Asian Journal of Information Technology 15 (23): 4851-4860, 2016
ISSN: 1682-3915

FCM Clustering and Experience Based ABC for Software Project Scheduling with EBS

[1]Sarojini Yarramsetti and [2]G. Kousalya
[1]Department of IT, Hindusthan College of Engineering and Technology,
Pin Code: 641 032, Coimbatore, India
[2]Coimbatore Institute of Technology, CSE and IT, Pin Code: 641 014
Coimbatore, India

Abstract: Effective management of complex software projects depends on the knack to solve complex problems. Most inquiries on software project management do not pay enough attention to perplexing problems such as employee-to-task assignments, since it requires optimal schedules, vigilant use of resources and matching the skill set of the employee to the requirements. The existing methods do not considered the user acquaintance while performing the event based scheduling task. To overcome this problem, this research presents, novel scheduling tactic that considers both employee experience and other expertise for scheduling task by proposed paradigm Fuzzy C Means (FCM) clustering method. The projected FCM method, groups the analogous user experience based on the working skills and experience. In Event Based Scheduler (EBS), the events are staring time of the project, the resources released time from the completed tasks and the employees join or , leave time. Project planning problem for each cluster in the EBS is solved by using the Artificial Bee Colony (ABC) algorithm. Experimental results validate that the proposed FCM-EBS with ABC produceprompting values.

Key words: Artificial Bee Colony (ABC), Event Based Scheduler (EBS), Fuzzy C Means (FCM), Resource Constrain Project Scheduling Problem (RCPSP), software project planning, task scheduling

INTRODUCTION

The high complexity of contemporary software projects justifies the research with computer aided tools, to accurately schedule the project development. As software projects become larger, the need to control people and processes and to efficiently allocate resources has turned out to be progressivelysignificant. Managing such projects usually encompasses scheduling, planning and monitoring tasks. This study focuses on minimizing both, the project cost and its make-span, during the assignment of employees to particular tasks in the milieu of a softwareproject. The problem studied here is realized in the literature as the Software Project Scheduling (SPS) problem (Chicano *et al*., 2011).

The Fuzzy C-Means (FCM) algorithm (Chatzis, 2011) isa typical clustering algorithm which is used in a wide variety of engineering and scientific disciplines.The Artificial Bee Colony (ABC) algorithm is a swarm based, meta-heuristic algorithm grounded on the foraging behavior of honey bee colonies. There are three groups of ABC named, scout bees, onlooker bees and employed bees. The scout bee carrying out random search, the employed bee goes to food sources that already visited and the onlooker bee will be hold-up at the dance area. The unemployed bees are the onlooker bees and scout bees (Tereshko and Loengarov, 2005). This study contains an effective approach for the task scheduling and human allocation issue, in the software project planning with an FCM-EBS and Artificial Bee Colony (ABC) algorithm. The proposed method is characterized by the following features. Initially, the representation scheme Fuzzy C Means (FCM) clustering methods for Event Based Scheduler (EBS) will be developed in whichthe similar employee characteristics data are grouped into cluster based on the task completion time and experience time of each one of the employee in the software projects. The representation scheme is composed of a task list based on the employee experience for each task and a planned employee allocation matrix. The task list delineates the priorities of tasks to consume resources and the planned employee allocation matrix states the originally premeditated workload assignments. Later, project scheduling problem is solved using the Artificial Bee Colony (ABC) algorithm. Finally, FCM-EBS with ABC promises prompting results.

Background study: There have been a number of approaches proposed over the years that aim at aiding software project managers resolve on various technical

Corresponding Author: Sarojini Yarramsetti, Department of IT, Hindusthan College of Engineering and Technology, Pin Code: 641 032, Coimbatore, India

factors such as project duration and effort, as well as, developer handiness,with most of the techniques offered tackling scheduling and staffing as an optimization problem. Earlier (Tereshko and Loengarov, 2005) a number of multi-objective Meta heuristics have been used to address this problem. The resulting project scheduling of the algorithms has been scrutinized in order to show their relevant features. Project managers prefer not only good scheduling but also scheduling that can accommodatesmall changes in the parameters of the problem without a large disparity in their cost. Thesechanges in the parameters of the problem can be a variation in the staff or in the task effort, the work doesn't consider the task list, allocation details of the task for each user however the estimates usually change during process execution.

Crawford *et al*. (2015) says that ABC is a meheuristic which is capable of providing solutions generated by complete and also by incomplete techniques. Gueorguiev *et al*. (2009) solve the problem of finding the optimal assignment of work packages to developer teams withthe intention of minimizing project time and maximizing robustness. Multi Objective Genetic Algorithm (MOGA) approach is to formulate this problem as a multi objective Search Based Software Engineering problem, in which robustness and completion time is treated as two competing objectives. Yannibelli and Amandi (2011) suggests a Knowledge-based evolutionary approach with the purpose of supporting to project managers at the early stage of scheduling software projects can be considered. Given a software project to be scheduled, the approach inevitably designs feasible schedules for the project that is priority for managers at the mentioned stage. Our intention is to assign the most effective set of employees to each project activity. In order to assess the performance of our evolutionary approach, present computational experiments is developed over eight different sets of problem instances. This assumption moderates the flexibility of resource allocation in software project planning,but the effectual management of the event based scheduling works not performed in this methods. While solving computationally difficult problems, exact methods often fail, particularly when the probleminstance size increases. Then marginal approaches, such as, heuristics, metaheuristics or hyper-heuristicsare preferred in problem solving. Kyriakidis *et al*. (2012) presents new mixed-integer linear programming models for the deterministic single and multi-mode RCPSP with renewable and non-renewable resources. The modeling approach depend on the Resource-Task Network (RTN) representation, a network representation technique utilized in process scheduling problems, based on continuous time models. First, we recommend new RTN-based network representation methods and then proficiently transform them into mathematical formulations comprising a set of constraints unfolding precedence relations, diverse types of resources and multiple objectives.

Multi-project variant of this problem can still be reduced to a single project.Formulation of RCPSP is to meet some predefined objectives and also the work of Akbari *et al*. (2011), also shows that minimizing the makespan in a single node to get a fixed resource requirements. Hindi *et al*. demonstrated that the evolutionary algorithm isoperative to solve Resource Constraint Project Scheduling Problems (RCPSP) Hindi *et al*. (2002). An empirical study of 2370 instances was conducted. The result disclosed that the algorithm is adept to find the best-known solution in 68% of the instances with an average overall error rate of 0.95%.

Hindi *et al*. (2002) projected a hybrid GA for the RCPSP (Valls *et al*., 2008). Although it is supportive for the scheduling problem, the emphasis on their work is based on the upgradation of the algorithm itself.Alba and Chicano have shown that GAs are quite flexible and accurate for project scheduling and regarded as an imperative tool for automatic project management (Alba and Chicano, 2007). They provided the rudimentary idea of applying GA for automated task assignments. However, in their model, the experiences and skills of employees do not illustrious. Considering and scrutinizing more human resource factors are similar to Plekhanova's research (Alba and Chicano, 2007). In that study, it is asserted that in mathematical theory for scheduling, resource capability factors are not considered as the factorsthat effects the schedule since the resources are made-up to be equal (i.e., they possess equal capabilities). However, in practice, the most effective and competent manner of resource allocation in software projects should be initiated on interested approaches mustinspectchanging skills.

MATERIALS AND METHODS

The study discusses about the problems of human resource allocation and task scheduling of a software project planning model. By recording the employees' information which contains wages, experience, skills and working constraints the software project planning model knows the working nature and behavior of the employees. By knowing the employees ability and providing the constraints where the employee can perform better, the employer can easily accommodate the resources and get a better benefit by providing better circumstances. Using project planning model the employee can also be able to

know about the mistakes where one can compare to other, mistakes can also be rectified easily. After the rectification employee can also check the performance. By providing the monthly or weekly or daily reports the employer can observe the performance of the work and that the employer can easily take necessary steps to improve the productivity or to reach the goal on time. The employer can also view the result on a day by day basis to get better productivity.

By using EBS, multiskill scheduling problem which combines the task list representation and employee allocation, matrix representation are addressed. Tasklist specifies the priorities of the task in the schedule. Multiskill model has various restrictions and it also reduces the relaxation of human resources. Therefore multiskill should be implemented according to the resources where the above conditions will not affect the system or else by combining the multiskill with other algorithms to get better performance. The EBS also adjusts the plan by following the two rules: When resource conflict occurs between two tasks, the task which appear first into the task list will get a higher priority to use the resource. New workload assignments are made when only new workload occurs. New workload occurs when an employee joins or leaves the project when the workload remains the same as the previous time period.

Consider examples that for n number of the employees working in the project for the ithemployee(i = 1,2,…,,) the following attributes are considered.

- b_s-The employee basic salary for time period (e.g., week /month)
- h_{si}-The employee salary for per-hour normal work.
- oh_{si}-The employee salary for overtime work.(e.g., per hour)
- Nh-Normal working hours per period (e.g., per month),
- $maxh_i$- For project, employee possible maximum working hours per month
- [$join_i$;$leave_i$]- Employee join and leave time for the project
- $\{s_i^1, s_i^2 \ldots s_i^{\Phi}$—The skill set for the employee where ϕ is the number of skills and $s_i^j \in_{[0,5]}$is the proficiency score of the j^{th}skill. S_i^j =0 means the employee is not proficient of that skill and s_i^j = 5 means the employee is most proficient of that skill
- $Taskexp_e$, experience level of Employee

Employees may have low proficiency in a skill but if they work on a task necessitating that skill long enough, they will become more proficient as a result of experience. This is often referred to as ‘‘on-the-job training”. It is presumed that each employee has ‘‘Initial Experience” for each skill area. If an employee’s experience level on a task is r at the beginning of a time unit and they work for a fraction of the time (b) on this task, then at the end of the time unit their experience will be $TaskExp_e$ = max ($join_i$ + μ/ϕ xb,5). Again, μ is the employee’s learning speed. The maximum value for experience is the same as that for proficiency, 5.0.

Learning speed threshold value (Lthr) for employee bee, nls normal learning speed value. The $salary_i^t$for the i^{th}employee at the t^{th} month is calculated by:

$$\text{salary}_i^t = \begin{cases} bs_i + \text{hours}_i^t.hs_i + \mu\text{hours}_i^t \le nh, \mu \le Lthr \\ bs_i + nh.hs_i + \left(\text{hours}_i^t - nh\right).ohs_i \\ +nls.hs_i + (\mu - nls).ohs_i \\ nh < \text{hours}_i^t \le \max h_i, nls < \mu \le Lthr \\ \infty\text{hours}_i^t > \max h_i, \mu > Lthr \\ 0\text{therwise} \end{cases} \quad (1)$$

For a task t_j(j = 1,2,…,n), the following traits are considered, pm_j-The appraised work effort of the task in person months , SK_j-The set of skills essential by the task Maxheadj-The maximum employee head count for the task. According to Kyriakidis *et al*. (2012), the maximum, headcount can also be valued based on the COCOMO model.

Deadline and penalty- In practice, it is common to outline deadlines for milestone tasks. If the task is delayed, a penalty will be incurred. The achievement A_j^tyielded by the employees for t at time t can be evaluated by the following steps: The proficiency $prof_{ij}$ of the i^{th}employee for t_j can be evaluated by:

$$\text{prof}_{ij} = \prod_{id \in SK_j} \frac{s_i^{id}}{5} \quad (2)$$

An employee i’s experience level on a task is $join_i$ at the beginning of a time unit and they work for a fraction of the time (b) on this task,

$$\text{askexp}_e = \max\max\left(\text{join}_i + \frac{\mu}{\Phi} \times b, 5\right) \quad (3)$$

The total fitness f_j^t of the employees for t_j on the t^{th} month is given by:

$$f_j^t = \frac{\sum_{i=1}^{m} \text{prof}_{ij}.\text{wh}_{ij}^t}{\sum_{i=1}^{m} \text{wh}_{ij}^t} + \frac{\sum_{i=1}^{m} \text{taskexp}_e \text{weh}_{ij}^t}{\sum_{i=1}^{m} \text{weh}_{ij}^t} \quad (4)$$

Convert f_j^t to a cost driver value V = 8 – round(f_j^t.7 + 0: 5) where the value of V belongs to 1-7. V = 1 means the employees are most suitable for the task and vice versa. The achievement A_j^t for $t_j$on the t^{th}month is calculated by,

$$A_j^t = \frac{\sum_{i=1}^{m} wh_{ij}^t}{V} \quad (5)$$

As the software project planning problem implicates task scheduling and employee allocation, a plan for a project must insist on when the tasks of the project are processed and how the workloads of employees are dispensed to the tasks. More specifically, the plan has to regulate the start time $start_j$ and the finish time $finish_j$ of each taskt_j(jϵ{1,2,..n})and the working hours wh_{ij}^t of all employees iϵ{1,2,.,m} to the task $t_j$during the time window t = {$start_j$, $finish_j$}. The plan must satisfy the following constraints:

The working hours of the i^{th}employee per month must not exceed the limit max h_i, i.e.,

$$\sum_{j=1}^{n} wh_{ij}^t = hours_i^t \le \max h_i, t = 1 \quad (6)$$

The working hours of the i^{th}experienced employee learning speed should not exceed the limit

$$\sum_{j=1}^{n} \mu = weh_{ij}^t \le Lthr, t = 1 \quad (7)$$

The number of employees apportioned to a task t_j is limited by the maximum headcount, i.e.,

$$\sum_{i=1}^{n} sign\left(\sum_{t=start_j}^{finish_j} wh_{ij}^t\right) \le maxhead_j \quad (8)$$

$$\text{where } sign(x) = \begin{cases} 1 \text{ if } x > 0 \\ 0 \text{ if } x = 0 \end{cases}$$

All tasks have to be complete. In other words, for a task t_j, the sum of the achievements for t_j during the time window ($start_j$, $finish_j$) must fulfill

$$\sum_{t=start_j}^{finish_j} A_j^t \ge pm_j \quad (9)$$

This study deliberates cost minimization as the objective function which is given by Eq. 10:

$$\min f = \sum_{t=1}^{end} salary_i^t + \sum_{j=1}^{n} penalty_j + \sum_{j=1}^{n} \mu \quad (10)$$

This offers a representation scheme with a novel event-based scheduler. Similarly to the representation in Yannibelli and Amandi's recent work [7] for the multi skill scheduling problem, representation scheme is given by Eq. 11: Task list : (t_{p1}, .. , t_{pn}). Planned employee allocation matrix :

$$\begin{Bmatrix} pwh_{11} & pwh_{12} & \cdots & pwh_{1n} \\ pwh_{21} & pwh_{22} & \cdots & pwh_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ pwh_{m1} & pwh_{m2} & \cdots & pwh_{mn} \end{Bmatrix}$$

Planned experienced employee allocation matrix :

$$\begin{Bmatrix} weh_{11} & weh_{12} & \cdots & weh_{1n} \\ weh_{21} & weh_{22} & \cdots & weh_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ weh_{m1} & weh_{m2} & \cdots & weh_{mn} \end{Bmatrix} \quad (11)$$

The Event Based Scheduler (EBS) amends a plan in the form of (Eq. 11) into an actual timetable by two rules. First, if there is resource conflict between two tasks, the task that appears earlier in the task list has a higher priority to use the resource. That is, presuming that the ith employee is formerlyslated to concurrently dedicate pwh_{ij} and pwh_{ik} of his working hours to t_j and t_k respectively, if pwh_{ij} + pw h_{ik} > $maxh_i$, $maxh_i$ the employee will first dedicate his working hours to the task with a higher priority ,weh_{ij} + weh_{ik} = Lthr. Second, new workload assignments are only made when events occur. If no employees join or leave the project or no human resource is released by the tasks just finished, the workload assignments remain the same as the previous time period.

In this anticipated system, the event based scheduler framework based on the clustering results is performed.Clustering method parallelto user experience and equal task completion time employees are grouped into same cluster .In order to implement the clustering task in this work use a Fuzzy C Means (FCM) clustering algorithm to assemble the number of the attributes of the employee based on their experience and the task list of the employee. In typical FCM clustering methods distance measure only ganges the difference between two individual employee which attributes to clutch similar user experience. From this point of view it igoners the global view clustering resutls. In order to conquer these

problems, these work devices are regulatory factor based cluster density density to measure the correspondence between two different employee data objects. The recommended distance measure function is dynamically corrected by the regulatory factor until the objective criterion is achieved. Given a employee $e_i = 1,2,\ldots,n$, for every data e_i, the dot density is usually defined as

$$z_i = \frac{1}{\min\min\left(\left\{d_{ij}\right\}\right)} d_{ij} \le e, 1 \le i \le n \tag{12}$$

where d_{ij} denotes the distance between the two employee attributes

$$e_1 = \{b_{s_i}, h_{s_i}, oh_{s_i}, nh, \max h_i, join_i,$$
$$leave_i, s_i^1, s_i^2 \ldots . s_i^{\Phi}, Taskexp_e\}$$

Each one of the employee cluster density is the weighted linear combination of dot densities as expressed in Eq. 13:

$$\hat{z}_i = \frac{\sum_{j=1}^{n} a_{ij} w_{ij} z_{ij}}{\sum_{j=1}^{n} a_{ij} w_{ij}} d_{ij} \le e, 1 \le i \le n \tag{13}$$

Where:

a_{ij} = The category is the label of employee data
e_j and w_{ij} = The weight of e_j
$a_{ij=1}$ = when e_j most likely fits to the cluster i, otherwise $a_{ij} = 0$.
w_{ij} = A positive constant which can be attuned by users

Using the cluster density, z_i the distance measure is revised as Eq. 14:

$$\hat{d}_{ij}^2 = \frac{\left\|e_j - v_i\right\|^2}{\hat{z}_i} 1 \le i \le c, 1 \le j \le n \tag{14}$$

Thus, the optimization expression can be written as follows base on Eq. 15:

$$J_{FCM-CD}(U,V,X) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m \left\|e_j - v_i\right\|^2 \tag{15}$$

$$\frac{\sum_{k=1}^{n} a_{ik} w_{ik}}{\sum_{k=1}^{n} a_{ik} w_{ik} z_k}$$

Applying Lagrange Multiplying Method to Eq. 14, can acquire the two update equations given in Eqs. 16 and 17.

$$v_i = \frac{\sum_{j=1}^{n} u_{ij}^m e_j}{\sum_{j=1}^{n} u_{ij}^m} 1 \le i \le c \tag{16}$$

$$u_{ij} = \frac{\hat{d}_{ij}^{-2/(3-1)}}{\sum_{j=1}^{n} \hat{d}_{kj}^{-2/(m-1)}} \tag{17}$$

In fundamental, essential difference between existing and proposed scheme is the metaphor of planning. The RCPSP and employee assignment models plan with the metaphor of the task. Thus it cannot deal with task preemption but it can make plan with each task having the fixed workload assignment. This model is fine-grained and workload assignment and large search space is desultory. The process of stepwise regression involves these steps are as follows:

Algorithm 1:Fuzzy C Means (FCM) clustering:

Step 1: Elect the number of clustersc, fuzziness index m, iteration error ", maximum iterations T and initialize the membership degree matrix $U^{(0)}$.
Step 2: Get the initial centroids using Eq. (17).
Step 3: Calculate the dot density of every employee data points using Eq. (13). And when the iteration index is t(t = 1,2, .. , T),
Step 4: Update the membership degree matrix $U^{(t)}$ and cluster centroids $V^{(t)}$using Eqs. (16) and (17).
Step 5:Calculate the value of the objective function $J^{(t)}$using Eq. (15).
Step 6:If $|U^{(t)} - U^{(t+1)}| <$ åor t = T, then stop the iteration and get the membership degree matrix and the cluster centroids V, otherwise set and return to step (4).

To decipher the software project planning problem, this study propositions an Artificial Bee Colony (ABC) approach. The core idea of ABC is to simulate the foraging behavior of honey bees. When bee searches for food, they usually deposit a special dancing behavior of bees. ABC has been efficaciously used for project scheduling problem as it is easy to develop and fix many optimization problems with only a few controls of parameters

In ABC optimization, employed bees visit the food source position, based on the task list allocation to employee and the similar employee cluster experience from the FCM clustering results. In ABC each one of the employee bee pleat about number of tasks owed to user and their experience list. The bees are accommodated according the performance based on the project planning

data. Employed bees execute the local investigation to estimate the shortest possible makespan of each task with the best of the experience level to each employee in the clustered group and try to exploit the nearest neighboring locations results of the scheduling food source. The bees waiting in the nest area to unravel the software project planning problem are termed as onlooker bees. The decision of software project planning problem is made based on the employee experience and the heuristic of choosing the ith employee to work for the task t_j given by employing bees. Onlooker bees accomplish the global investigation for solving the software project planning problem and appraise global optimum software project scheduling results. Scout bees discover the new employee task list and planned experienced employee allocation matrix that are not focused by the employed bees. These three steps are sustained until a maximum number of the iterations termination criterion is satisfied. The fitness value for employee task list to elucidate software project scheduling problem is calculated based on the following parameters.

The heuristic of choosing the ith employee to work for the task t_j is denoted as $\gamma_e^{(i,j)} = prof_{ij}/hs_i$. The connotation of this heuristic definition is the rate between the proficiency $prof_{ij}$ of the ithemployee for the task $t_j$and the hour salary of the employee. An employee with a lower salary and a higher proficiency score for $t_j$is more probable to be chosen to work for t_j.If an employee's experience level on a task is $join_i$ at the beginning of a time unit and they work for a fraction of the time (b) on this task:

$$\delta_e(i,j) = \frac{Taskexp_e}{hs_i * Lthr} \tag{18}$$

These parameters are those which belong to single cluster employee attributes. Then construction of the employee allocation matrix is based on the following steps:

Set all values in the employee allocation matrix to 0.

For each task t_j (j = 1; 2; …; m), ascribe the workloads for t_j by the following substeps:

Step b-1: Evaluate the value of $\gamma_e^{(i,j)}$,$\delta_e(i,j)$ for all employees and then check on fitness value for each employee bees by:

$$fit_i = \frac{1}{1+f_i} \tag{19}$$

The fitness of each employee bee is premeditated based on the parameters from (Eq. 18). An artificial onlooker bee software project planning problem based on the probability value p_i, is estimated by the following expression,

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \tag{20}$$

Where fit_iepitomizes the fitness value of the task to each employee i in the location and SN is the size of the population. The nominated task position updates a following Eq. 21:

$$v_{ij} = \chi_{ij} + \theta_{ij}\left(\chi_{ij} - \chi_{kj}\right) \tag{21}$$

Where k and j are randomly selected different task ε {1,2,…,SN} and j ε {1,2,….,D}. Φ_{ij} ε [-1.1], From this result, the parameter value of x_{ij}exceeds its threshold value, the upshot of scheduling problem is acceptable else it is not acceptable as best scheduling results, it is also replaced by the scouts bees, In ABC, if a bee position does not recover the result within a pre - specified number of iterations, then the current task position is assumed to be neglected and it is updates as:

$$\chi_i^j = \chi_{min}^j + rand(0,1)\left(\chi_{max}^j - \chi_{min}^j\right) \tag{22}$$

All the above cited steps majorly depend on following parameters which restricts the operation SN, Maximum Number of the Cycles (MNC). There are various ABC variants developed. Parameters are set in the beginning stage and in the building plans according to the problems by ants and have been processed on further stages. The number of stages for the process is planned by the project planning model according to the model proposed and selected to obtain the best result.

The algorithm 2 shows the optimization of ABC for software project scheduling. First, Initializing and appraising the population. Second, Set cycle value to 1 and repeat the process till the initialization is done. Third, produce the new software project scheduling and evaluating the greedy selection process. Fourth, computing the probability values, producing the software project scheduling solutions, applying the greedy selection process and abandoning the task results in the scout and memorizing the best solution using cycle=cyle+1 it will be done until cycle= MCN.

Algorithm 2: Artificial Bee Colony (ABC) optimization for Software Project Scheduling:

Initialize the population of solutions x_i, i = 1, …..,SN, ,each population as a number of tasks for each employee

Appraise the population

Set cycle = 1

Repeat

Produce new Software Project Scheduling v_i for employed bees (tasks) by using (20) and evaluate

Relate the greedy selection process for the employed bees

Compute the probability values $P_i$for software project scheduling solutions $x_i$by (21)

Produce the Software Project Scheduling solutions v_i for the onlookers from the solutions X_i designated depending on P_i and evaluate them
Apply the greedy selection process for the onlookers
Adopt the abandoned task results in the scout, if exists and replace it with a new randomly produced solution $\vec{v}_i$ by (21)
Memorize the best solution achieved so far
cycle = cycle + 1
until cycle = MCN

RESULTS AND DISCUSSION

The projected procedure on 5 real projects and 10 incidentally generated instances are tested in this experiment. The data from the software construction project for departmental stores is taken as real instances. The four kinds of personnel in a group those are regular chosen staff, regular usual staff, temporary skilful and temporary usual staff. These are the four main types in a project team where one can develop the project with this team itself. The chosen staffs are the one who have more skills and the employer has to pay huge sum for them. Normal staffs are also skilful but they are not well versed in all aspects. These kinds of staffs have stuff in some major areas but not completely in all areas like chosen members. Therefore the usual staffs are the major staffs of the team. Temporary staffs are equal to usual staffs but their service will be required for some specific jobs and they will be sent back once when the work is done. According to the staff type and properties employees are randomly engendered using the haphazard instance generator. Feasibility check is the last process to ensure the skillsets of the employee are matched to the requirements of the project.

The basic information of the test instances is given in Table 1. Here test instances taken for test are Project_1-5. These are assigned with the task number 11, 8, 8, 13 and 14 respectively.Task number is an integer number which is allotted by the computer to complete the given task.Each and every skill is given with an integer value which gives an output that which skill is required and the employee number is an employee id which shows whois eligible to do the skill on time.

Maximum number of solutions is predefined result which produces to show how many time the same work has already achieved. Therefore the employer can easily identify the employee who is skilled to do the work so that the work can be done easily and efficiently and on time.
The best and mean outcomes (averaged over 20 runs) of Various algorithms for whole projects are in Table 2 and 3.

The parameters of the proposed ABC are of number of ants POPSIZE=10. By using different configurations the convergence time of the ABC is differed. According to the used references the parameters for the ABC algorithm are defined and the results for the same are also produced according to the parameters used. Best value is measured.

Table 1: Information of the test projects

Name of the project	Task number	Skill proficiency	Employee Id	Max. number of solutions
Project_1	11	5	10	60000
Project_2	8	5	5	80000
Project_3	8	5	2	30000
Project_4	13	5	10	90000
Project_5	14	5	18	200000

Table 2: Comparison of the results for projects 1-3

	Project_1		Project_2		Project_3	
Instance	Best	Mean	Best	Mean	Best	Best
ABC	658977	624897	1102456	1025841	1658489	1658489
ACO-L	725152	785976	1214580	1158469	1748971	1748971
KGA	1148793	125746	1398413	1245723	1874136	1874136
TS	958796	986478	1471568	1378412	1987491	1987491

Table 3: Comparison of the results for projects 4-5

	Project_4		Project_5	
Instance	Best	Mean	Best	Mean
ABC	1658489	1547890	1467325	1378919
ACO-L	1748971	1647459	1597869	1438968
KGA	1874136	1741648	1769878	1657967
TS	1987491	1847597	1897595	1897654

according to the number of cost taken for test and the mean is calculated by adding all the best cost and dividing it by the test runs. Therefore the ABC algorithm significantly outperforms with other algorithms ACO-L, KGA and TS in all the test instances after the successive of 30runs.

Further it scrutinizes the convergence behavior of the algorithms based on their evolutionary curves. The evolutionary curves which give the best grades found by the algorithms for fitness function verses number of Iterations are illustrated in Fig. 1. The plots indicate that ABC has less Iterations with high fitness than the prevailing methods such as ACO-L, KGA and TS can usually find moral solutions at the very beginning of the search process and preserve its advantage till the end of the search process, since proposed ABC cluster the user experience before scheduling the projects to employee for precise instance Project_1, another instance Project_2. The Instance Project_1 and Project_2are the benchmark instances called as methods. It also shows that the proposed ABC has less cost than the surviving methods. These results validate that the planned EBS is operative.

The above two Fig. 1 and Fig. 2 show the comparison of the evolutionary curves versus the methods Project_1 and Project-2. Both the figures show that the ABC is an efficient model when compared to ACO-L, KGA and TS. The parameter used by the methods and curves are predefined. Figure 3 and 4 show the Cost versus Time comparison of various methods ABC, ACO-L, KGA and TS. Among these, ABC schedules the workload and resource assignment with low cost and earlier time.

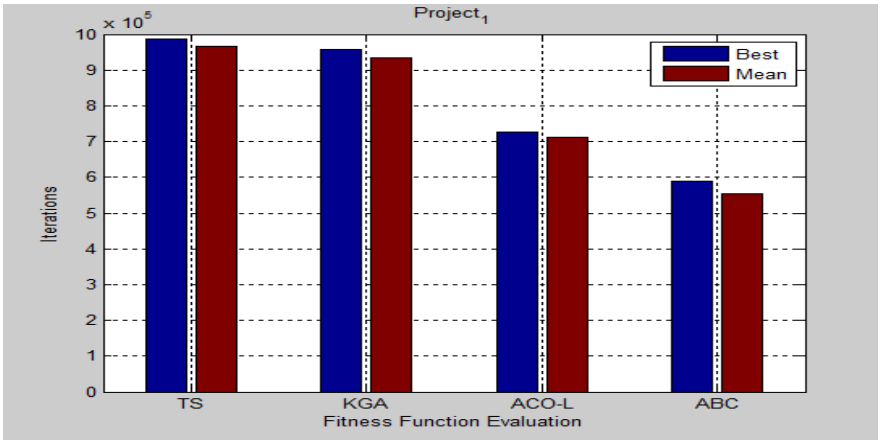

Fig.1: Comparison of the evolutionary curves vs methods (Project_1)

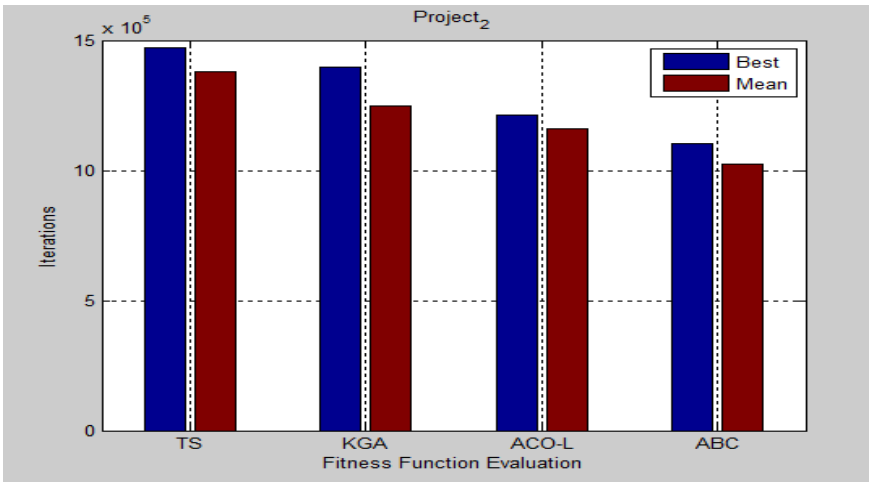

Fig. 2: Comparison of the evolutionary curves vs methods (Project_2)

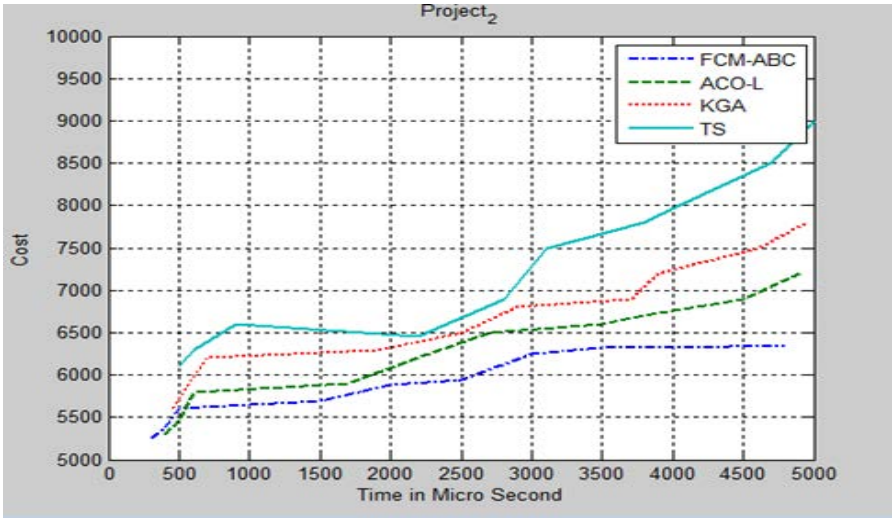

Fig 3: Cost vs time in micro second for various methods (Project_1)

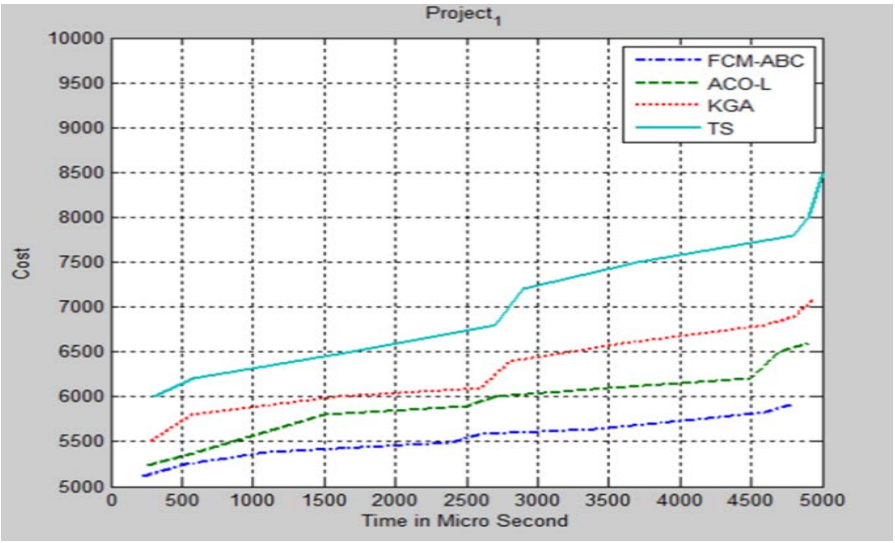

Fig 4: Cost vs. Time in Micro Second for various methods (Project_2)

CONCLUSION

In this study, we present a novel scheme for software project planning issue. Foremost contribution of the proposed methods is following views. Initially, the method introduces a Fuzzy C Means (FCM) clustering methods for event-based scheduler. Later, the method accomplishes the Artificial Bee Colony optimization method to solve the intricate planning problem. Tentative results confirm that the representation, in the anticipated work, the similar features and experienced employee information are convened into same cluster to improve the results of the event based scheduler, then execute the ABC method to solve scheduling problem. The proposed algorithm influences better plans with lower costs in short time and produce unwavering employment appointment collated with distinct subsistent tactics, since it additionally considers the characteristics of the experience. Future work may include better system dynamics integrated with the modeling of progress of tasks where better training and experience models will be encompassed. Moreover, we can explore the impact of team size on these topics.

ACKNOWLEDGEMENTS

I would immensely grateful to G. Kousalya who provided insight expertise greatly assisted the research.

REFERENCES

Akbari, R., V. Zeighami and K. Ziarati, 2011. Artificial bee colony for resource constrained project scheduling problem. Int. J. Ind. Eng. Computations, 2: 45-60.

Alba, E. and J.F. Chicano, 2007. Software project management with GAs. Inform. Sci., 177: 2380-2401.

Chatzis, S.P., 2011. A fuzzy C-means-type algorithm for clustering of data with mixed numeric and categorical attributes employing a probabilistic dissimilarity functional. Expert Syst. Appl., 38: 8684-8689.

Chicano, F., F. Luna, A.J. Nebro and E. Alba, 2011. Using multi-objective metaheuristics to solve the software project scheduling problem. Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, July 12-16, 2011, ACM, Dublin, Ireland, ISBN: 978-1-4503-0557-0, pp: 1915-1922.

Crawford, B., R. Soto, F. Johnson, M. Vargas, S. Misra and F. Paredes, 2015. A Scheduling Problem for Software Project Solved with ABC Metaheuristic. In: Computational Science and Its Applications ICCSA 2015, Osvaldo, G., B. Murgante, M. Sanjay, L.G. Marina and A.A.C.R. Maria et al. (Eds.). Springer International Publishing, Switzerland, ISBN: 978-3-319-21409-2, pp: 628-639.

Gueorguiev, S., M. Harman and G. Antoniol, 2009. Software project planning for robustness and completion time in the presence of uncertainty using multi objective search based software engineering. Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, July 8-12, 2009, ACM, Montreal, Quebec, ISBN: 978-1-60558-325-9, pp: 1673-1680.

Hindi, K.S., H. Yang and K. Fleszar, 2002. An evolutionary algorithm for resource-constrained project scheduling. IEEE Trans. Evol. Comput., 6: 512-518.

Kyriakidis, T.S., G.M. Kopanos and M.C. Georgiadis, 2012. MILP formulations for single-and multi-mode resource-constrained project scheduling problems. Comput. Chem. Eng., 36: 369-385.

Plekhanova, V., 1998. On Project Management Scheduling where Human Resource is a Critical Variable. In: Software Process Technology. Volker, G. (Ed.). Springer Berlin Heidelberg, Berlin, Germany, ISBN: 978-3-540-64956-4, pp: 116-121.

Tereshko, V. and A. Loengarov, 2005. Collective decision making in honey-bee foraging dynamics. Comput. Inf. Syst., 9: 1-7.

Valls, V., F. Ballestin and S. Quintanilla, 2008. A hybrid genetic algorithm for the resource-constrained project scheduling problem. Eur. J. Oper. Res., 185: 495-508.

Yannibelli, V. and A. Amandi, 2011. A knowledge-based evolutionary assistant to software development project scheduling. Expert Syst. Appl., 38: 8403-8413.