SOFTWARE PROJECT SCHEDULER USING ANT COLONY OPTIMIZATION AND EVENT-BASED SCHEDULER

[1]UTKARSHA PATIL, [2]JAGRUTI PATIL, [3]JAYSHRI PATIL, [4]SONALI WAGHMARE, [5]R.A.DESHMUKH

[1,2,3,4,5]Department of Computer Engineering, Savitribai Phule Pune University, Pune, India
E-mail: [1]patilutkarsha88@gmail.com, [2]patilj64@gmail.com, [3]patiljayshri94@gmail.com, [4]sonaliwaghmare2393@gmail.com, [5]radesh19@gmail.com

Abstract— In software engineering, planning software projects is important and tough assignment. Apart from other field's projects, software projects are people intensive and they mainly require human resources. Thus decent model for software project planning must consider problem of project task scheduling as well as human resource allotment. But both of these two problems are crucial, and existing model has to suffer from very enormous search space and have to curb the flexibility of human resource allotment to facilitate the model. Thus we build an innovative approach with an Event Based Scheduler(EBS) and an Ant Colony Optimization(ACO) algorithm to develop a flexible and decent model for software project planning. This approach serves a plan by task list and a plan employee allocation matrix. In this manner the issue of task scheduling and employee allocation both can be considered. In the EBS, initiation time of project, the time when resources are released from finished task and the time when employee join or leaves the project are view as event. The key concept of EBS is to adjust the allocation of employee at events and retain the allocation untouched at non-events. With this scheme, the proposed method enables modelling of resource strife, task pre-emption and conserves the flexibility in human resource allotment. To fix the planning issue, an ACO algorithm is designed

Keywords— Software project planning, Project scheduling, Resource allocation, Ant Colony Optimization(ACO), Event Based Scheduler (EBS).

I. INTRODUCTION

As software industry is rapidly developing, highly competitive market is being faced by software companies. Therefore, to reduce the cost of software development, companies have to make efficient project plans [1]. Apart from other field's projects, software projects are people intensive and they mainly require human resources. However, the problem of project planning is very tough assignment for medium to large scale projects [2]. As planning software projects is important and tough assignment, there is a growing need to develop computer aided software for planning projects task and employee allocation in recent years [4][5][6]. The project manager needs to estimate project cost and workload for software project planning and then decide the project resource allocation and schedule. The famous model named as COCOMO [7][8] is developed and widely used for workload and cost estimation. For all type of projects, scheduling and staffing management is done by using management tools and techniques. For example, in software project planning, traditional program evaluation and review technique (PERT), the critical path method (CPM) [9], and the resource constrained project scheduling problem (RCPSP).[10] are applied. Though these techniques are good and widely used, they are widely considered to be inadequate for modeling the unique characteristics of today's software projects [11] [12] [13] [14]. The key reason for that is, for software project planning, model technique must consider problem of project task scheduling as well as human resource allotment [4]. But both of these two problems are crucial, and existing model has to suffer from very enormous search space and have to curb the flexibility of human resource allotment to facilitate the model. Different software project tasks require employees with different skills, and skill proficiency of employees mainly influences the efficiency of project execution. Therefore, assigning tasks to the employees to on the basis of their best skills is challenging for software project managers, and human resource allocation has become a challenging part in software project planning. Resource allocation is lacking in the techniques like PERT and CPM and allocation of employees with various skills is not considered by RCPSP. Therefore, existing system considers task scheduling and human resource allocation as two distinct activities and leaves the job of human resource allocation to be done by project managers manually [15]. So it gives inefficient resource allocation and poor performance management. Resources in software projects can usually be allocated in a more flexible way than those in construction or manufacturing projects, as the important resources are humans instead of big machines.

II. SYSTEM ARCHITECTURE

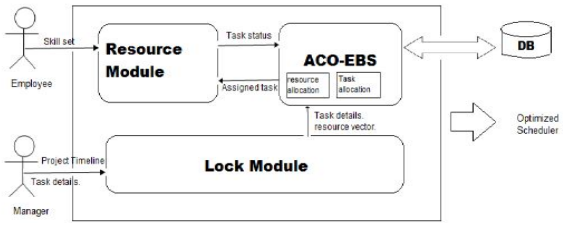

Fig. 1: System Architecture

Description:

- In this architecture three entities are considered database, Apache Tomcat server and browser.
- First project owner will keep the application on database.
- Then that database will be connected to the Apache Tomcat server.
- And the Tomcat server will be connected to the
- Browser through internet.
- So that user can access the application from internet at anywhere.

Basically system architecture is divided into three modules, they are as follows: -

Module1: Resource module
Input: Resource login
Output: Updated task status and updated skillset.
Module2: EBS module
Input: Resource login or manager login
Output: - Assign tasks by ACO, precedence graph, Resource vector, Task allocation matrix.
Module3: Lock module
Input: Manager login
Output: Resource addition or removal, Task details.

III. PROPOSED SYSTEM

Traditional project management techniques need to be further extended to build more suitable models and tools. To model software project planning as search based optimization is one noteworthy approach. Met heuristic algorithms have been applied success-fully and various software engineering activities like cost estimation, module clustering, design, testing, and software release planning have been modeled as search-based problems.

Thus we build an innovative approach with an Event Based Scheduler (EBS) and an Ant Colony Optimization (ACO) algorithm to develop a flexible and decent model for software project planning. The proposed method is characterized by the following two important features.

The first is novel event-based scheduler is developed. This approach serves a plan by task list and a plan employee allocation matrix. The priorities of task to consume resources are defined by task list and the task workload assignments are specified by planned employee allocation matrix. In this manner the issue of task scheduling and employee allocation both can be considered. In the EBS, initiation time of project, the time when resources are released from finished task and the time when employee join or leaves the project are view as event. The key concept of EBS is to adjust the allocation of employee at events and retain the allocation untouched at non-events.

Second is the Ant Colony optimization (ACO) approach. ACO was proposed by Dorigo, Dorigoand Gambardella in the early 1990s. ACO develops solutions in a step-by-step manner and enables the use of problem-based heuristics to guide the search direction of ants. Therefore, to guide the ants to schedule the critical tasks earlier, and to assign the project tasks to suitable employees with required skills, it is possible to design useful heuristics.

IV. MATHEMATICAL MODEL

Our system can be represented as a set
$X = \{I,O,S_C,F_C,C\}$
Where,
I=set of inputs
O=set of outputs
S_C= set of outputs in success cases
F_C = set of outputs in failure cases
C = set of constraints
$I = \{E, T\}$
where,
E = set of employees
T = set of tasks
$O = \{W, L\}$
where,
W = set of working hours of employees allocated to a task
L = set of lists of tasks rearranged for optimization
$S_C= \{W_n, L_n\}$

where,
W_n = optimal/updated set of working hours of employees allocated to a task
L_n = optimal/updated set of lists of tasks
$F_C = \{W_o, L_o, NULL\}$

where,
W_o = non-optimal/not updated set of working hours of employees allocated to a task
L_o = non-optimal/not updated set of lists of tasks
NULL represents no output
$C = \{C_1, C_2, C_3\}$

Where,
C_1 = “Every task must be completed”
C_2 = “Every task must be allocated to at least one employee”
C_3 = “The project must be completed before the given deadline”
$E = \{E_1, E_2,\ldots,E_n\}$

Where,
E_1, E_2, …, En are employees
$T = \{T_1, T_2,\ldots,T_n\}$

Where,
T_1, T_2, …,T_n are tasks
W, W_n, W_o are in the form
$W = \{W_1, W_2,\ldots,W_n\}$

where,
W_1, W_2, …,Wnare working hours of each employee for each task

L, L_n, L_o are in the form
$L = \{L_1, L_2, \ldots, L_n\}$

where,
$L_1, L_2, \ldots, L_n$ are lists of tasks for each project

V. TECHNICAL DETAILS

Overview of Ant Colony Optimization (ACO): - To deal with the software project planning problem, we develop an ACO approach. The basic idea of ACO is to resemble the foraging nature of ants. When ants search for food, they usually deposit a special chemical on the path they travel through. This kind of chemical, which is called pheromone, serves as a intermediate 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 this herd intelligence phenomenon, ACO was developed by Dorigo and has been successfully applied to various optimization problems.

1.Solution construction—During each iteration of the algorithm, a group of ants set out to frame solutions to the problem. Each ant builds a solution in a constructive manner by selecting components step by step to form a complete solution. The selections are made according to pheromone and heuristic information. In ACO, pheromone is a record of the past search experience of ants for guiding the following ants to make decisions. The components belonging to the best solutions found by the previous ants usually accumulate more pheromone, attracting more ants to select in future iterations. Heuristic is some problem-dependent information that helps ants to have higher probabilities to select promising components in the solution construction procedure.

2. Pheromone management—Along with the solution construction procedure, pheromone values are up-dated according to the performance of the solutions built by ants. Ants tend to deposit more pheromone to the components of better-performed solutions.

3. Daemon actions—Daemon actions mean the centra- lized operations that cannot be done by single ants in the design of ACO algorithms, daemon actions are optional, but many existing ACO variants use different kinds of daemon actions to improve performance. One commonly seen daemon action is the local search procedure.

Event Based scheduler: - 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. The first is novel event-based scheduler is developed. This approach serves a plan by task list and a plan employee allocation matrix. The priorities of task to consume resources are defined by task list and the task workload assignments are specified by planned employee allocation matrix. In this manner the issue of task scheduling and employee allocation both can be considered. In the EBS, initiation time of project, the time when resources are released from finished task and the time when employee join or leaves the project are view as event. The key concept of EBS is to adjust the allocation of employee at events and retain the allocation untouched at non-events.

Software Requirements:

- Operating system: Windows 7, Linux
- Languages: JAVA, JSP, JQuery
- Database: MySQL
- Web server: Apache Tomcat
- Others: Eclipse IDE

Hardware Requirements:

- Processor : Intel i5 core
- Speed : 2.6 GHz
- RAM : 2 GB (min)
- Hard Disk : 250 GB
- Key Board : Standard Windows Keyboard
- Mouse : Two or Three Button Mouse
- Monitor : LCD

REFERENCES

[1] [1] N. Nan and D.E. Harter, “Impact of Budget and Schedule Pressure on Software Development Cycle Time and Effort,” IEEE Trans. Software Eng., vol. 35, no. 5, pp. 624-637, Sept./Oct. 2009.

[2] [2] J. Duggan, H. Byrne, and G.J. Lyons, “A Task Allocation Optimizer for Software Construction,” IEEE Software, vol. 21, no. 3, pp. 76-82, May/June 2004.

[3] [3] R.-G. Ding and X.-H. Jing, “Five Principles of Project Management in Software Companies,” Project Management Technology (in Chinese), vol. 1, 2003.

[4] [4] A. Barreto, M. de O. 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.

[5] [5] E. Alba and J.F. Chicano, “Software Project Management with GAs,” Information Sciences, vol. 177, pp. 2380-2401, 2007.

[6] [6] 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, pp. 1142-1154, 2008.

[7] [7] B. Boehm, Software Engineering Economics. Prentice-Hall, 1981.

[8] [8] B. Boehm et al., Software Cost Estimation with COCOMO II. Prentice-Hall, 2000.

[9] Fig. 8. Analysis of the local refinement steps on the instance j301_1.

[10] Fig. 7. Comparison between the structures of the solutions built by different algorithms.

[11] [9] A. Shtub, J.F. Bard, and S. Globerson, Project Management: Processes, Methodologies, and Economics, second ed. Prentice Hall, 2005.

[12] [10] P. Brucker, A. Drexl, R. Mohring, K. Neumann, E. Pesch, “Resource-Constrained Classification Project Scheduling: Notation, Models and Methods,” European J. Operational Research, vol. 112, pp. 3-41, 1999.

[13] [11] C.K. Chang and M. Christensen, “A Net Practice for Software Project Management,” IEEE Software, vol. 16, no. 6, pp. 80-88, Nov./Dec. 1999.

[14] [12] 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.

[15] [13] A. Kumar V.K. and L.S. Ganesh, "Use of Petri Nets for Resource Allocation in Projects," IEEE Trans. Eng. Management, vol. 45, no. 1, pp. 49-56, Feb. 1998.

[16] [14] L.C. Liu and E. horowitz, "A Formal Model for Software Project Management," IEEE Trans. Software Eng., vol. 15, no. 10, pp. 1280- 1293, Oct. 1989.

★★★