RESEARCH ARTICLE OPEN ACCESS

Enhanced Project Task Planning With Runtime Scheduler

Ms.B.Sruthi [1], Ms.S.Savitha [2], Ms.R.Sathya [3], Ms.K.R.Sarumathi [4]
Student [1], [2] & [3], Assistant Professor [4]
Department of Computer Science and Engineering
Sri Krishna College of Technology, Coimbatore
Tamil Nadu - India

ABSTRACT

A huge volume of software corporations is growing at a very high rate. The fundamental issue encountered by these software companies are project scheduling, staffing and accessing possessions. Software development organizations often face difficulties in delivering jobs on time, considering the economical and the essential value. Feasible reasons for this issue are poor programming project management, inadequate project scheduling, and insufficient group staffing. To overcome these issues, two algorithms are empowered such as Event based scheduler and particle swarm optimization. The project-based scheduler is differentiated by projecting events such as cost priority and employee experience, which makes the process of scheduling more reliable. This includes regularizing employee's commitment for diversified jobs, and to safeguard that they are not employed over time; a dynamic proxy based project scheduling requires fewer pre-defined parameters and provides a clear gradient towards the feasible solution. The proposed approach is able to deal with the issues of planning and employment in software project management. A job administrator application is developed to perceive the process of source distribution, which is done using dynamic proxy, based project scheduling and if an error occurs during source, distribution it can be resolved with help of task administrator application.

Keywords :- Project Scheduling, Runtime analysis, Resource allocation, Event based scheduler, Software Project Management.

I. INTRODUCTION

Software engineering is the engineering discipline through which the software is developed. Usually the process involves finding out what the client needs, composing this in a list of requirements, designing an architecture that supports all the requirements, designing, coding, testing and combining the separate parts , testing the whole, utilizing and maintaining the software the application of engineering to the design, develop, implement, test and maintain the software in a systematic method. Software engineering deals with research, design, development and testing operating systems software, compiler, and network distribution software for medical, industrial, military, communications, aerospace, business, scientific and general computing applications. It can be divided into sub-disciplines such as software's requirements, design, construction, testing, maintenance, management, development process and quality.

The end product needs to be satisfactory for the client. Hence each of these stages is important since every stage contributes to the end product. The first stage hence is highly important since understanding the requirements from the client plays a major role in the right delivery of the product. The last stage before delivering the product is testing which is very important to test the products in order to check the bugs and faults present in the developed software.

II. LITERATURE REVIEW

Indira Priyadharshini, states that the search based software engineering optimizes the cost of system design by using algorithmic search techniques. Dynamic programming is a problem-solving technique which solves the optimization design cost. Their method provides how cost. Constrained problem can be modeled as the set of two-dimensional knapsack problems and provides a framework and algorithm to optimize design cost of the system. The result shows that the proposed technique reaches the maximum of optimization solution value [4].

Ling Wang, states that a hybrid estimation of distribution algorithm is proposed to solve the resource-constrained project scheduling problem. In the HEDA, the individuals are encoded based on the extended active list and decoded by serial schedule generation scheme, and a novel probability model updating mechanism was put forward for

well sampling the promising searching region. To improve the searching quality, Forward–Backward iteration and a permutation-based local search method were incorporated into the EDA based search to enhance the exploitation ability. The results of the simulation were based on benchmarks and comparisons with some existing algorithms demonstrate the effectiveness of the proposed HEDA [7].

Zamani, presents a genetic algorithm was used for solving the resource-constrained project scheduling problem. The unprecedented component of the algorithm was the use of a magnet-based crossover operator that can pre- serve up to two contiguous parts from the receiver and one contiguous part from the donator genotype. The ability to maintain till three contiguous parts from two parents differs this crossover operator from the powerful and the famous two-point cross-over operator, that can maintain only two contiguous parts from the same parent. Analyzing the performance of the new procedure with that of other procedures indicates its effectiveness and competence [9].

III. MULTI-OBJECTIVE EVOLUTIONARY ALGORITHM

Effective software project scheduling is pivotal when managing the development of medium to large scale projects to meet the deadline and budget. The process of software project scheduling includes identifying project activities, activity dependencies, estimation of resources activities, allocation of people to activities and creating project charts. The project scheduling problem (PSP) deals with the allocation of employees with certain skills so that the required objectives like project cost, duration, etc. can be achieved subject to various constraints [2].

Initially, software project scheduling was assumed to be the system information, such as the effort required for each task and the skills of each employee, are known in advance and remain unchanged[3]. They also assumed that the task execution will never be interrupted by any disruptions. But in the real world, the working environment changes dynamically by unforeseeable events, such as a new urgent task arriving suddenly, an employee leaving, change in client requirements etc.

Multi-objective dynamic project scheduling problems(MODPSP) has been made to capture the dynamic features of real-world software projects that include predictive-reactive scheduling and proactive scheduling[1]. Predictive-reactive scheduling has a scheduling or rescheduling process where previous schedules are adapted to the new environment caused by dynamic events, whereas proactive scheduling attempts to generate a schedule in advance, that has the ability to satisfy performance requirements predictably in an uncertain environment [5].

The project cost, duration, robustness, and stability conflicts with each other. It handles multiple objectives using a multi-objective evolutionary algorithm (MOEA), that provides various trade-offs among different objectives on the Pareto front. The Pareto front can help make informed decisions in dynamic scheduling [6].

The primary aim is to model the software project scheduling problem in a dynamic and uncertain environment by considering multiple objectives and constraints, and propose an MOEA-based proactive-rescheduling method by investigating three aspects such as (i) PSP is articulated as a dynamic scheduling problem with one type of uncertainty and three kinds of dynamic events that often occur in software projects; (ii) MODPSP is constructed, considering the four objectives such as project cost, duration, stability and robustness, and a variety of practical constraints; (iii) a proactive-rescheduling method is formulated to solve MODPSP [10]. The basic idea of the method is to create a robust schedule that considers the project uncertainties, and then revise the previous schedule by MOEA-based rescheduling method in response to critical lively events [8].

IV. DPPS SCHEDULING

A. Individual And Multiple Task Wise Project Scheduling

Initialize the number of employees E_{count} and the number of tasks T_{count} for project scheduling and set employee name, salary per day and the technology known for $\sum_{empid=1}^{n}$

Set the task number, number of days fixed, start date, deadline date, number of hours per day, technology to use and estimated budget for individual task for the $\sum_{Taskid=1}^{n}$

Initialized $E_h \in$ estimated hours as 0, $A_b \in$ allowed budget as 0 ,$P_b \in$ planned budget as 0, $G_a \in$ gained amount as 0. Get Technology known from the assigned task as $Tech_{known}$, allotted hours assigned to the task as AT_{hours} and the Planned Budget assigned to the task as P_{Budget}

$$whereTechnology = \sum_{empid=0}^{n} Tech_{known}$$

$$cnttempasinteger = count$$

$$incntasinteger = \frac{totalhours}{cnttemp}$$

Display Estimated hours, Allowed budget , Planned budget , Gained amount.

B. Individual Or Multiple Employees To Multiple Task Wise Project Scheduling

Initialize the employee id Emp_{id} and the number of tasks T_{count} for project scheduling. Set the employee name, salary per day and the technology known for $\sum_{empid=1}^{n}$
Set the task number, number of days fixed, start date, deadline date, number of hours per day, technology to use and estimated budget for individual task for $\sum_{Taskid=1}^{n}$
Initialized $E_h \in$ estimated hours as 0, $A_b \in$ allowed budget as 0, $P_b \in$ planned budget as 0, $G_a \in$ gained amount as 0.
Get Technology known from the assigned task as $Tech_{known}$, allotted hours assigned to the task as AT_{hours}, and Planned Budget assigned to the task as P_{Budget}

$$whereTechnology = \sum_{empid=0}^{n} Tech_{known}$$

$$cnttempasinteger = count$$

$$incntasinteger = \frac{totalhours}{cnttemp}$$

Display calculated salary, estimated time in minutes, number of tasks debited.

V. PROJECT TASK PLANNING

The idea is to create a practical, effective, formalizable, automatable strategies to integrative testing. The approach initially begins with the user login where he uses DPPS for assigning the tasks and/or assigning the employees.

In the assign employees section, it includes the employee code, employee name, employee address, contact number, designation, experience, and salary per hour.

In the assign tasks section, multiple tasks are inputted and each task contains the task number, number of days fixed, start date, deadline date, the number of hours per day, the technology used and estimated budget.

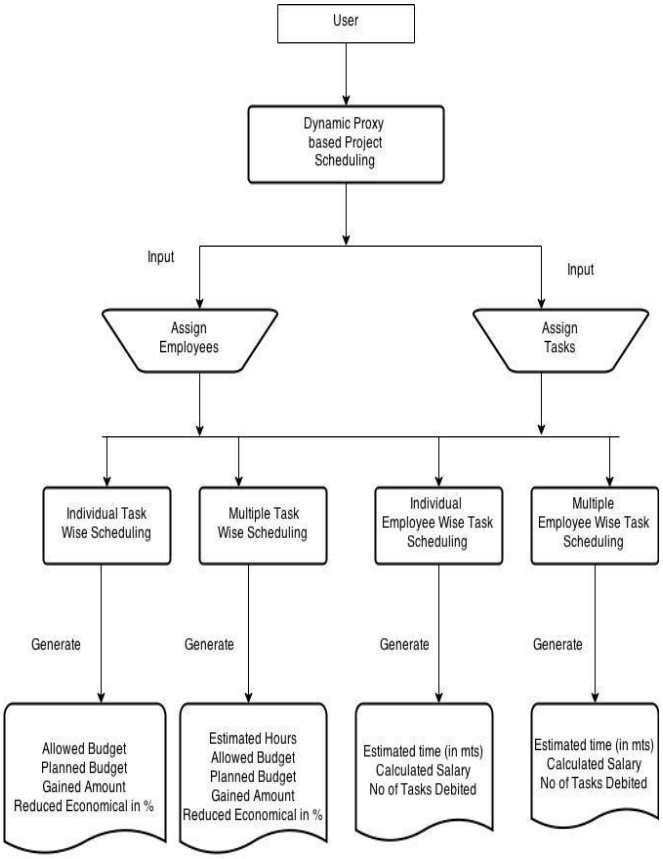

Figure 1:Block Diagram

C. Task Wise Project Schedule

In the task wise project schedule, once the user selects the individual task and calculates the best scheduler using dynamic proxy based project scheduling. The system displays the employee list with their Employee id, employee name, salary per hour details, the number of hours allowed to that employee for the particular task, allowed technology and total salary. Then the gross allowed budget, planned budget, gained amount and reduced economically in percentage is displayed with the estimated hours' details.

D. Entire Task Wise-Project Scheduler

In the entire task wise project scheduler, the current task list is displayed to the user for the task selection which includes the task number, a number of days fixed, start date, deadline date, the number of hours, the technology used and estimated budget. Once the user selected the task from the task list, then the employees' list is displayed for the selected task which includes the employee id, employee name, salary per hour, number of hours allowed and total salary for each employee. The gross results display the estimated hours for the selected task, allowed budget, planned budget, gained amount and reduced economically in percentage.

E. Employee Wise-Project Schedule

In the employee wise project schedule, the user can debit an employee for multiple tasks at a single run time. Once the tasks are selected from the available task list, the employee code is given as input. The Dynamic Proxy based project scheduling connects the selected tasks with the selected employee code and displays the Task Number, task assign date, allotted hours, salary per hour, total salary and assigned technology. The gross result shows the estimated time in minutes, calculated salary for the selected employee and number of tasks debited to the selected employee.

F. Overall Employee Wise Project Schedule

In this module, the user can debit the entire employees for multiple tasks at a single run time. Once the tasks are selected from the available task list and then the employee code is retrieved from the database. The Dynamic Proxy based project scheduling connects the selected tasks with all employee code and displays the Task Number, task assign date, allotted hours, salary per hour, total salary, assigned technology and the allocated employee code. The gross result shows the estimated time in minutes, calculated salary for the selected employee and number of tasks debited to the selected employee. results display the estimated hours for the selected task, allowed budget, planned budget, gained amount and reduced economically in percentage.

VI. EXPERIMENT AND RESULT ANALYSIS

In this system, three algorithms such as Particle Swarm Optimization, Evolutionary Algorithms, Dynamic Proxy Based Project Scheduling (DPPS) are analyzed. The performance of the proposed algorithm (DPPS) is comparatively better than the other two algorithms. This comparison is represented in the form of chart.

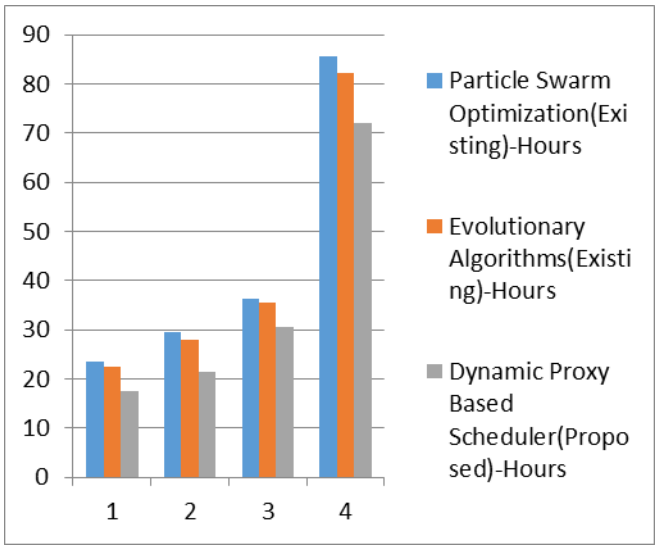

Figure 2: Number Of Hours Utilized

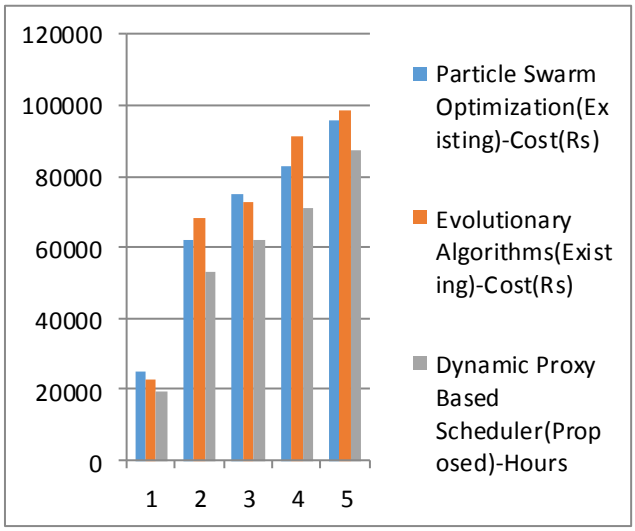

Figure 3: Number Of Cost Spent

VII. CONCLUSION

In our proposed work the customers and the jobs that they lead are controlled using operative planning through the dynamic proxy based project scheduling with added measures such as employee experience and budget priority. This dynamic proxy based project scheduling is employed by the task supervisor which is the most serious part because the extreme quantity of failure in assignment is caused by the improper planning. Also, the distribution of the resources consumes a huge amount of time which can cause rejection of service by a customer, using the proposed system the time administration for the source distribution is done effectively. The task manager application is used to observe the actions of employees and the pool of resources database, capable of refining from errors that may happen regularly. The future scope is to modify certain problems faced the cases with huge and more composite software projects, where the number of jobs, the type of needs and the number of obtainable creators was exposed to inspiration the capability of the algorithm to produce best results.

REFERENCES

[1] Xiaoning Shen, Leandro L. Minku, Rami Bahsoon, Xin Yao, ”Dynamic Software Project Scheduling through a Proactive-Rescheduling Method, IEEE Transactions on Software Engineering, vol. 42, pp. 658-686,July 2016.

[2] Oya Icmeli, S. Selcik Erenguc, Christopher J. Zappe, "Project Scheduling Problems: A Survey", International Journal of Operations & Production Management, vol. 13, pp.80-91.

[3] Claudia-Georgeta (2014), Know-how in Software Project Management", vol. 233 (1), 511-528.

[4] Indira Priyadarsini (2014). Optimizing Design Cost of System by using Dynamic Programming. International Journal of Computational Engineering Research. 04 (2), 2250-3005.

[5] Sabuncuoglu, I., & Goren. S (2003), "A Review of Reactive Scheduling Research: Proactive Scheduling and New Robustness and Stability Measures", Technical Working Paper.

[6] Das, Sanjoy, and Bijaya K. Panigrahi, "Multi-objective evolutionary algorithms", encyclopedia of artificial intelligence,pp-1145-1151, IGI Global.

[7] Ling Wang. (2012). A hybrid estimation of distribution algorithm for solving the resource-constrained project scheduling problem. Expert Systems with Applications. 39 (1), 2451-2460.

[8] Zhang, Qingfu, and Huli Li(2007), "MOEA/D: A Multiobjective evolutionary algorithm based on decomposition", IEEE transactions on evolutionary computation,11(6), 712-731.

[9] Reza Zamani. (2013). A competitive magnet-based genetic algorithm for solving the resource-constrained project scheduling problem . European Journal of Operational Research. 229 (1), 552-559.

[10] Herroelen, Willy, and Roel Leus(2005), "Project Scheduling under uncertainty: Survey and Research potentials." European Journal of Operational Research 165(2), 289-306.