STUDY OF PROJECT SCHEDULING AND RESOURCE ALLOCATION USING ANT COLONY OPTIMIZATION

[1]Prajakta Joglekar, [2]Pallavi Jaiswal, [3]Vandana Jagtap
Maharashtra Institute of Technology, Pune
Email: [1]somanprajakta@gmail.com, [2]pallavijaiswal2330@gmail.com, [3]vandana.jagtap@mitpune.edu.in

Abstract: Task Scheduling and Resource Allocation is an important phase in Project Management. Scheduling Problem in Software Projects is NP-complete and various algorithmic approaches have been studied to develop an optimal solution to generate a schedule, which will be cost effective, utilize resources effectively and meet the target deadline. This paper addresses the above problem in detail and reviews how Ant Colony Optimization (ACO) approach is being used for Resource Allocation and Task Scheduling. We attempt to propose a modified objective function to minimize cost, based on the rate card of employees in a company.

Keywords- scheduling; resource; project; ACO

I. INTRODUCTION

Project Management is very important in the life cycle of a software project. As with any other project, Planning and Scheduling of Tasks and Resources becomes an important step in Software Project Management. The Project Manager, who is responsible for planning tasks and allocation of resources, uses tools like Microsoft Project to develop a schedule. These tools serve as an aid to develop the plan; they do not propose an intelligent solution for resource allocation to meet the deadline. The Manager has to use his experience and knowledge to generate an optimal schedule. Here, the use of Artificial Intelligence in developing automated schedules is proving to be beneficial as it minimizes human errors as well as saves the Project Manager's time. This paper does a survey of how Artificial Intelligence can be or is being used to automate the scheduling of Human Resource Allocation to Software Projects.In the next sections, we first define the scheduling problem at hand, its objectives, followed by a Mathematical model and description of the problem. We then describe the Ant Colony Optimization Algorithm and how it can be used to solve Resource Allocation and Scheduling problem. We then give brief description of other techniques used in Scheduling Problems in comparison with Ant colony optimization approach.

II. PROJECT SCHEDULING AND RESOURCE ALLOCATION PROBLEM

A project scheduling problem consists of defining which resources are used to perform which task and what is the duration of each. The objectives of the Project Scheduling Problem are minimization of project duration, minimization of project cost and maximization of product quality.

Software Project Scheduling can be compared with the famous Job Shop Scheduling Problem. Here jobs are software tasks like Requirement Gathering, Analysis and Design, Coding, Testing, Deployment etc. and Resources are actual human resources who are to be allocated to the tasks. Job Shop Scheduling is known to be NP-Hard Problem [1]. Since Resource Allocation in software project can be reduced to Job Shop Scheduling, Resource Allocation also is NP- Complete.

Similar to this problem is the Nurse Scheduling Problem or Nurse Rostering Problem of finding an optimal way of assigning nurses to shifts, with a given set of constraints. The approach to solve the Nurse Scheduling Problem can also be applied to our Resource Allocation and Scheduling Problem in Software Projects.

There are various algorithms used to find feasible solutions for different types of Scheduling problem. Neighbourhood Search Methods like Tabu Search, Simulated Annealing Method, Hill Climbing, Ant Colony Optimization as well as Genetic Algorithms have been used to find feasible solution to scheduling problems from various industries. This paper particularly does a study of how Ant Colony optimization is being used for Software Project Resource Allocation to automate the planning and scheduling phase.

III. MATHEMATICAL MODEL AND DESCRIPTION OF RESOURCE ALLOCATION AND TASK SCHEDULING PROBLEM

A software project is a sequence of tasks in predefined order.

Let T be a set of tasks (t_1, t_2, t_3…..t_j)

Let R be a set of n resources available for the project (r_1, r_2,r_4 ….., r_n)

For i^{th} employee, the following attributes are considered:

- d_i - Dollars per hour. In the approach paper by Chen and Zhang [2], they have considered the basic salary as well as salary per hour. In practice, the basic salary of an employee is a sensitive data, maintained by HR team of the company. It is not easily available to the Project Manager who is going to schedule the project plan. Instead the company has a rate card designed for each designation and technology which is known to the client as well. The Manager can pick up the value of d_i on the basis of the designation of the employee and his technology. E.g. a Java Programmer is charged at $x per hour while a Testing Team Leader is charged at $y per hour.
- h_i- Total working hours per month
- [$join_i$, $leave_i$] – The time window when the resource is available for the project
- {$s1_i$, $s2_i$, $s3_i$…..sp_i} - The skill list of i^{th} employee. This i^{th} employee has p skills and s_{ji} ϵ [0, 5] is the proficiency score of that employee for j^{th} skill. Here the skills include C programming, HTML, Testing, Architecture and many more.

A. *Project Task List:*

Some tasks may be dependent on others. A commonly used technique to represent tasks is using a Task Precedence Graph (TPG). A TPG is an acyclic directed graph G(T,A). The set of nodes T{t1; t2; . . . tn} corresponds to the set of tasks, where n is the number of tasks in the project. The set of edges A represents the precedence relations among tasks. An edge (i,j) ϵ A means t_i is a direct predecessor task of t_j.Under the precedence constraint defined by the TPG, a task can only start when all of its direct predecessor tasks have finished.

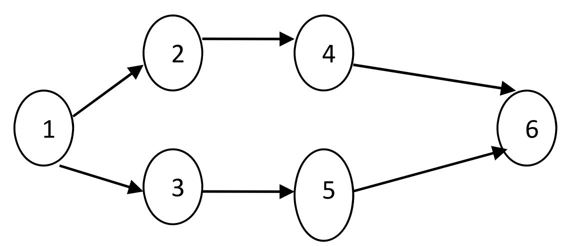

Fig.1 Task precedence graph (TPG)

In the above TPG (Fig.1), the nodes labelled 1 to 6 are tasks in software project

Example: 1: Requirements, 2: Design, 3: Test Case Writing, 4: Coding, 5: Test Case Review, 6: Testing

For each task t_j in set T, the following attributes are considered:

- e_j – work effort estimated for the task, in person hours. Any of the known models like COCOMO model can be adopted for work estimation [2].

- SK_j – set of skills required for the task j

Using this, the proficiency $prof_{ij}$ of i^{th} employee for the j^{th} task can be found out as:

$$prof_{ij} = \prod_{id \epsilon SKj} \left(\frac{s_i^{id}}{5}\right) \quad \text{……….} (1)$$

The proficiency score is assumed to be between 0 and 5, hence the above equation.

IV. PLANNING OBJECTIVE

A Software Plan must indicate the following:

- Start and End dates of the Project
- Total duration of the project
- Start and End dates of each task t
- Duration of each task t
- Mapping of resource j to task i with the number of hours (workload)

The Plan must satisfy the following constraints:

- The order of the tasks must follow the constraint defined by its Task Precedence Graph.
- The working hours w allocated for i^{th} resource must not exceed his/her total working hours per month i.e. h_i

$$\sum_{j=1}^{n} w^t{}_{ij} < h_i \text{ …..} for\ t = 1,2,3 \dots n \quad \text{….}(2)$$

- All the tasks have to be completed in the time window.
- There should be some objective function for the plan. Here, we consider **Cost Minimization** as the objective function.ie. If resource i is paid d_i dollars per hour and his working hours are as mentioned above, then the objective function of the project can be stated as:

$$\textbf{min f} = \sum_{i=1}^{r} \sum_{j=1}^{n} \boldsymbol{w^t{}_{ij} * di} \text{ .. } \textbf{(3)}$$

where r = total resources on the project

Constructing a task list and employee allocation matrix are common ways to schedule task and allocate resources. However, in practice, these 2 activities cannot be treated as separate. This is because allocation of a particular resource and his availability will decide the start and end duration of the task and not just the task precedence graph.

V. ANT COLONY OPTIMIZATION APPROACH (ACO)

This section studies the ACO approach used to solve Scheduling Problems.

ACO takes inspiration from the behaviour of ants. Ants walk to and from a food source and deposit pheromone on their path so that other members can sense it and follow their path. The ants which lead the way, traverse different paths, reach the food source and come back to their nest. The ant which happens to pick the shorter path, will return early and hence pheromone concentration on that path will be more, as it is deposited while going as well as coming back. The other ants tend to follow the path where pheromone concentration is higher. In other words, ants eventually find the shortest path between food source and their nest and in doing so, they keep depositing pheromone on that path so that other ants also take up the same path. Through this mechanism, ants can transport food from source to nest in less time and less efforts. This behaviour inspired the researchers to devise the ACO technique.

Ant Colony Optimization algorithms are being used by companies for real-world applications. One such application area is different types of scheduling problems: job shop scheduling, course timetabling, nurse scheduling and project scheduling [3].

Scheduling Problems are actually optimization problems wherein an objective function has to be optimized. In our case, we have to plan a schedule such that total cost of the project is minimized. ACO has been formalized into a metaheuristic for optimization problem by Dorigo.

In order to apply ACO to a given optimization problem, following model is given by Dorigo [3]:

A model P =(S, Ω, f) of combinatorial optimization problemconsists of:

- Search space S defined over a finite set of discrete decision variables Xi, i = 1 to n;

- A set Ω of constraints among the variables;
- An objective function f to be minimized, in our case, it is the cost function
- A feasible solution s ∈ S is a complete assignment of values to variablesthat satisfies all constraints in Ω.
- A solution s∗∈ S is called a Global Optimum iff: f (s∗) ≤ f (s) ∀s ∈ S.

In our problem, we can consider S to be different tasks in the project related by their precedence (as per TPG). Ω is the list of constraints that the plan has to follow. These include task related as well as resource related constraints mentioned earlier. Objective function is the cost function of the project which should be minimum. This model of ACO is used to associate pheromone value with each solution component. i.e. if a solution component c_{ij} consists of assigning task t_i to employee j, then a pheromone value τ_{ij} is associated with c_{ij}.

For the selection of an activity the ant uses heuristic information as well as pheromone information [4]. These two are indicators of how good it seems to put activity i at place k of the activity list for the project, as also how good it is to select resource r_j instead of r_m for task i.

Ant Colony Optimization Algorithm [3]:

```
Set parameters, initialize pheromone trails.
While Termination Condition not met do

        ConstructAntSolutions
        ApplyLocalSearch (optional)
        UpdatePheromones

EndWhile
```

There are 3 main procedures in ACO Algorithm:

A.Solution construction

During each iteration, a group of ants start to build solutions. Each ant selects components (c_{ij}) to build the solution. The components are selected on the basis of pheromone and heuristic information. The components belonging to the best solutions found by previous ants usually accumulate more pheromone.

Solution construction consists of constructing a Task List followed by construction of Employee Allocation Matrix.

Minimum Slack Heuristic [5] is used for task list construction. A task with relatively smaller Minimum Slack means it is more urgent. Therefore, heuristic H(j) for task j is given by

$$H(j) = 1/ \text{minSlack}(j) \quad \text{…….(4)}$$

For Employee Allocation Matrix construction the heuristic used is

$$H(i,j) = \text{prof}_{ij}/ d_i \quad \text{………….. (5)}$$

This means that an employee whose rate (dollars per hour) is less but proficiency is more for a given task j is more desirable for that task.

B. Pheromone Management

Pheromone values are updated according to the performance of the solutions built by ants. The aim of this step is to increase the pheromone values associated with good solutions and to decrease those associated with bad solutions. This is done by first doing pheromone evaporation, followed by increasing pheromone levels associated with good solution.

Initially, all pheromone values are set to an initial value $\tau_{initial}$, where

$$\tau_{initial} = \frac{1}{\sum_{1=i}^{r} d_i * tp} \quad \text{……(6)}$$

The above value is assumed for a time period tp. The denominator is an upper bound of cost function estimation, assuming that all r employees work for all hours of period tp, with i^{th} employee paid at the rate of \$d per hour. In other words $\tau_{initial}$ becomes the lower bound of all pheromone values.

C. Apply Local Search

This is an optional step undertaken to improve performance by diversifying search. Here, the pheromone concentration on traversed edges is reduced to encourage other ants to choose different edges.

Study of Experimental Results:

ACO Optimization to Resource Allocation and Scheduling Problem gives promising results, according to the work done by Chen and Zhang[2] and Merkle[4]. ACO algorithm has been proved to work better than Genetic algorithm in many cases. The heuristics applied here are more practical and as observed in software industry. When used with an event based scheduler, ACO technique gives more accurate results and modifies the plan only when employee joins or leaves the project or when a task is finished.

VI. OTHER ALGORITHMS FOR SOFTWARE RESOURCE ALLOCATION AND TASK SCHEDULING PROJECTS

Jeffcoat and Bulfin (1993) [06] applied Simulated Annealing (SA) to resource-constrained scheduling problem (RCPSP). While SA is relatively easy to code, lot of time is needed to compute quality solutions. The constants used in the heuristic of SA have major effect on the quality of the results, hence have to be fixed with good precision. Tabu Search [01] is similar to SA, however, it generates many mutations at a time beforemoving to the optimal solution. It records the bad solutions in memory in order to avoid traversing them in future. Due to this, it avoids getting trapped into local minima. Genetic algorithm (GA) is an optimization technology based on Darwin's theory of natural selection and mutations. When GA is combined with heuristic methods, it works well for scheduling problems, where less optimal solutions are acceptable. Hill Climbing works as efficiently as Simulated Annealing when random restart variant of hill climbing is used. In general, there exists a trade-off between run time and solution quality in all these approaches.

VII. CONCLUSION AND FUTURE WORK

A Software Project Scheduling and Resource Allocation Problem is comparable to Job Shop and other known NP-hard scheduling problems. Though these problems are difficult to solve, they impact the ability of the project to meet its deadline and satisfy the client's demand. In today's industry, where customer satisfaction is the motto, use of optimization techniques in planning phase would aid the software manager to meet the target on time while keeping the cost of the project minimum. As Ant Colony Optimization shows promising results for our scheduling problem, it can certainly be used in building tools for Resource Allocation and Project Scheduling.

The next step is to survey and study if any such tools are available in IT industries which use ACO and solve the Resource Allocation and Scheduling Problem in Software Project Planning. The comparative study of such tools will help us understand their advantages and limitations. An equivalent tool can then be built which should overcome these limitations while still satisfying the objective function. Additional constraints can be included in the heuristic function to cater to any other practical difficulties faced by the Project Managers.

REFERENCES

[01] Albert Jones and Luis C. Rabelo," Survey of Job Shop Scheduling Techniques" Wiley Encyclopedia of Electrical and Electronics Engineering (DOI: 10.1002/ 047134608X. W3352)

[02] Wei-Neng Chen, Member, IEEE, and Jun Zhang, Senior Member, IEEE, "Ant Colony Optimization for Software Project Scheduling and Staffing with an Event-Based Scheduler", *IEEE Transaction Jan13*.

[03] Dorigo, Birattari,Stutzle," Ant Colony Optimization",*IEEE magazine 2006.*

[04] Daniel Merkle, Martin Middendorf, and HartmutSchmeck, "Ant Colony Optimization for Resource-ConstrainedProject Scheduling, *IEEE Transaction Aug2002.*

[05] Russel and Norvig,"Planning and Acting in the Real World", Artificial Intelligence – A modern approach, 3rd edition.

[06] David E.Jeffcoat and Robert L.Bulfin ,"Simulated annealing for resource-constrained scheduling" (*ScienceDirect, doi:10.1016/0377-2217(93)90231-B)*

[07] Alba and Chicano," Software Project Management with GAs",doi:10.1016/j.ins.2006.12.020.

[08] "Ant Colony Optimization Algorithms", https://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms.

[09] Antoniol, Massimiliano and Harman,"Search–Based Techniques Applied to Optimization of Project Planning for a Massive Maintenance Project. (*21st IEEE International Conference on Software Maintenance)*

[10] Cheng, Ozaku, and Kuwahara,"Simulated Annealing Algorithm for SchedulingProblem in Daily Nursing Cares", *IEEE International conference, Oct 2008*