ISSN: 2321-7782 (Online)
Volume 2, Issue 3, March 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Paper / Case Study
Available online at: www.ijarcsms.com

Software Project Planner with Ant Colony Optimization

Supriya H. Kale[1]
BE Computer
Dr.D.Y.Patil Institute of Enginnering & Tech
Ambi, Talegaon-Dabhade – India

Pallavi P. Malavadkar[2]
BE Computer
Dr.D.Y.Patil Institute of Enginnering & Tech
Ambi, Talegaon-Dabhade – India

Sumaiyya R. Kagale[3]
BE Computer
Dr.D.Y.Patil Institute of Enginnering & Tech
Ambi, Talegaon-Dabhade – India

Arti S. Narwade[4]
BE Computer
Dr.D.Y.Patil Institute of Enginnering & Tech
Ambi, Talegaon-Dabhade – India

Seema R. Salgar[5]
Lecturer, Computer Engg
Dr.D.Y.Patil Institute of Enginnering & Tech
Ambi, Talegaon-Dabhade – India

Abstract: ***Software project planner (SPP) is NP-hard problem faced by project manager in software industry.SPP is the problem of employee and task scheduling which is important and challenging part of industry. It is NP-hard because there is increasing number of task and employee. For solving NP-hard problems only few algorithms are available but the output is not satisfactory. To solve SPP one meta-heuristic technique is used i.e. Ant Colony Optimization. ACO contains heuristic and pheromone values. Heuristic is calculated with the help of previous available solution, dependency of tasks, employee and efforts. There are six heuristic techniques are available from that we use allocation and dedication to solve our SPP problem. As compared to other genetic algorithms meta-heuristic provide more accuracy.***

Keywords: ***SPP-Software Project Planner, ACO-Ant Colony Optimization, GA- Genetic Algorithm, RCPSP-Resource Constraint Project Scheduling Problem.***

I. INTRODUCTION

It is challenging and critical task to develop the satisfactory tool for project management. Project management is one of the application which requires skills, knowledge, tools and techniques to solve the scheduling problem .In industry scheduling of project is very critical task in project development. SPP problem is related to NP-hard problems[5].Generally SPP is used to find an optimal data arranging the human resources i.e. software engineers to perform one or more task with minimum salary and duration[2].Thus automatically scheduling a software with high performance is very helpful to software project manager. SPP problem is similar to RCPS problem which is used to find an optimal solution that meets the precedence requirements and minimize the project cost and duration [2].There is some difference between SPP problem and RCPSP. In SPP problem employee is considered with multiple skills and in RCPSP numbers of resources are considered. The cost related with employee is optimized in SPP problem in addition to the project duration minimization objective in RCPSP. To solve SPP problem classical meta-heuristic approach i.e. genetic algorithm is used.

Following models have been designed before ACS for S(Software Project Scheduling).

1. RCPSP model:

To solve NP-hard problem of project management RCPSP is first used. It is used to manage the tasks of project related to precedence and resource constraints. To make a schedule it need to find order of tasks which satisfy task precedence constraint

and generate task list. RCPSP solve the problem of project scheduling but it does not consider the employee allocation matrix. The RCPSP is not a satisfactory model to solve the problem of project scheduling.

2. Employee allocation model:

In this model the problem of how to assign employees to different tasks is find out. The main objective is to minimize the number of constraint violations or to minimize project duration and cost. In this the software project planning have to assume that the task can be done or conducted by an unlimited number of employees and an employee can be assigned to an unlimited number of tasks at a time, which is usually not possible.

3. Multi-skill scheduling model:

The model is same as the RCPSP for task scheduling and regards different combinations of employees as different alternative modes for the implementation of a task. This model solve problem of employee allocation matrix as well as task scheduling this model only consider the allocation of employee to task at only one time, but they does not consider the pre-emption of task . This model reduces the allocation flexibility of the human resources. In this model if one of the employee is busy in other task that time the whole team has to wait till the employee is released. This is the drawback of this model and it reduces the efficiency of the project.

4. The time line based model:

The new technique that combines both the human resource allocation and task scheduling is a time line based model. This model generates the time line axis to solution representation and makes a possible plan. This model has two drawbacks. First, assign workload period by period instead of task by task and second, the plans produced by this model may assign two completely different groups of employees to the same task in different periods.

II. ARCHITECTURAL FLOW

The following diagram shows the flow of data and transformation applied on that data.

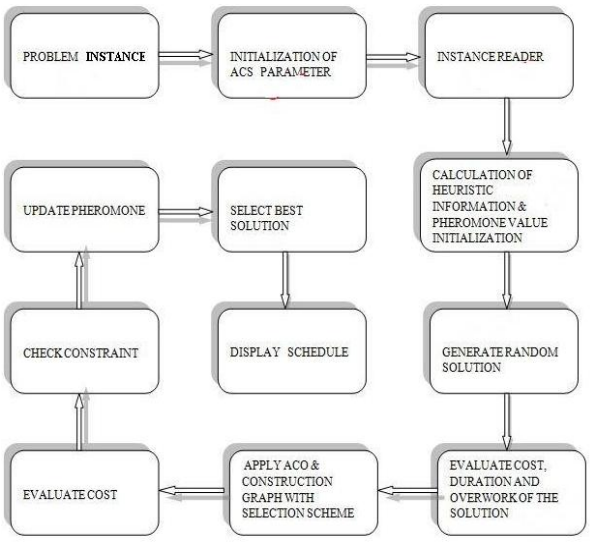

Fig 1.Block Diagram of SPP Problem

III. AOC ALGORITHM

1. Initialize all parameters i.e. Q0, Ngen, Nant which are used in ACO. These parameters are used to evaluate the importance of Heuristic information and history, which also adjust the pheromone updating, balance the behavior of ants. Nant is number of ants and Ngen is number of generations.
2. Initialize all pheromone value as 0.

3. Each ant selects her own path for finding solution. Each ant select next node as per selection scheme and fill the matrix. When travelling of ants is completed Solution matrix is constructed.
4. By using fitness function evaluates the quality of solution, also calculate the cost, duration of project and overwork for that project.
5. Compare the solutions and select the best one, update the pheromone value.
6. Repeat the procedure till condition is satisfied. Generally termination condition is determined by fix number of generation.
7. Select and display the best solution whose cost and duration is less.

IV. CONCLUSION

Software Project Planner Problem is NP-Hard problem. Employee and task scheduling is not simple task because its time complexity is very high. Due to this there are some mete heuristic algorithm implemented to solve such problem. Genetic algorithms and tabu search is used to solve the problem.GA is the popular method used to solve NP-Hard problem.GA is used to solve SPP Problem but the success rate is not satisfying. To solve the problem of scheduling different heuristic techniques compared so we get that AD (allocation dedication) is the best method to achieve best solution. Comparing with GA, SPP Problem gives following result-

1. ACO has good graph-based search problem solving capability by splitting task and distributing employee dedication to that tasks. Construction graph is generated based on that and naturally SPP Problem is converted into graph-based search problem.
2. ACO supports to use heuristic information to increase the search ability of ants. There are total six heuristics are used in SPPP-ACS, including total dedication of employee, allocation dedication and importance of tasks.ACO gives satisfactory solution as compare to other.

References

1. Wei-Neng Chen and Jun Zhang, "Ant Colony Optimization for Software Project Scheduling and Staffing with an Event-Based Scheduler" IEEE Transaction on Software Engineering, JAN 2013.
2. JingXian, Xian-TingAo, YongTang, Solving software project scheduling problems with ant colony optimization, May 2012.
3. Marco Dorigo, Mauro Birattari, and Thomas St¨utzle, Ant Colony Optimization Artificial Ants as a Computational Intelligence Technique.
4. WANG Lu, WANG Zhiliang, HU Siquan, LIU Lei, Ant Colony Optimization for Task Allocation in Multi-Agent Systems.
5. C. Blum and M. Sampels, "An Ant Colony Optimization Algorithm for Shop Scheduling Problems," J. Math. Modelling and Algorithms, vol. 3, pp. 285-308, 2004.

