International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org
Volume 4, Issue 1, January 2015 **ISSN 2319 - 4847**

A Proposed Scheme for Software Project Scheduling and Allocation with Event Based Scheduler using Ant Colony Optimization

Arjita sharma[1], Niyati R Bhele[2], Snehal S Dhamale[3], Bharati Parkhe[4]

NMIET, University of Pune

ABSTRACT

In software engineering field, developing software tools is challenging and important. In software project humans are important. Human resources are mainly needed. In software project, planning is important. Since software project is much related to human resource, the human resource allocation is the important problem. A software project planning tool must consider the project planning as well as human resource allocation problem. An Event Based Scheduler (EBS) and an Ant Colony Optimization (ACO) algorithm is used to develop a flexible and effective model for software project planning. The proposed system represents a plan by task list and employee allocation matrix. In the EBS, the beginning time of the project, the time when resources are released from finished tasks, and the time when employees join or leave the project are regarded as events. The basic idea of the EBS is to adjust the allocation of employees at events and keep the allocation unchanged at nonevents. For planning and employee allocation ACO is used. Thus in this paper, with the help of ACO and EBS, the proposed method enables the modeling of resource conflict and task preemption and preserves the flexibility in human resource allocation.

Keywords: Ant colony optimization (ACO), event scheduler, resource allocation, software project planning, project scheduling, workload assignment.

1. INTRODUCTION

There is a rapid development of software industry so software companies are now facing a highly competitive market. To succeed, companies have to make efficient project plans to reduce the cost of software construction. However, in medium to large-scale projects, the problem of project planning is very complex and challenging. Due to the importance and difficulty of software project planning, there is a growing need for developing effective computer aided tools for software project planning in recent years. To plan a software project, the project manager needs to estimate the project workload and cost and decide the project schedule and resource allocation. To plan a software project, the project manager needs to estimate the project workload and cost and decide the project schedule and resource allocation. To build more suitable models and tools, traditional project management techniques need to be further extended. In this paper, a practical and effective approach for the task scheduling and human resource allocation problem in software project planning with an ant colony optimization (ACO) algorithm is proposed. The representation scheme is composed of a task list and a planned employee allocation matrix. The task list defines the priorities of tasks to consume resources, and the planned employee allocation matrix specifies the originally planned workload assignments. In this way, the representation takes both the issues of task scheduling and resource allocation into account. ACO was proposed by Dorigo and Gambardella in the early 1990s and till today has been successfully applied to various combinatorial optimization problems. As ACO builds solutions in a step-by-step manner and enables the use of problem-based heuristics to guide the search direction of ants, it is possible to design useful heuristics to direct the ants to schedule the critical tasks as early as possible and to assign the project tasks to suitable employees with required skills. Therefore, ACO promises to converge fast and perform well on the considered problem. Various ACO variants have been developed. Two of the best performing ACO variants include ant colony system (ACS) and max-min ant system (MMAS). In this paper, the ACS variant to develop the ACO approach is followed to solve the software project planning problem. The main characteristics of the ACS are in two aspects. First, in the solution construction procedure, the ACS applies a pseudorandom proportional selection rule which aggressively biases selecting the components with the maximum pheromone and heuristic values. In this way, the ACS strongly exploits the past search experience of ants and has a fast convergence speed. Second, in the pheromone management procedure, ACS has two pheromone updating rules, namely, the global updating and the local updating.

2. RELATED WORK

In the literature review, several works have been done on developing search-based approaches for software project planning. Duggan et al. [2] and Barreto et al. [3] built models for the staffing problem of software projects and proposed genetic algorithm (GA) approaches. But, their models only focused on staffing and the problem of task

scheduling was not considered. Author Chang, proposed the software project management net (SPMnet) model [4] and the project management net (PM-net) model [5] successively, and then further improved the models to a richer version with a GA [6]. Other GA-based approaches were also proposed in [6] and [10], [11]. In these approaches, a plan is described by a 2D matrix which specifies the workload of each employee on each task. But, as this representation is inadequate for modeling resource conflict, these models all implicitly uses the "Mongolian Horde" strategy [7] that assumes an unlimited number of employees can be assigned to a task and an employee can join an unlimited number of tasks simultaneously, which is usually not the case in practice. Bellenguez and Ne´ron [8], [9] proposed a multiskill scheduling model by extending the traditional RCPSP model. The model considers both the problems of human resource allocation and task scheduling, and takes the skill proficiency of employees and resource conflict into account. Tabu search (TS) [10], branch and bound [10], and GA [11] have been developed for the model. In all of the above-mentioned models, there is an assumption that pre-emption is not allowed. As discussed, this assumption reduces the flexibility of human resource allocation for software projects. Task pre-emption in software projects is only considered in a few studies. In Chang's recent work [3], he improved his previous scheduling model by introducing a 3D matrix representation, specifying the workload assignment of each employee for each task on each time period. Although this representation is much more flexible, it makes the search space very large and suffers from the problem of desultory assignment of workloads. In this paper, an effective approach for project scheduling and human resource allocation problem is proposed. It also considers the uncertain events handling. The proposed method is characterized by two features. Firstly, a representation scheme, event-based scheduler (EBS) is used [12]. The representation scheme is composed of project list, task list and a planned employee allocation matrix. Second, an ant colony optimization (ACO) algorithm is used to schedule the critical tasks as early as possible and to assign the project tasks to suitable employees with required skills. Therefore, ACO performs well on the considered problem.

3. PROPOSED DIAGRAMS

Following are some of the diagram which are designed to develop the proposed system. By studying below diagrams the basic concept of the system will be more clearly explained.

1. Use case Diagram – Refer Figure 1.
2. Component Diagram – Refer Figure 2.

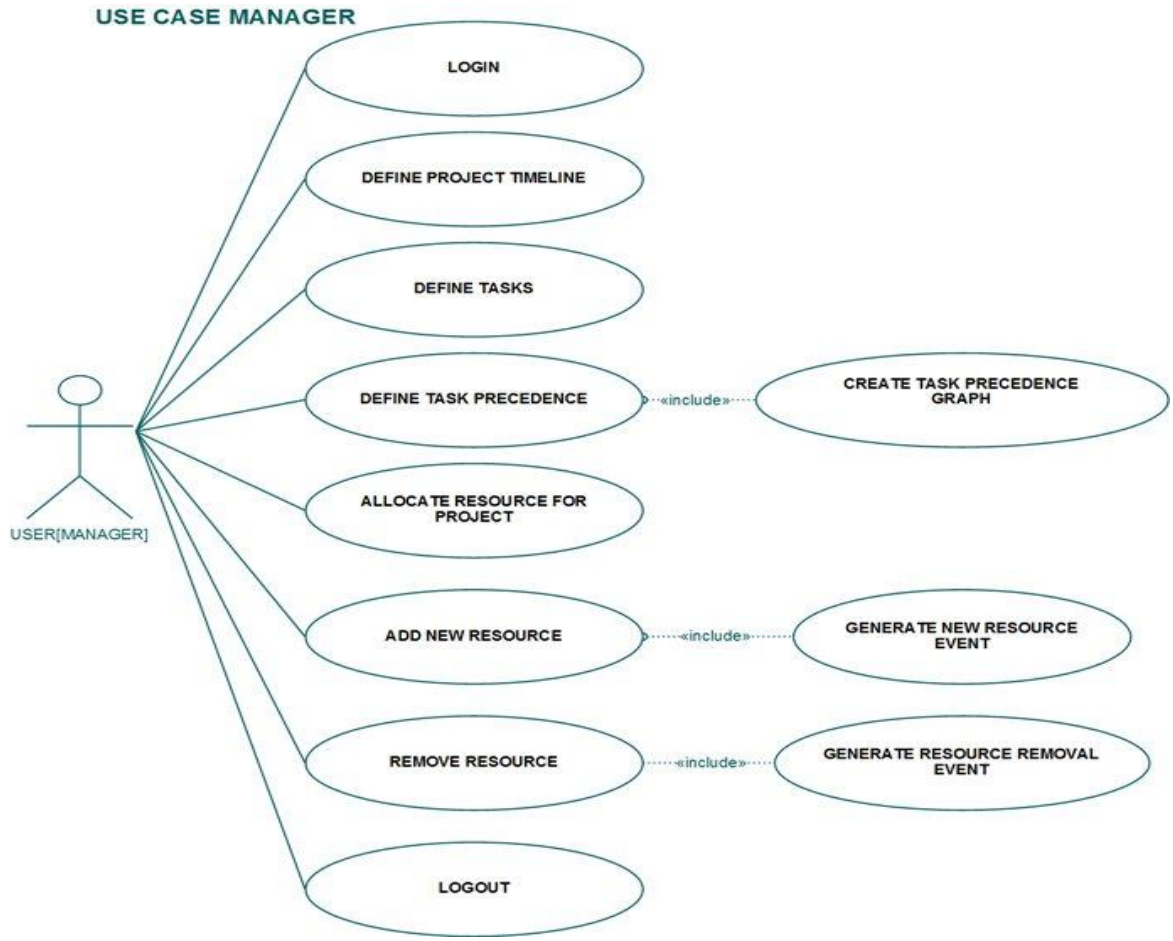

Figure 1. Use Case Diagram

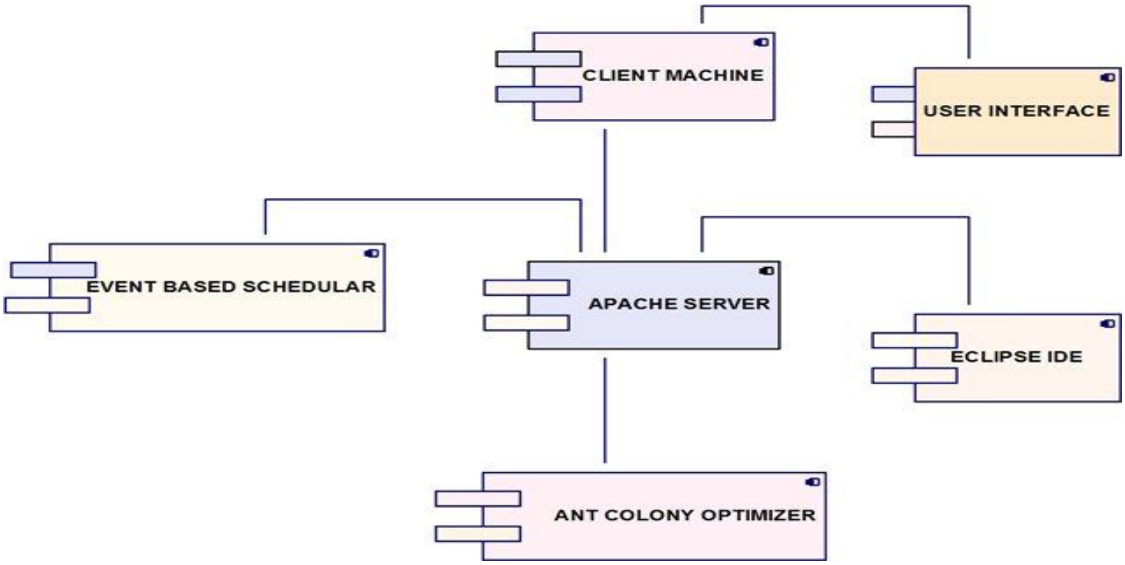

Figure 2. Component Diagram

4.PROPOSED METHODOLOGY

4.1. Event Based Scheduler (EBS)

The basic idea of the EBS is to adjust the allocation of employees at events and keep the allocation unchanged at nonevents. With this strategy, the proposed method enables the modeling of resource conflict and task preemption and preserves the flexibility in human resource allocation.

Following image show the studied pseudo code for EBS implementation:

```
procedure  scheduler      //EBS
01    initialize the number of available human resources during the whole scheduling window;
02    set the beginning time and the time when employees join or leave the project as events;
03    t = 1;
04    while the project is not finished
05        if t is an event
06            find the tasks that can be implemented at time t and arrange these tasks into a sequence seq
                according to the order defined by the task list;
07            while seq is not empty
08                set t_j as the first task in seq and remove t_j from seq;
09                for i=1 to m                //for every employee
10                    if the planned working hours pwh_ij is not larger than the remaining available working
                        hours of the i-th employee at t
11                        wh_ij^t = pwh_ij;
12                    else
13                        wh_ij^t is set to the remaining available working hours of the i-th employee at t;
14                    end if-else
15                end for
16            end while
17            (local refinement: let regular employees devote all of their normal working hours to the project.)
18        else
19            the workload assignments are the same as those at t-1;
20        end if-else
21        evaluate the completion situation of the tasks at time t;
22        if there are any tasks finished at time t
23            set the time t+1 as an event;
24            (local refinement: release redundant working hours of employees.)
25        end if
26        t = t+1;
27    end while
```

4.2. Ant Colony Optimization (ACO)

An ACO algorithm works by dispatching a group of artificial ants to build solutions to the problem iteratively. In general, an ACO algorithm can be viewed as the interplay and the repeated execution of the following three main procedures:

1] Solution construction—During each iteration of the algorithm, a group of ants set out to build solutions to the problem.

2] Pheromone management—Along with the solution construction procedure, pheromone values are updated according to the performance of the solutions built by ants.

3] Daemon actions—Daemon actions mean the centralized operations that cannot be done by single ants.

The different modules of the proposed Ant Colony Optimization approach are described below.

1] Input module: The following data pertaining to the problem are given as input: Number of Tasks (n), number of machines in the shop (m), number of operations Ji of each task i.

2] Initialization module: The number of ants is defined, and the pheromone trails used by them for constructing solutions are initialized. This problem uses two pheromone trails: pheromone trail intensity for route selection gives information about the desirability of choosing route r for operation Oij at iteration tn and pheromone trail intensity, which indicates the desirability of choosing operation Oij directly after the operation Oi'j' is loaded on machine k, is used for task conflict resolution while generating feasible schedule.

3] Solution construction and Evaluation module: Each ant constructs a solution in two stages. In the Ist stage, an ant, at each construction step, allocates an operation of a particular task to one of its available resources. The ants use a probabilistic choice rule which is a function of the pheromone trail and heuristic information based on processing time. In the IInd stage, on allocation of all operations to the machines, each ant generates a schedule based on algorithm.

4] Sorting module: The best solution of the current iteration and the global best are sorted and stored separately.

5] Termination Check module: A specified number of iterations (no_iter) is estimated to terminate the algorithm depending on the size of the problem. Termination directs to the output module; otherwise, continue to the pheromone updating module.

6] Pheromone updating module: At the end of iteration, the pheromone trails corresponding to only one single ant is updated. This ant may be the one which found the best solution in the current iteration or the one which found the best solution from the beginning of the trial.

7] Output Module: This module prints the best solution of the optimal route choices of all operations and schedule for minimum make span time criterion.

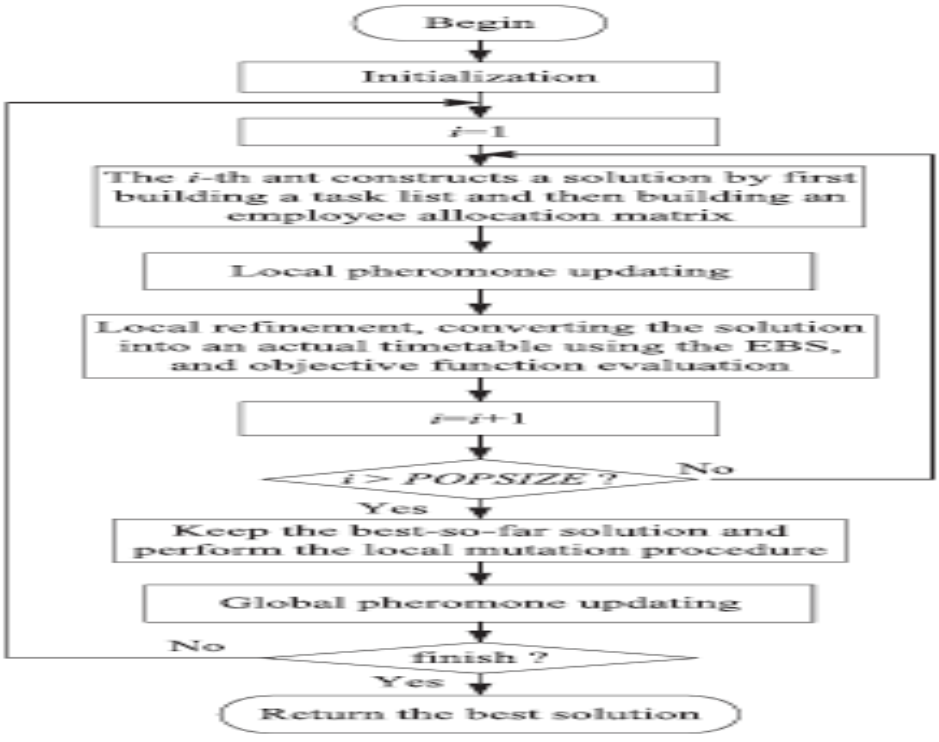

Figure 3. Flowchart of ACO Algorithm

5. CONCLUSION

ACO is a recently proposed heuristic approach for solving hard combinatorial optimization problems. Artificial ants implement a randomized construction heuristic which makes probabilistic decisions. The accumulated search experience is taken into account by the adaptation of the pheromone trail. ACO Shows great performance with the structured problems like network routing. In ACO Local search is extremely important to obtain good results. The proposed algorithm manages to yield better plans with lower costs and more stable workload assignments compared with other existing approaches. In addition, since the model proposed in this paper provides a flexible and effective way for managing human resources, it is promising to apply the proposed approach to other complex human-centric projects like consulting projects.

References

[1] Wei-Neng and Jun Zhang," Ant Colony Optimization for Software Project Scheduling and Staffing with an Event-Based Scheduler" IEEE, JANUARY 2013.
[2] J. Duggan, H. Byrne, and G.J. Lyons, "A Task Allocation Optimizer for Software Construction," IEEE Software, May/June 2004.
[3] A. Barreto, M. de O. Barros, C.M.L. Werner, "Staffing a Software Project: A Constraint Satisfaction and Optimization-Based approach," Computers & Operations Research, 2008.
[4] E. Alba and J.F. Chicano, "Software Project Management with GAs," Information Sciences, 2007.
[5] C.K. Chang, M.J. Christensen, C. Chao, and T.T. Nguyen, "Software Project Management Net: A New Methodology on Software Management," Proc. 22nd Ann. Int'l Computer Software and Applications Conf., 1998.
[6] Y. Ge, "Software Project Rescheduling with Genetic Algorithms," Proc. Int'l Conf. Artificial Intelligence and Computational Intelligence, 2009.
[7] J. Wglarz, J. Jo´zefowska, M. Mika, and G. Waligo´ ra, "Project Scheduling with Finite or Infinite Number of Activity Processing Modes—A Survey," European J. Operational Research, 2011.
[8] O. Bellenguez and E. Ne´ron, "Methods for the Multi-Skill Project Scheduling Problem," Proc. Ninth Int'l Workshop Project Management and Scheduling, 2004.
[9] P. Brucker, A. Drexl, R. Mohring, K. Neumann, E. Pesch, "Resource-Constrained Project Scheduling: Notation, Classification, Models and Methods," European J. Operational Research, 1999.
[10] O. Bellenguez and E. Ne´ron, "A Branch-and-Bound Method for Solving Multi-Skill Project Scheduling Problem," RAIRO-Operations Research, 2007.
[11] L.Ozdamar, "A Genetic Algorithm Approach to a General Category Project Scheduling Problem," IEEE Trans. Systems, Man, and Cybernetics-Part C: Applications and Rev., Feb 1999.
[12] W.N Chen and J Zhang, "Ant Colony Optimization for Software Project Scheduling and Staffing with an Event-Based Scheduler," IEEE Trans. Software Engineering, Jan 2013.