International Journal of Research and Reviews in Applied Sciences And Engineering (IJRRASE)
Vol 8. No.1 –2016 Pp.244-251
©gopalax Journals, Singapore
available at : www.ijcns.com
ISSN: 2231-0061

SOFTWARE PROJECT SCHEDULING BY REPETITIVE STAFFING USING ANT COLONY OPTIMIZATION (ACO)

C. KALAIVANI
Department of Computer Science and Engg.
CHRIST Institute of Technology (CIT)
Puducherry, India.

N. SOWRI RAJA PILLAI
Department of Computer Science and Engineering
CHRIST Institute of Technology (CIT)
Puducherry, India.

ABSTRACT - To expand an effective PC aided techniques for making plans of software project is essential and hard for software engineering. Specific from initiatives in other fields, software projects are humans-extensive and their associated sources are especially human sources. As a consequence, software program task planning has to deal with no longer only the problem of project scheduling however also the problem of Human Resource Allocation. To develop a flexible and powerful version for software program challenge planning, a technique with Task Pre-Knowledge of Employees and an Ant Colony Optimization (ACO) set of rules with event based. The proposed technique represents a plan by using a undertaking listing and a planned worker allocation matrix. On this way, each the problems of project scheduling and employee allocation may be taken into consideration. With the event based, the start time of the task, the time while resources are released from completed responsibilities, and the time when employees be a part of or depart the assignment are seemed as activities. The primary concept of the Pre-Knowledge of employees is to regulate the allocation of employees at events based on the employee's previous experiences or knowledge on the task and hold the allocation unchanged at non-events. With this approach, the proposed method allows the modelling of resource war and undertaking pre-emption and preserves the flexibility in human useful resource allocation. To remedy the planning trouble, an ACO technique is carried out.

INDEX TERMS – Software Project Scheduling, Resource Allocation, Ant Colony Optimization

I. INTRODUCTION

To plan a software project, the project manager desires to estimate the workload , cost and decide the task scheduling and human resource allocation. For workload and cost estimation, a few famous approaches like COCOMO were advanced and widely used. For scheduling and staffing management, in addition to other initiatives (e.g., creation tasks), control is commonly carried out via project control equipment and strategies. As an example, traditional project management strategies like the Program Evaluation and Review Technique (PERT), the Critical Path Method (CPM), and the Resource-Constrained Project Scheduling problem (RCPSP) model were carried out in software project planning. Although those techniques are essential and helpful, they may be an increasing number of taken into consideration to be insufficient for modelling the particular characteristics of today's software initiatives. The primary reason is that, unlikely other tasks, a software projects are a humans-extensive and its associated

assets are specifically human sources. One-of-a-kind software project require employees with exclusive abilities, and talent of personnel drastically impacts the performance of project execution. As such, assigning personnel to the first-rate-equipped tasks is hard for software project managers, and human resource allocation has become an essential element in software program project planning.

Techniques like PERT and CPM lack the consideration of resource allocation and scheduling fashions like the RCPSP do no longer consider the allocation of employees with diverse skills. Consequently, the gear based totally on these conventional project management techniques generally regard challenge scheduling and human resource allocation as separated activities and go away the task of human resource allocation to be carried out by means of undertaking managers manually, resulting in inefficient resource allocation and negative management overall performance. Moreover, as the main resources in software development are humans instead of big machines, resources in software projects can usually be allocated in a more flexible way than those in construction or manufacturing projects. In modern software program tasks, it's miles commonplace that a programmer joins a couple of module improvement tasks simultaneously, and it's also feasible that he stops his modern work and joins the other extra crucial duties. If undertaking pre-emption is nicely designed, human resources may be prepared in a greater efficient way. Some research has pointed out that assignment pre-emption will have a big effect on decreasing the time and value of a software task. But, presently, project pre-emption continues to be in moderation taken into consideration in software task management models and tools. Most existing models have the idea that each employee can only be assigned to a single project at one time. This assumption reduces the flexibility of aid allocation in software program projects making plans.

To build greater appropriate fashions and equipment, conventional assignment control techniques want to be similarly prolonged. One noteworthy method is to version software project planning as a search-based optimization problem. Over the past decade, the idea of formulating software program engineering as search-based problems has attracted growing interest. Various software program engineering activities such as price estimation, module clustering, design, and testing and software program launch making plans had been modeled as search based problems and metaheuristic algorithms were implemented successfully.

II. ACO ALGORITHM

Ant colony optimization (ACO) is one of the techniques for approximate optimization. The inspiring source of ACO algorithms are real ant colonies. More specifically, ACO is inspired by the ants' foraging behaviour. At the core of this behaviour is the indirect communication between the ants by means of chemical pheromone trails, which enables them to find short paths between their nest and food sources. This characteristic of real ant colonies is exploited in ACO algorithms in order to solve, for example, discrete optimization problems.

Marco Dorigo and colleagues brought the primary ACO algorithms in the early 1990's. The improvement of those algorithms became stimulated by the remark of ant colonies. Ants are social bugs. They live in colonies and their behavior is ruled through the aim of colony survival in place of being targeted at the survival of individuals. The behavior that provided the foundation for ACO is the ants' foraging behavior, and especially, how ants can find shortest paths between food sources and their nest. While attempting to find food, ants initially explore the area surrounding their nest in a random way. Even as transferring, ants go away a chemical pheromone trail at the ground. Ants can scent pheromone. Whilst selecting their way, they have a tendency to select, in possibility, paths marked with the aid of

sturdy pheromone concentrations. As quickly as an ant reveals a food supply, it evaluates the amount and the quality of the food and includes a number of it returned to the nest. For the duration of the go back trip, the quantity of pheromone that an ant leaves on the floor may additionally rely on the quantity and quality of the food. The pheromone trails will manual different ants to the food supply. It has been proven in that the oblique communication between the ants thru pheromone trails—called stigmergy—enables them to locate shortest paths between their nest and food sources. That is defined in an idealized placing in Fig. 1.

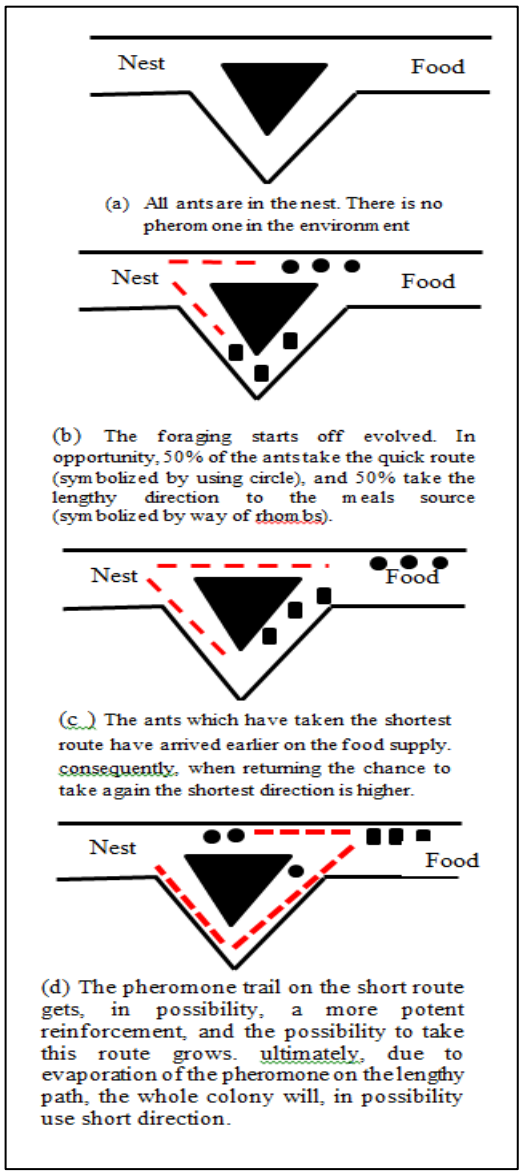

Fig1. Ant Colony

III. REPETATIVE STAFFING

The representation of this scheduler consists of a challenge list and a planned worker allocation matrix. The project list defines the priorities of responsibilities to consume resources, and the deliberate employee allocation matrix specifies the at first deliberate workload assignments. In this way, the illustration takes both the problems of project scheduling and resource allocation into account. The EBS regards the start time of the assignment, the time whilst resources are launched from any completed undertaking, and the time while employees be part of or go away the undertaking as events. To generate a real timetable, the EBS adjusts the workload assignments of employees at activities and useful resource allocation is solved consistent with the priority of the tasks. But in this EBS the task would be assigned to any staff that has no or less knowledge on the assigned tasks, which will affect quality and time line of the task. In the repeated staffing, best fitted person is identified by employee's earlier experience and knowledge of the tasks. In this way, the proposed scheme is practical and flexible as it enables the modelling of task pre-emption and resource conflict. At the same time, compared with the 3D matrix representation, the proposed scheme reduces the size of the search space and thus accelerates the search process. In addition, as the EBS only makes new assignments at events, it is able to keep the implementation of tasks in a more stable manner.

IV. RELATED WORK

In Chang's work [10] 'Genetic algorithm for assignment management', it considers a genetic set of rules for project planning. The task scheduling and allocation of useful resource in tasks is an extremely difficult hassle. Even though we've got an premier solution the changing conditions will have an effect on it. Brute force exhaustive or branch and bound search techniques cannot deal with the complexity inherent in fining satisfactory answers to help project

managers. In existing project management (PM) strategies, business PM tools, and studies prototypes is not efficient in computational skills and best offer passive task tracking and reporting aids. Project managers need to make all most important selections based on their man or woman insights and revel in and need to build the project database to document such decisions and convey reviews in various codecs including Gantt or Pert charts.

In this paper a new approach became evolved based on genetic algorithms (GA) that robotically determines, the usage of a programmable purpose function, a near-most desirable allocation of assets and ensuing schedule that satisfies a given challenge structure and resource pool. It assumes effort is understood, or it calculates from present modeuls such as COCOMO version.

With the result of the set of rules, the software program manager may be able to assign tasks to body of workers in most desirable way and expect the corresponding destiny repute of the assignment which incorporates an intensive analysis on time- and cost versions in the solution space. A new GA is developed which can function on a good deal greater complicated scheduling networks regarding multiple initiatives. They can also address extra practical programmatic and organizational assumptions. The effects of the GA have been evaluated using exhaustive look for 5 check cases. in this paper a 2D array format is used to represent planning. Array includes one for tasks and different for employees. This paper affords a heuristic approach, a genetic set of rules, to clear up the scheduling problem in project management. It represented the workload assignment in percentage.

In Yannibelli's 'knowledge based evolutionary assistant to software program development project scheduling', a knowledge based evolutionary approach is proposed with the aim of assisting to project managers on the early stage of scheduling software projects. Given a software program task to be scheduled, the technique routinely designs feasible schedules for the projects, and evaluates every designed time table in line with an optimization goal that is precedence for managers on the stated level. The objective is to assign the best set of personnel to each project pastime. For this reason, the evaluation of designed schedules in our method is based on the available knowledge information approximately the competence of the personnel concerned in every agenda. This understanding is from historic information about the employees.

The design of an initial scheduler for a software program improvement task is a crucial, non-trivial and high priced task for software companies. This task implies defining feasible start instances (i.e. priority relation should be taken into consideration) and feasible human useful resource allocation (i.e. the resource allocation should be satisfied) for mission sports. in addition, to define the mentioned resource assignments, it's far necessary to estimate the performance of the human resources on the subject of one of a kind challenge sports. This is due to the fact the improvement and the results of an activity depend on the affectively of the sources assigned to it.

The primary benefit is the multi skills of employees. The main disadvantage is that it assumes each employee is assigned to each task at a time. Pre-emption is not allowed.

V. SYSTEM DESCRIPTION

A. Employee Description

Software program development is a humans-intensive activity. To manage employees, an employee database is needed to record the employees' information of wages, abilities, and running constraints. The trouble of human resource allocation is to assign employees to suitable duties so that the tasks can be accomplished correctly. For employees the subsequent attributes are taken into consideration.

- Basic Salary – paid to employees regardless of the workload.

- Over Time Salary – paid for overtime work
- Legal Normal working hours- standard working hours of employee.
- Maxi hours – Maximum possible working hours of employee.
- Pre-Knowledge – employee's knowledge on tasks based on the nature of task or earlier work experience on the repeated tasks.

B. Task Description

In a software program, task can be any activity concerned in software creation, as an instance, class layout, programming, and trying out. A usually used approach for Task Precedence Graph(TPG). A TPG is an acyclic directed graph G (T,A). A task can most effective start whilst all of its direct predecessor duties have completed. The trouble of assignment scheduling is to construct an efficient timetable for the implementation of tasks difficulty to the priority constraints.

- Workload Estimation - the estimated work effort of the tasks in person months.
- Skills – skills requirement for task completion
- Time line – deadline of the tasks
- Max head count – maximum number of employees required for the task completion
- Repeated task – Similarity or Reappearance of task

C. Planning

As the software project planning problem involves task scheduling and employee allocation, a plan for a project just specify when the tasks of the project are processed and how the workloads of employees are assigned to the tasks.

The common constraints of Software planning are as described below:

- The processing order of tasks ought to obey the precedence constraint described by the TPG.
- The working hours of employee should not exceed the maximum hours.
- The number of employees assigned to task should not exceed the maximum head count
- The target objective of project manager is minimizing the cost of effort in total
  - Min(sum of all employee's salary)

D. Scheduler and Staffing

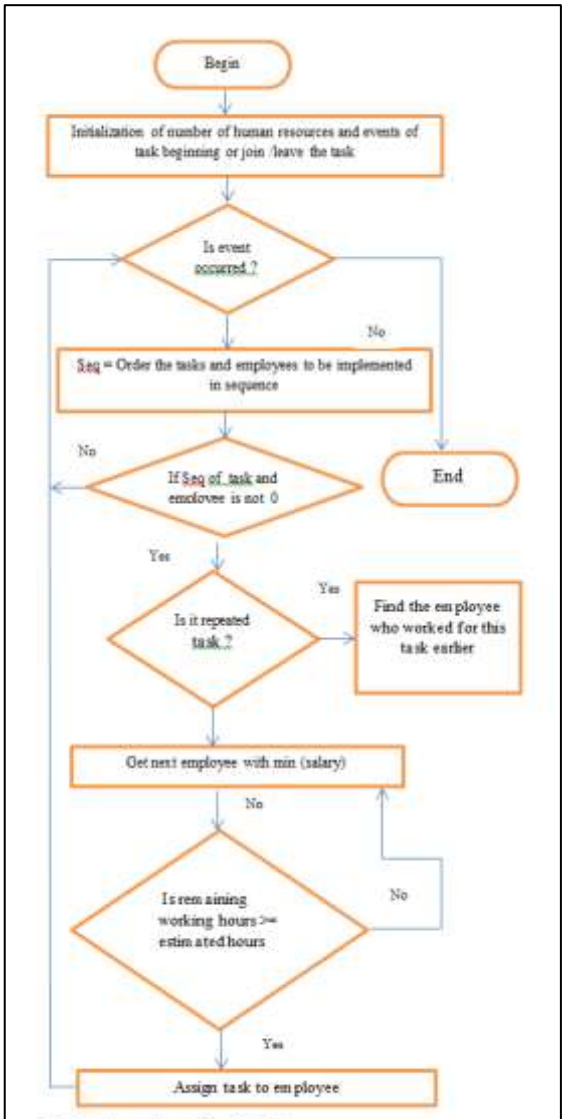

Fig 2. Flow chart of scheduler

Fig 2. is the flow chart for the scheduler and repeated staffing.

Step a: Arrange the employees in ascending order of their skills.

Step b: Choose 1st employee ordered in step a

Step c: Repeatedly assign tasks to employees until working hours fully utilized

Step d: Go to the next employee when working hours become zero.

The existing methods is the granularity of planning. The RCPSP and the employee allocation model make plans with granularity to project. consequently they can handiest make plans with each undertaking having a hard and fast workload challenge and can't address task preemption. Alternatively, the time-line-based totally model makes plans with the granularity to each time period. Though the model is flexible, the plan is simply flexible and therefore the problems of massive search space and desultory workload assignments are incurred. In comparison, the proposed scheme makes plans with the granularity to event. on this manner, the scheme has enough information for modeling task preemption and reduces the search space as well.

E. ACO (Ant Colony Optimization) Approach

The main differences among the behaviour of the actual ants and the behaviour of the artificial ants in our model are as follows:

- While actual ants move in their surroundings in an asynchronous way, the artificial ants are synchronized, i.e., at every generation of the simulated gadget, every of the artificial ants movements from the nest to the meals source and follows the identical direction again.
- Even as real ants go away pheromone at the floor on every occasion they flow, synthetic ants most effective deposit synthetic pheromone on their manner again to the nest.
- The foraging behaviour of actual ants is primarily based on an implicit assessment of an answer (i.e., a route from the nest to the meals source). With the aid of implicit answer evaluation we mean the reality that shorter paths may be finished earlier than longer ones, and consequently they'll get hold of pheromone reinforcement more fast. In contrast, the artificial ants evaluate a solution with admire to some best measure that's used to decide the strength of the pheromone reinforcement that the ants carry out in the course of their return trip to the nest.

To solve the software program project planning problem, this paper proposes an ACO technique. The underlying concept of ACO is to simulate the foraging behavior of ants. When ants look for food, they normally deposit a special chemical at the direction they journey thru. This form of chemical, that's known as pheromone, serves as a medium for ants to talk with every different. With the aid of sensing the awareness of pheromone, different ants can comply with the path to locate the food. Stimulated by way of this swarm intelligence phenomenon, ACO was advanced by means of Dorigo et al. [7], [8] and has been efficiently carried out to numerous optimization issues. An ACO set of rules works via dispatching a group of synthetic ants to construct solutions to the problem iteratively. In widespread, an ACO set of rules can be viewed as the interaction and the repeated execution of the subsequent three fundamental procedures:

Build Solution: For the duration of each iteration of the algorithm, a set of ants got down to build solution to the problem. Each ant builds an answer in an optimistic way by way of deciding on components grade by grade to form a whole solution. The selections are made in keeping with pheromone and heuristic facts. In ACO, pheromone is a history record of the past seeks revel in of ants for guiding the following ants to make decisions. The

additives belonging to the first-rate solutions observed via the previous ants normally accumulate extra pheromone, attracting more ants to choose in future iterations. Heuristic is a few problem-structured data that helps ants to have better possibilities to pick out promising components within the solution construction procedure.

Pheromone Update: Along with the building solution, pheromone values are up- dated in line with the performance of the solution built through ants. Ants have a tendency to deposit more pheromone to the components of better-executed solutions.

Daemon Actions: Daemon movements suggest the centralized operations that cannot be executed by using single ants.

Within the layout of ACO algorithms, daemon actions are optional, however many present ACO variants use distinctive forms of daemon movements to enhance performance [9]. One normally visible daemon motion is the local search system.

As now, diverse ACO variants had been evolved. Two of the fine acting ACO versions include ant colony system (ACS) and max-min ant system (MMAS). ACS variant is followed to develop the ACO approach for software project scheduling. While building the solution the ACS applies a pseudorandom proportional selection rule which aggressively biases selecting the components with the maximum pheromone and heuristic values. On this manner, the ACS strongly exploits the beyond search revel in of ants and has a fast convergence velocity. The global updating rule makes the components corresponding to the best-so-far solution become more attractive. The local updating rule reduces the pheromone on the components just selected by ants to increase the search diversity of the algorithm.

```
01  Procedure ACS
02  Begin
03      Initialize the number of ants and trials
04      While maximum trial is not reached do
05          Initialize each ant in a first node
06          Repeat
07              For each ant do
08                  Choose next node by applying the state transition rule
09                  Apply step by step pheromone update
10              End for
11          Until every ant has built a solution
12          Update best solution
13          Update pheromone
14      End while
15  End
16  End Procedure
```

Fig. 3. Pseudo code for Ant Colony System

The pseudo code of the proposed ACO approach is shown in Fig. 3. At the beginning, parameters of the algorithm are initialized. In each iteration, ants set out to build solution for the problem. Since a plan for the considered problem includes the task list and the planned employee allocation matrix, the solution construction procedure in the proposed algorithm has two steps: 1) building of the task list and 2) building of the employee allocation matrix. Pheromone values are updated by the global and the local updating rules.

VI. CONCLUSION

Software Project Scheduling is one of the most important tasks for Software Project management team. Project scheduling refers to the planning or scheduling of the projects and human resource allocation. In the proposed model of scheduling repeated staffing method is used by analyzing the history of tasks and staff. ACO technique is used to solve the complex planning problem.

REFERENCE

[1] E. Alba and J.F. Chicano, "Management of Software Projects with GAs," Proc. Sixth Mateheuristics Int'l Conf., P. 1152, 2005.

[2] G. Antoniol, M. Di Penta, and M. Harman, "Search-Based Techniques Applied to Optimization of Project Planning for a Massive Maintenance Project," Proc. 21st IEEE Int'l Conf. Software Maintenance, 2005.

[3] G. Antoniol, M. Di Penta, and M. Harman, "Search-Based Techniques for Optimizing Software Project Resource Allocation," Genetic and Evolutionary Computation, vol. 3103, pp. 1425-1436, 2004.

[4] Y. Ge, "Software Project Rescheduling with Genetic Algorithms," Proc. Int'l Conf. Artificial Intelligence and Computational Intelligence, 2009.

[5] L. Ozdamar and G. Ususoy, "A Survey on the Resource- Constrained Project Scheduling Problem," Trans. IIE, vol. 27, pp. 574-586, 1995.

[6] Dorigo M, Blum C. Ant colony optimization theory: A survey. Theoret Comput Sci 2005;344(2–3):243–78.

[7] Dorigo M, Di Caro G, Gambardella LM. Ant algorithms for discrete optimization. Artificial Life 1999;5(2):137–72.

[8] Dorigo M, Gambardella LM. Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Trans Evolutionary Comput 1997;1(1):53–66.

[9] C.K. Chang, M.J. Christensen, and T. Zhang, "Genetic Algorithms for Project Management," Annals of Software Eng., vol. 11, pp.107-139,2001.