behavior of ants, ACO was designed by Italian author Dorigo et al. The proposed approach considers the allocation of employee with various skills and their availability.

Software development is a people-intensive activity. In software development, a project schedule is an assignment of the tasks to the employee by considering all the constraints of the project to get the best optimal solution with minimum cost and duration. For employee allocations and project scheduling following attributes are considered.

- Let P be a Project. Each project has a set of tasks.
- Let t be a set of tasks:
  - $t = \{t_0, t_1, \dots, t_{n-1}\}$ Where n is the total number of tasks.
- Let e be a set of employees:
  - $e = \{e_0, e_1, \dots, e_{m-1}\}$ Where m is the Total number of employees
.
- Let s be a set of skills:
  - $s = \{s_0, s_1, \dots, s_{j-1}\}$ Where j is the total number of skills.
- Let st be the set of skills required to perform each task, it is represented as:
  - $st = \{\{st_0, st_1, st_6, st_9\}_0, \dots, \{st_4, st_6, st_8\}_{n-1}\}$ where n is the number of tasks.
- Let $effort$ be the effort required for the each task in T
  - $effort = \{effort_0, effort_1, \dots, effort_{n-1}\}$
  where n is the number of tasks.
- Let se be the set of skills possessed by each employee :
  - $se = \{\{se_0, se_1, se_6, se_4\}_0, \dots, \{se_4, se_6, se_8\}_{m-1}\}$ where m is the number of employee
- Let $skill\_rate_i^{\delta}$ be the rating of the skills possessed by each employee, where δ is skill number and i is employee number.
- Let bs be the basic monthly salary of each employee:
  - $bs = \{bs_0, bs_1, \dots, bs_{m-1}\}$
- Let os be the per-hour over time salary of each employee:
  - $os = \{os_0, os_1, \dots, os_{m-1}\}$
- Let nh be the normal monthly working hours of each employee:
  - $nh = \{nh_0, nh_1, \dots, nh_{m-1}\}$
- Let $maxh$ be the monthly maximum working hours of each employee:
  - $maxh = \{maxh_0, maxh_1, \dots, maxh_{m-1}\}$
- Take $start_j$ date and end_j date of each task j

Construction of task list and employee allocation matrix

1. Calculate the hours wh_j required to complete each task j from the $start_j$ date and end_j date.
2. Estimate the maximum headcount for each task - $maxhead_j$ using COCOMO model [4].
3. Given the start date and end date of a task, calculate the number of hours an employee can dedicate to the task.
4. The algorithm performs resource loading i.e. total hours of work assigned to the employee is less than $maxh_i$
5. Construct a task list of tasks $t = \{t_0, t_1, \dots, t_{n-1}\}$ that satisfies the precedence constraint defined by TPG [2]. Here we first define the pheromone and the heuristic for task list construction.
6. Since in software project planning problem, one task is assigned to a several employees, one employee can undertake several tasks and skill proficiency is considered, the pheromone model adopted by the algorithm is absolute position model with the summation rule.
7. Absolute position model defines the pheromone of putting a task t_j at the k th position of the task list as $\tau_t(j,k)$ [2]. The summation rule uses the sum of the pheromone value of putting a task $t_j$to the position 1 to k instead of simply using $\tau_t(j,k)$ in the selection [2].
8. The minimum slack heuristic (MINSLK) [6], [7] is adopted for constructing the task list.
9. To estimate MINSLK, estimate the shortest possible make span of t_j. Select $maxhead_j$ number of most proficient employees to perform task t_j and assume these $maxhead_j$ employees dedicate all their working hours to task t_j.With this assumption, the shortest make span of t_j can be estimated.
10. Based on that, the earliest start and end date for each task can be evaluated, and the MINSLK value is calculated by the difference between early start [6]. Then the heuristic for task t_j denoted as $h_t(j)$ is given by [2] :

$$h_t(j) = 1/MINSLK_j \qquad \dots (1)$$

11. To construct a feasible task list, an $eliglibleSet$ of the task that satisfy the precedence constraint is maintained [2].
12. For $k = 1$ to n, probability $\Pr(j,k)$ of selecting the task t_j to the kth position using the roulette wheel selection scheme is given by [2]:

$$\Pr(j,k) = \begin{cases} \frac{\sum_{l=1}^{k} \tau_t(j,l)\, h_t(j)}{\sum_{t_u \in eligibleSet} \sum_{l=1}^{k} \tau_t(u,l).h_t(j)} & if\ u \in eligibleSet \\ 0, & \text{otherwise.} \end{cases} \quad \text{….. (2)}$$

13. Probability $\Pr(j,k)$ is used to build the feasible task list.
14. Fetch the skills required to perform each task j from st.
15. Compare the skills required by each task t_j to the skills possessed by each employee i and prepare a $Eligible$ employee list
16. Based on the skill_rate, the proficiency $prof_{ij}$ of ith employee for task t_j can be evaluated by [2] :

$$prof_{ij} = \prod_{id \in ST_j} \frac{s_i^{id}}{5} \quad \text{…. (3)}$$

17. Calculate per hour working salary for each employee.
18. The heuristic of choosing the ith employee to work for the task t_j is denoted as $h(i,j)$. We define $h(i,j)$ as [2] :

$$h(i,j) = prof_{ij}/hs_i \quad \text{…. (4)}$$

19. For construction of employee matrix, the pheromone of selecting ith employee for task t_j is denoted by $\tau_{e1}(i,j)$.
20. Evaluate the value of $\tau_{e1}(i,j).h_e(i,j)^{\beta}$ for all the employees, where β is a parameter whose value is considered as 5 [2].

    Select an employee u from $Eligible$ employee list. The probability $\Pr(u,j)$ of selecting the employee u for task t_j is given by [2]:

$$\Pr(u,j) = \begin{cases} \frac{\tau_{e1}(u,j).h_e(u,j)^{\beta}}{\sum_{i \in Eligible} \tau_{e1}(i,j).h_e(i,j)^{\beta}} & if\ u \in Eligible \\ 0, & \text{otherwise.} \end{cases} \quad \text{…. (5)}$$

21. Employees having first three highest probabilities $\Pr(u,j)$ are suggested for allocation.

Pheromone Management

Initially all pheromone values are set to $\tau_{initial}$, where [2]

$$\tau_{initial} = \frac{1.0}{\sum_{i=1}^{m}[bs_i + maxsalary_i].(deadline + n)} \quad \text{… (6)}$$

Here project is delayed by n time periods (months) with respect to predefined $deadline$ and $maxsalary_i$ is the maximum possible salary [2] of the ith employee, which can be calculated by

$$maxsalary_i = bs_i + os_i * overtime\ working\ hours_i \quad \text{…. (7)}$$

where $overtime\ working\ hours_i = maxh_i - nh_i$ …. (8)

While constructing the employee allocation matrix, immediately after the employee selection is done the pheromone values are updated so that in the next iteration there are better chances of selecting other employees. In this way algorithm manages to maintain the diversity. The pheromone is updated by [2]

$$\tau_{e1}(i,j) = (1-\rho).\tau_{e1}(i,j) + p.\tau_{initial} \quad \text{….. (9)}$$

where ρ is a parameter, at the end of each iteration additional pheromone is deposited to the components associated with the best-so-far plans found by the algorithm. All the pheromone values associated with the best-so-far plan are updated by [2]

$$\tau_{e1} = (1-\rho)\,.\tau_{e1}(i,j) + \rho\,.\Delta\tau \quad \text{…. (10)}$$

where $\Delta\tau = 1/f$ is the reciprocal of the objective function value of the plan. More pheromones will be added to the best-so-far components which make them more attractive for later iterations.

IV. ACKNOWLEDGEMENT

We are grateful for the support and guidance of our mentor Pallavi Kulkarni for the in-depth knowledge given to us in the implementation of the project and providing the external support needed for the technical papers and also guiding us through various hurdles and leading us in the correct direction and bearing with us throughout the building of the project.

V. CONCLUSION

The Ant Colony Optimization approach studied in this paper provides solution for both projects staffing and scheduling problem. Other techniques like Evaluation of human Competences and Human resource Management (ECHRM) considered only employee allocation problem and Evolutionary Technique based on Run Time Analysis (ERTA) considered only projects scheduling problem. In addition, experimental results show that ACO manages to yield better plans with lower cost, complicated planning and more stable workload assignments compared with other existing approaches.

VI. REFERENCES

[1] Hazeyama, A.; Komiya, S., "A software project management system supporting the cooperation between managers and workers-design and implementation," *Computer Software an Applications Conference, 1994. COMPSAC 94. Proceedings., Eighteenth Annual International* , vol., no., pp.44,50, 9-11 Nov 1994 doi: 10.1109/CMPSAC.1994.342834

[2] Wei-Neng Chen; Jun Zhang, "Ant Colony Optimization for Software Project Scheduling and Staffing with an Event-Based Scheduler," *Software Engineering, IEEE Transactions on* , vol.39, no.1, pp.1,17, Jan. 2013

[3] Minku, L.L.; Sudholt, D.; Xin Yao, "Improved Evolutionary Algorithm Design for the Project Scheduling Problem Based on Runtime Analysis," *Software Engineering, IEEE Transactions on* , vol.40, no.1, pp.83,102, Jan. 2014

 ISSN: 2321-8169
1724 - 1727

[4] B. Boehm et al., Software Cost Estimation with COCOMO II.Prentice-Hall, 2000.
[5] A. Bauer, B. Bullnheimer, R.F. Hartl, and C. Strauss, "Minimizing Total Tardiness on a Single Machine Using Ant Colony Optimization," Central European J. Operations Research and Economics,vol. 8, no. 2, pp. 125-141, 2000.
[6] R. Kolisch and S. Hartmann, "Heuristic Algorithms for Solving the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis," Handbook on Recent Advances in Project Scheduling, J. Weglarz, ed., pp. 197-212, Kluwer, 1999.
[7] L. Ozdamar, "A Genetic Algorithm Approach to a General Category Project Scheduling Problem," IEEE Trans. Systems, Man, and Cybernetics-Part C: Applications and Rev., vol. 29, no. 1,pp. 44-59, Feb. 1999.