Improved Software Project Scheduling by Event Based Optimization

J.Amala[#1] and S.Kavitha[#2]

[1]*M.Phil Scholar, Department of Computer Science, Auxilium College (Autonomous),*
[2,] *Head of the Department, Department of Computer Science, Auxilium College (Autonomous),*
Vellore, Tamilnadu, India.
[1]*amalasankar22@gmail.com*
[2]*kavithasenthil14@yahoo.com*

***Abstract*— recent days Software development companies are facing one of most important problem is project scheduling for different projects. Scheduling is important that play vital role in software project management. When the efficient project planning is reduce the cost and time for the project management. Some software modelling gives resource allocation and human resource allocation even though the project managers facing problems uncertainties in requirements, human resources. Still there is no efficient approach yet proposed to solve uncertainty events project scheduling. This research mainly focused about resource plan and unexpected joining and leaving the staffing. Some systems gives very good solutions for project scheduling like ACO is effective approach that provide project scheduling problems. But it is not satisfactory model to solve uncertainties events project scheduling. This research presents improved ACO based project scheduling algorithm to face uncertainty events in project.**

***Keywords*— Ant Colony optimization, Project Scheduling.**

I. INTRODUCTION

Software project plans play vital role in the end product of software. Software project managers take responsibility for plan the software project. In preparation project scheduling and resource allotment are exciting issues. Scheduling is the process where tasks in the project are ordered to execute by the time. Resource allocation is the process in which available resources are assigned tasks in order to execute the given project. Scheduling and allocations are usually complex depending on the constraints and factors considered in a project. Well-organized project plan decreases the Software creation Cost which will make the companies to be subsequent. NP Complete is one finest method solve project scheduling. Project Scheduling employs by Software project managers frequently to perform preface time and resource estimates, general assistance, and analysis of project alternatives.

Where project head preparing the new software project it's a challenging task, because resources and activities are mostly concentrated. Some different Software modelling methods are gives for scheduling and human resource allocation. But many of the modelling techniques have not considers the variations in joining and leaving event of human resources. In addition, budget and schedule compression has become the norm in today's software industry as a result of its intensified competitiveness due to movements such as globalization, open-source software and outsourcing. Software companies have to strive to get their jobs done faster for less money in order to succeed; automated scheduling tools that can provide insight into how to reduce cost and completion time become increasingly important

.

Many classic models, like Project Evaluation Review Technique (PERT), Critical Path method (CPM) and GERT have presented for Software Project Scheduling Problem (SPSP) by researchers. Project manager using the classical model cannot have accurate and reliable scheduling from final status of software projects in terms of time and cost required to complete projects. In the classical models, scheduling factors are usually obtained from experimental data of various projects and previous projects. The performance of these models is carried based on relationships between projects and changes in relationships of tasks leads to several changes in the amount of time. An incorrect value of cost factors may cause major changes in the final results of SPSP.

In software engineering, the issue of SPSP is very important because the human resources and funding must be optimally managed so that the project is terminated successfully. Last few years, so many papers has been done in the field of SPSP via meta-heuristic algorithms. Today, using the meta-heuristic algorithm has been more familiar in the optimization of the method of solving the complex problems. The GA is used to solve SPSP. Assessment is done from various aspects such as start and end times of each project, the project's overall time and total cost of the project.

The investigational results show that the GA algorithm has good quality recital in estimating the cost and time. One of the most important issues in software project management is choosing the best solution for each project constituent tasks, in the case that the end time and cost of the project have the possible least amount of them. Due to the large number of tasks and the solution of choice for any activity, usually this choice has not a unique answer, but will be formed a set of answers. There are no other solutions over others. Alternatively, in real projects, usually expected costs to perform tasks usually related with uncertainty which lead to substantial changes in the cost of the complete project.

Software project management is "The process of planning, staffing, monitoring, organizing, controlling and leading a software project". Software project managers are responsible for planning and scheduling of software project development. Software project manager leads the development team and is the interface with initiator, suppliers and senior management. The software project manager's job is to ensure that the software project its constraints and

delivers software in time. Software project management is a method of organizing all activities related to a project and its parts.

According to project management organization, it consists of five stages: Proposal writing, project planning, project scheduling, project tracking, personal selection and estimate and project report script .Project Schedule management can be applied to all types of project but it is widely used to control the complex processes of software developments projects. It is an application of knowledge, skills and techniques to execute projects effectively and efficiently. We need software project management because professional software engineering is always subject to organizational budget and schedule constraints. The following are the major activities in software project management.

Software project scheduling is one of the most difficult responsibilities for software managers. It is a heaviness faction that distributes conventional effort transversely the proposed project period by allotment the endeavour to precise engineering tasks. Project preparation involves unscrambling the total work involved in a project into unscrambling activities and moderator the time required to finished these activities. Project Head must also calculate approximately the resources needed to complete each task. Software project scheduling consists of many benefits like: discover harms, save time, build steadiness, improve visibility, attach problem etc. The project scheduling is typically corresponded to as a set of charts viewing the work breakdown makeup, activities dependencies and staff allotment.

An event based ACA was proposed to handle the event of joining and leaving. This approach represented a plan by a task list and a planned employee allocation matrix. Here in method, both the subject of task scheduling and employee allotment can be in use into description. The fundamental idea of the EBS is to adjust the allotment of employees at events and keep the allotment unaffected at non-events. This paper addresses a resolution to the problem of uncertain events which may occur in the software project planning and resource allocation.

II. RELATED WORKS

Carl K. Chang and Hsin-yi Jiang [1] proposed that Effective management of complex software projects depends on the ability to solve complex, subtle optimization problems. Most studies on software project management do not pay enough attention to difficult problems such as employee-to-task assignments, which require optimal schedules and careful use of resources. Commercial tools, such as Microsoft Project, assume that managers as users are capable of assigning tasks to employees to achieve the efficiency of resource utilization, while the project continually evolves. The new model is described along with a new GA that produces optimal or near-optimal schedules.

In this approach based time line model gives two deadlines 1) Quantum of the time-line. 2) Practical upper bound of the time dimension. Authors approach several models like a) Skill model. B) Employee model. C) Task model. Then they are calculating fitness function for cost and time the objectives of these heuristics are to allow the populations to search the space quickly without imposing an excessive processing burden or introducing unrealistic constraints. Since the fitness function is computed using the finer-grained time-line representation, they must be applied during each time step because the properties of both tasks and employees evolve overtime.

A.Charan Kumari, K. Srinivas [2] presents a hyper-heuristic based multi-objective evolutionary algorithm for the solution of scheduling and inspection planning in the software development process, based on the model multi object evolutionary algorithm. This research presents a Multi-objective Hyper-heuristic Evolutionary Algorithm (MHypEA) for the solution of Scheduling and Inspection Planning in Software Development Projects. Scheduling and Inspection planning is a vital problem in software engineering whose main objective is to schedule the persons to various activities in the software development process such as coding, inspection, testing and rework in such a way that the quality of the software product is maximum and at the same time the project make span and cost of the project are minimum.

The MHypEA is an effective metaheuristic search technique for suggesting scheduling and inspection planning. It incorporates twelve low-level heuristics which are based on different methods of selection, crossover and mutation operations of Evolutionary Algorithms. The selection mechanism to select a low level heuristic is based on reinforcement learning with adaptive weights. The efficacy of the algorithm has been studied on randomly generated test problem. This model presents scheduling and inspections are based on project activities like Coding and testing. Each activates are inspected and rework that activity after that forward to next activity. This approach basically have three objective 1) Quality of the product. 2) Project make plan. 3) Cost. After that apply Hyper heuristic algorithm for find the optimal solution of project activity.

Leandro L. Minku, Dirk Sudholt, and Xin Yao [3] authors was presented an improved design for software project based on runtime analysis, which consists of two main points. The first one is to normalize employee's dedications in order to address the problem of overwork. It provides an alternative solution to the repair mechanism using MOFA algorithm that trying to reduce dedication values only where necessary, instead of reducing all dedication values across the board. The second one is to give a clear gradient for searching towards feasibility by introducing a new type of penalty in the evaluation of cost and completion time. Both theoretical and empirical results show that in this design is very effective. Combining the use of normalization to a population gave the best results in these experiments, and normalization was a key

component for the practical effectiveness of the new design. In the normalizing dedications are used employee detections which he or she can attend the several tasks based on their experience and involvement of the project that can be calculated by normalization.

The employee dedication matrix is called genotype. Then this approach was estimate the cost and time using schedule driven algorithm. First we calculate genotype infeasible in that not every task can be completed in finite time. Then the authors are applied population Evolutionary algorithm that need choose encoding, mutation, crossover and fitness function. Here the authors were introduced runtime analysis based on the mutation and crossover results. After that authors are finding optimal solution for time and cost that should minimal for the project.

Francisco Luna, David L. et al [4] are analyse the scalability of four multi-objective algorithms when they are applied to the SPS problem using instances of increasing size. The algorithms are a genetic algorithm (NSGA-II), an evolution strategy (PAES), a differential evolution (DEPT) and a firefly algorithm (MO-FA). The results suggest that PAES is the algorithm with the best scalability behaviour. The paper proposed the Firefly Algorithm (FA) is one of the latest nature inspired optimizers. This algorithm is defined by and it is inspired by the flash pattern and characteristics of fireflies. To solve the SPS problem we have developed the Multi objective Firefly Algorithm (MOFA). In the simplest form, the light intensity I(r) varies according to a fixed absorption coefficient in the media, so the brightness decreases as $e^{-ri2.}$ Therefore, in this algorithm the two most important factors are the variation of the light intensity and the formulation of the attractiveness.

The efficient resolution of the problem is important in the context of large software companies, which have to deal with large software projects. In the experimental evaluation Authors used a benchmark composed of 36 automatically generated instances with increasing size. The results clearly suggest that PAES is the best algorithm in terms of quality of the solutions and scalability. The second best algorithm seems to be MO-FA for small instances and NSGA-II for large instances.

III. PROPOSED METHOD

A. *Event Based Software Project Scheduling*

Since the project scheduling task very critical for project manager some classical method followed some standard based on the complexity of the project even its not good for some uncertainty events and market risks. This approach proposed to give solution that cover of uncertainty events of projects. This system takes all the proceedings of the project into first target or project completion events like Develop the list of project behavior, succession the list of project activities, conclude the relationships between activities , Establish the duration for each activities and Determine the project duration. This each and every event is first milestone and this algorithm also working based on the all events of the project that is work breakdown structure (WBS), Time, Cost, Resource unconfined by completed task and Employee enters and leaves the project.

The suggested strategy is defined by two most important features. First, a description scheme made from activity list along with a planned employee allotment matrix together with a novel event based scheduler takes shape. It allows the modelling of resource conflict and activity preemption. Second, an ACO strategy is suggested since it shows successful application to varied combinatorial maximization issues. ACO builds solutions inside a step-by-step manner that will actually make d ants to plan the critical tasks far back as possible and then to assign the construction project tasks to appropriate employees with required skills. The suggested method efficiently manages employees using an employee database and it also identifies tasks utilizing a Activity Precedence Graph which defines than a activity are only able to start when all of that direct predecessor tasks have finished. Hence the planning objective among the suggested strategy is promising. The suggested system will reduce overall project cost, resources are resourcefully employed in this task as well as a new technique for resolving the software program project planning problem it certainly will decrease the two basic issues in software project management that might be activity scheduling and employee allotment. They provide the clear idea for time scheduling and resource allotment and is going to lessen the manual effort. The suggested system utilize the resources efficiently and allowing the employers to finalize anyone activity among the given time. It furnishes one of the best ways to solve this activity scheduling and employee allotment issues in software project management process. Calculation on the worker knowledge, apportion employee to particular activity reserve are authorize the creation project to become finished on time.

By calculating staff work time that is without doubt /hour income pay for usual efforts and eventually the expenditure of this project can perhaps be reduced. The suggested system helps Project manager in allocating projects to Team leaders and in its place for Team leaders for allotment of activity to team members. It aids allot member of staff to perform in overtimes to deal with their tasks as well as having the optional algorithm is able to yield enhanced plans with stronger workload assignments, inferior costs, decreases the extent of the request space in difference to other existing methods.

B. Proposed Algorithm

Construction of task list

A task list is an order of tasks (tk1, tk2…Tkn) that satisfies the precedence constraints defined by the TPG. Here first define the pheromone and the heuristic for task list construction.

Pheromone:

To build a task list, an ant has to conclude an arrange of the tasks. For the considered software project planning problem, since one task can be assigned to several employees, one employee can undertake several tasks simultaneously, and skill proficiency is considered, it is more difficult to define related tasks for the relation-learning model. Therefore, I adopt the absolute position model with the summation rule in the proposed approach.

Heuristic:

The minimum slack (MINSLK) heuristic is adopted for task list construction. A task with a relatively smaller MINSLK implies that this task is more urgent.

Step -1:

Estimate the shortest possible make span of each task.

Step-2:

Based on the shortest possible make span of each task, the earliest start time and the latest start time of each task can be evaluated, and the MINSLK is calculated by the difference between the latest start time and the earliest start time of the task.

Construction procedure:

To build a feasible task list, each ant maintains an eligible Set of the tasks that satisfy the precedence constraint.

The construction includes the following steps:

Step 1: Put the tasks that can be implemented at the beginning of the project (i.e., the tasks that do not have any precedence tasks) into the eligible Set.

Step 2: For k = 1 to n, process the following sub steps b-1 and b-2, repeatedly:

Step 2(i):

Select a task from eligible Set and put the task to the T_{kn} position of the task list.

Step 2(ii):

Update the eligible Set by removing the selected task from eligible Set and adding new feasible tasks that satisfy the precedence constraint into eligible Set.

After Steps b-1 and b-2 repeat n times, a feasible task list is built.

Employee allocation matrix

Employee allocation matrix describes the structure of estimated working hours of the employees towards their tasks. This matrix addresses the problem of flexibility of human resources and task preemption in the project. Task preemption is achieved by making the regular employees to devote all their normal working hours to the project and by assigning their remaining working hours to some other efficient task. Hence the cost of hiring new employees to the work will be prevented and it also minimize the duration of the project.

Resource allocation

Work load will be assigned to each employee based on their skill set proficiency form description of employee and description of tasks.

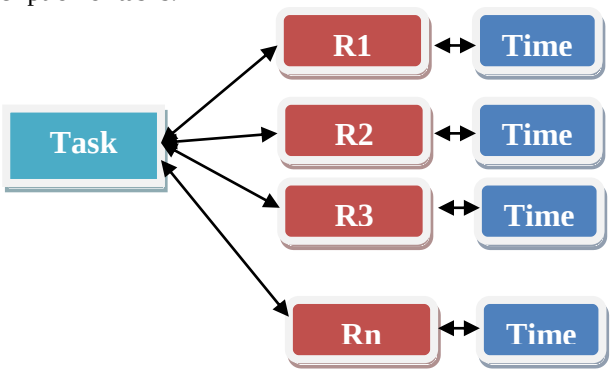

Fig .1 Resource allocation.

Step 1:

T- Get All Tasks.
R- Get All Resources.
t –time

Step 2:

For each task in T
Do
List (L) = Check Available Resources (T, R)
Map each job with available resources as Matrix $^{T}_{Rt.}$
Set time t=0,
For each available Resources R and Available Time t
Do
List Temp 1:= Min { Ti, R, t }—Minimum resources minimum time.
List Temp 2:=Min {Ti, R, t } – Multiple Resources constrained.
Min Duration = min {Temp1, Temp 2} – Optimal Resource Allocation.
Done.

Task representation

Tasks have the following attributes that are taken into consideration for creating tool.

- Expected effort, the expected number of person hours required to complete the task.
- Pessimistic effort, the pessimistic expectation of effort.
- Presumed criticality, whether or not Microsoft Project considers this task critical.
- Required skill sets, a list of skills required to complete the task.

Task duration estimates

The tool determines how long a task is actually going to take based on the estimates turned in to the project manager at the beginning of the project. Each estimate is made up of an optimistic, expected, and pessimistic estimate. The tool determines the actual effort by selecting a value between the optimistic and pessimistic estimates, weighted by the implied

confidence of the estimate. The confidence is inferred from the ratio of the pessimistic to expected estimates. As the confidences of the estimates go up, so does the probability of the actual duration of the task being closer to the expected estimate.

Task reassignment parameters

The learning curve is represented as a penalty associated with resource reassignment. When a resource is reassigned to a task, it is expected that there will be some period during which the resource is not operating at full capacity. I use a simple cumulative Gaussian distribution.

P (T <= a)

Where *P- is total Project, T- Total Task, a is the Resources.*

Task estimation uncertainty

The time granularity of the simulation is assumed to be daily, although any unit of time may be used. For each task, I have an initial estimate of the work required for completion.

EBS ALGORITHM

Step 1: Initialize the number of available human resources.

Step 2: Find the task

Step 3: If the designed working hours are not greater than the residual working hours of the i^{-th} employee, allot designed working hours of the project to the number of working hours of the i^{-th} employee for the task j.

Step 4: Else, the number of working hours of the i^{-th} employee for the task j is set to the remaining working hours of the i^{-th} employee at t.

Step 5: Evaluate the completion situation of the task at time t.

Step 6: If any task is finished at time t, set t+1 as event.

Step 7: Increment t.

V.EXPERIMENTS

In this experimental result, first demonstrate that our algorithm using .NET to reduce the task allocation over head. To comparing ACO and GA based algorithm our method handled well against uncertainty events. In this experimental results shows time reduction when the uncertainty happened in the project events. The Event based ACO algorithms were very responsive to its limitations and limitation settings can have a important point on their performance. Therefore, limitations setting can result in greater elasticity and effectiveness of the proposed model.

Fig .2. Task Allocation.

Fig.3. Monitoring process by Project Manager.

For each task monitoring by the events when the uncertainty happen the task allot to some other staff without workload. Other figure shows how much EBS reduces the time and cost while in the project allocation.

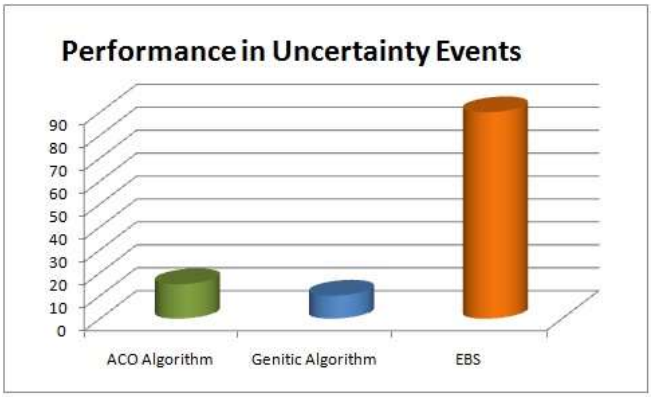

Fig .4. Performance against Uncertainty Events.

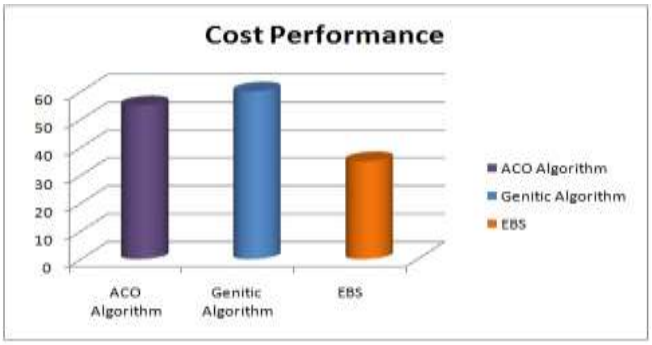

Fig 5. Cost Efficent by EBS.

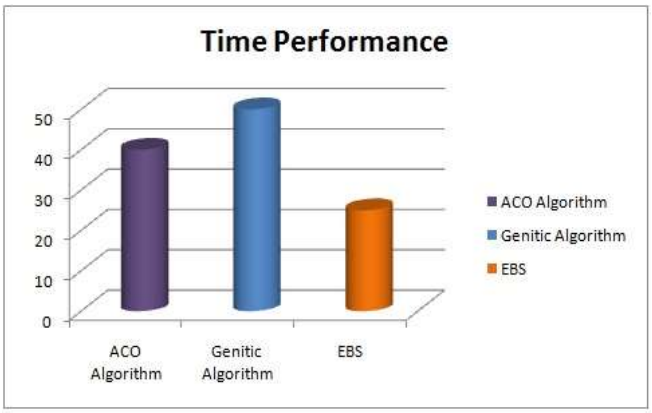

Fig 6. Time Reduction by EBS.

IV. CONCLUSION

Our new method for resolving the software project planning difficulty has been urbanized. The important characteristics of the proposed method are in two aspects. First, initiate an event-based scheduler. Second, the method takes advantage of ACO to solve the complex preparation problem. Improved approaching into the criticality of tasks can help software development managers cope with the uncertainties that they face in project planning. This will better model with sufficiently precise models; its works enhanced job of estimating task criticality than static analysis. However, managers must do project planning in the face of huge uncertainties about how the project will unfold, and better insights into the relative criticality of those many uncertainties can help a manager construct a more robust plan. Ultimately our goal is to develop a decision tool that would help managers with better insights into the criticality of project tasks, as discovered by simulating the way the various uncertainties might unfold and interact as the project progresses. In future project scheduling tasks take account of market risks, shared resources costs and human resources. And also in future research will focused on runtime analysis of the performance of project and mile stones.

REFERENCES

[1]. 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.

[2]. A.Charan Kumari and K. Srinivas," *Scheduling and Inspection planning software evelopment Projects Using Multi- Objective Hyper – Heuristic Evolutionary Algorithm*", International Journal of Software Engineering & Applications (IJSEA), Vol.4, No.3, May 2013.

[3]. Leandro L. Minku, Dirk Sudholt, Xin Yao, (2012) "*Software Evolutionary Algorithms for the Project Scheduling Problem : Runtime Analysis and Improved Design*", Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference, pp 1221-1228.

[4]. E. Alba & J. F. Chicano, (2007) "Software project management with GAs", Information Science, Vol. 177, pp 2380-2401.

[5]. E.Alba, J.F.Chicano,"Software project management with GAs", Information Sciences,Vol.177,No.11,pp.2380- 401,2007.

[6]. Daniel Merkle and Martin "Ant Colony Optimization for Resource Constrained Project Scheduling" ieee transactions on evolutionary computation. Vol. 6, no.4, Aug 2002.

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

[8].Ren, J., Harman, M. and Di Penta, M., 2011. Cooperative co-evolutionary optimization of software project staff assignments and job scheduling. In International Symposium on Search Based Software Engineering. Szeged, Hungary, 10-12 September 2011. Berlin, Germany: Springer-Verlag.

[9]. Wei- Neng Chen and Jun Zhang, "Ant Colony Optimization for Software Project Scheduling and Staffing with an Event-Based Scheduler", IEEE Transactions on Software Engineering. Vol. 39, No. 1, January 2013.

[10]. Lazaros Sarigiannidis, Prodromos D. Chatzoglou, "Software Development Project Risk Management: A New Conceptual Framework", Journal of Software Engineering and Applications, 2011.

[11]. Ali Bou Nassifa, Danny Hob, Luiz Fernando Capretza, "Towards an early software estimation using log-linear regression and a multilayer perception model", The Journal of Systems and Software, Elsevier, Aug 2012.

[12]. Liu Jun, Wang Qiuzhen, Ma Qingguo, "The effects of project uncertainty and risk management on IS development project performance: A vendor perspective", International Journal of Project Management, Elsevier, Nov 2011.

