VOL. 10, NO. 6, APRIL 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

AN OPTIMIZED EVENT BASED SOFTWARE PROJECT SCHEDULING WITH UNCERTAINTY TREATMENT

Sarojini Yarramsetti[1] and G. Kousalya[2]
[1]Department of IT, Hindusthan College of Engineering and Technology, Coimbatore, India
[2]Department of Computer Science Engineering and IT, Coimbatore Institute of Technology, Coimbatore, India

ABSTRACT

Software organizations every day meet new challenges in the workflow of different projects. Scheduling the software projects is important and challenging for software project managers. Efficient Project plans reduce the cost of software construction. Efficient resource allocation will obtain the desired result. Task scheduling and human resource allocation were done in many software modeling. Even though we are having large number of scheduling and staffing techniques like Ant Colony Optimization, Particle Swarm Optimization (PSO), Genetic Algorithm (GA), PSO-GA, there is a need to address uncertainties in requirements, process execution and in resources. But many of the resource plans was affected by the unexpected joining and leaving event of human resources which may call uncertainty. We develop a prototype tool to support managing uncertainties using simulation and simple models for management decisions about resource reallocation. We also used some real-world data in evaluating our approach. This paper presents, a solution to the problem of uncertain events occurred in the software project planning and resource allocation. This paper presents a solution to the uncertainties in human resource allocation.

Keywords: resource constrained project scheduling problem (RCPSP); software project scheduling problem (SPSP); ant colony algorithm (ACA), genetic algorithm (GA).

INTRODUCTION

Software project plans play vital role in the end product of software. Software project managers take responsibility for plan the software project. In planning project scheduling and resource allocation are interesting 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. Efficient project plan reduces the Software Construction Cost which will make the companies to be succeeding [1]. The task scheduling, problem has been solved for years and is known to be NP complete [2]. Project Scheduling employs by Software project managers frequently to perform preface time and resource estimates, general assistance, and analysis of project alternatives [3]. Planning of software projects using computer software is a challenging task, because activities and resources are mostly human intensive. Various Software modeling techniques are present for scheduling and human resource allocation. But many of the modeling techniques have not considers the variations in joining and leaving event of human resources. 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. In this way, both the issues of task scheduling and employee allocation can be taken into account. The basic idea of the EBS is to adjust the allocation of employees at events and keep the allocation unchanged at non-events. This paper addresses a solution to the problem of uncertain events which may occur in the software project planning and resource allocation.

LITERATURE REVIEW

Assigning the right task to the right person at the right time is a challenging job of project manager. To solve the resource allocation and scheduling problems, traditional models and tools are required for project management and the techniques which are comprehensive. For Software Project Planning, Search Based optimization is the remarkable approach. In last decade, the thought of software engineering planning as search-based problems were increasing attention [4], [5]. P.Brucker *et al*., proposed RCPSP which find an optimal schedule that meets the preceding requirement and minimize the project duration and cost. RCPSP do not consider the human resource allocation with various skills. [6]. Daniel merklel *et al*., [7] proposed an ant colony optimization algorithm for RCPSP which addresses both the problem of human resource allocation and task scheduling. But neither do not consider the task preemption.

Enrique Alba *et al*., [8] proposed a genetic algorithm for managing software projects. The main goal is to reduce the time and cost of a project. But these two goals are in conflict. GA is quite flexible and accurate techniques for this application and can be an important tool for project managers. This is suitable for small projects and for large tasks it is difficult to solve and it was expensive in their solution and project with more employee is easy to address and can be successful in short time. The problem of dealing with a large set of tasks needs to be addressed.

Lai-Hsi Lee [9] proposed an Taguchi's parameter design approach which considers human resource allocation problem using preemption rules. Project preemption allows resources flow passively from one project or task to another. So that flexibility of human resource usage can be increased.

Cedric pessan [10] Proposed branch and bound method to address the problem of project scheduling for maintenance activities in an organization. Preventive maintenance activities are usually planned in advance. So that production is stopped and all maintenance activities should be processed as fast as possible in order to restart the production early. The Branch-and-Bound method uses two lower bounds. The first one is based on a compatibility graph it checks which activities can be processed at the same time. The second lower bound is an adaptation of energetic reasoning that checks if the mandatory parts of the activities during an interval can be processed on the available resources, respecting skill constraints.

A. Barreto [11] proposed an optimization based approach to the problem of staffing a software project. The Software Project Schedules the tasks to the employees based on the abilities possessed and are strongly related to process productivity and end product quality. Thus, one of the most important decisions to be made by a software project manager is how to assign the task to right staff in the project. This staffing problem is regarded as a constraint satisfaction problem. Hence the decision support tool was developed which takes the information of professionals, project activities and characteristics and utility function of the project as an input and then performs an optimization algorithm to give a suggested team for the work as a output. This output will give possible teams that satisfy the given constraints and also suggest the best optimal team to perform the work projects. This model should refine their dynamic aspects like developer motivation, learning curve and error propagation further.

Chang [12] proposed a time-line-based model to address the problem of task scheduling and human resource allocation in a more flexible way. He considered task scheduling and employee allocation together. Different from the previous approaches, the time-line-based model introduces the timeline axis to solve the scheduling and resource allocation problem and uses 3D employee allocation matrix. This model solves both the issues of task preemption and employee allocation and it overcome the disadvantages of RCPSP model. Because of using 3D matrix instead of 2D, the search space is increased in large amount.

Ramasuamy [13] applied queuing theory to software maintenance projects. Podnar and Mikac [14] performed simulations of software maintenance for the function of estimating different process strategies rather than staffing the system.

Bertolino [15] proposed Performance engineering technique based on the usage of queuing models and UML performance profiles, which assist project managers for decision making in teams and tasks. The search based techniques developed by Bertolino *et al*. suggests to managers the configuration capabilities to minimize the completion time and exploit the efficiency in resource usage.

The selection of scheduling and staffing problem is a nonlinear problem presented by Gutjahr *et al*., [16]. He proposed Ant's Colony Optimization (ACO), and GAs combined with a problem-specific greedy technique to solve the scheduling and resource allocation problem. The objective function aggregates two terms: Project gain and global.

Many researchers have been paying attention to optimization techniques as computational intelligence. The techniques include evolutionary algorithms [17, 18], fuzzy logic [19] and constraint satisfaction [20] which reduces the problem search spaces.

Barreto *et al*., [21] proposed Constraint Logic Programming (CLP) for scheduling and staffing problem, which was not based on the idea of search based optimization. CLP focuses on conveying maintenance requests to the majority of qualified team or to the team consisting of highest productivity. To this aim, they assumed the survival of a relationship between completion times and develop skills, which were not empirically sustained by data from real projects. As an alternative, the authors showed a tool for a manager who schedules the project improved and faster.

Wei- Neng Chen [22] proposed an event-based scheduler (EBS) and an ant colony optimization (ACO) algorithm to solve software project planning. The presented approach had shown a plan by seeing the task list and the well planned employee allocation matrix. By this way, both problems of task scheduling and employee allocation were to be considered. In the EBS, the commencement time of the project, the time when resources are released from completing tasks and the time when employees join or leave the project are observed as events. The fundamental idea of the EBS is to adjust the allocation of employees at actions and maintain the allocation unchanged at non actions. The presented approach facilitates the modelling of resource variance and task preemption and conserves the flexibility in human resource allocation. ACO is considered for planning issues.

PROBLEM STATEMENT

Software project planning has to deal with both the problems of project task scheduling and human resource allocation. Already proposed Models for software project planning suffers from very large search space and restrict the flexibility of human resource allocation. Traditional models consider task scheduling and human resource allocation as two separated activities and leave the job of human resource allocation to be done by project managers manually. Search based approaches makes the search space very large and suffers from the problem of desultory assignment of workloads. Event-based scheduler (EBS) and ACO approaches doesn't handle uncertainty even occurred in the project planning.

Classical management methods cannot prevent these kinds of unexpected problems. No matter how much effort is put in, no matter how many stones we turn over, the potential for uncertainty remains. It leads to a third guiding principle for managing uncertainty. The existence of uncertainty is not due to an inherent failure to execute project management processes thoroughly.

PROPOSED SYSTEM

Aco

Construction of task list

A task list is an order of tasks (t_{p1}, t_{p2} , . . . , t_{pn}) that satisfies the precedence constraints defined by the TPG. Here we first define the pheromone and the heuristic for task list construction.

Pheromone: To build a task list, an ant has to determine an order 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, we 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-a: Estimate the shortest possible makespan of each task.

Step-b: Based on the shortest possible makespan 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 a: 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 b: For k = 1 to n, process the following sub steps b-1 and b-2, repeatedly:

Step b-1: Select a task from eligible Set and put the task to the k^{th} position of the task list.

Step b-2: 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.

We have developed a new simulation framework that builds on the experience gained in our initial implementation. This section provides a little more detail on that implementation.

MODEL DESCRIPTION

Description of employee

Software development is a people-intensive activity. To manage employees, an employee database is needed to record the employees' information on wages, skills, and working constraints. The employees can be assigned to suitable tasks so that the tasks can be done efficiently. Suppose m employees are involved in the project, for the i^{th} employee (i =1, 2, . . .,m) all the attributes i.e., Eid (Employee identification), Ename (Employee name), BS (Basic Salary), HS (salary per hour), OHS (Over time salary per hour) NH (Normal working Hours), MAXH (Maximum working Hours) Join Time (The employee Join Time, Leave Time(Leave time of employee) Skill Set (The proficiency of employee) has to considered. Figure-1 shows the Description of Employee.

Eid	Ename	BS	HS	OHS	NH	MAXH	JOINTIME	LEAVETIME	SKILL SET
1	Emp1	35000	97	194	97	340	9	5	Presentation
2	Emp2	25000	70	140	70	355	9	6	Java Documentation
3	Emp3	15000	47	94	47	360	9	6	Oops
4	Emp4	5000	14	28	14	340	9	9	Network
5	Emp5	10000	28	56	20	360	9	9	DotNet
6	Emp6	7000	20	40	20	370	9	8	NS2
7	Emp7	9000	25	50	25	365	9	7	Matlab
8	Emp8	13000	38	72	36	345	9	7	PHP
9	Emp9	17000	47	94	47	265	9	6	Analyst
10	Emp10	13000	38	72	36	245	9	7	Documentation
11	Emp11	11000	31	62	31	236	9	8	Java
12	Emp12	8500	24	48	24	345	9	8	C++
13	Emp13	9500	27	54	27	355	9	8	C

Figure-1. Shows the description of employee.

ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

Description of tasks

In a software project, tasks can be any activity involved in software construction, for example, class design, programming, and testing. Tasks can be described using the parameters called Tid (Task Identification), TName (Task name), and Skill Set (Proficiency of Task), Duration (time need to complete the task) Cost (wages paid to task), NEMP (Number of Employees needed to complete), Penalty (Extra wages paid if not complete in Time). Figure-2 shows the Description of Tasks.

Tid	TName	SkillSet	Duration	Co34st	NEMP	Penalty
1	Network Connection	Network	30	4200	10	2
2	Programming	Oops	40	5200	15	3
3	Programming	DotNet	20	3500	16	4
4	Web Services	DotNet	25	4000	18	5
5	Image Project	MatLab	36	4200	12	3
6	Data Mining	Java Documentation	78	7500	20	6
7	Research Paper	Analyst	45	6000	16	5
8	Research Paper	Analyst	15	2600	5	2
9	Programming	Java Documentation	25	4000	9	3
10	Data Mining	Java	35	3400	10	4

Figure-2. The description of tasks.

Resource allocation

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

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.

Task list

Task list describes the priority order of tasks in which the resources have to be utilized. Task list is constructed using ant colony optimization approach through pheromone and heuristics values. The problem of resource conflict is addressed through the construction of task list.

Event based scheduler

EBS composed of task list and employee allocation matrix so it address the both the problems of task scheduling and employee allocation. The main goal of EBS is to adjust the allocation of employee whenever events occur and keep it constant when events do not take place. The following condition is determined as events

1) Starting of the project
2) Resource released by completed task
3) Employee enters and leaves the project.

EBS, Schedules the tasks to the available resources to complete the projects.

Figure-3 show scheduling the Tasks to the available employee.

VOL. 10, NO. 6, APRIL 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

Tid	Eid	Tname	Ename	SkillSet	Duration	Cost	MaxH	JoinTime	LeaveTime
1	21	Network Management	Emp21	Network	30	4200	345	9	6
2	3	Programming	Emp3	Oops	40	5200	360	9	6
3	20	Programming	Emp20	DotNet	20	3500	360	9	7
4	20	Web Services	Emp20	DotNet	25	4000	360	9	7
5	7	Image Project	Emp7	MatLab	36	4200	365	9	6
6	18	Data Mining	Emp18	Java Documentation	78	7500	325	9	5
7	19	Research Paper	Emp19	Analysist	45	6000	340	9	5
8	19	Research Paper	Emp19	Analysis	15	2600	340	9	5
9	18	Programming	Emp18	Java Documentation	25	4000	325	9	5
10	14	Data Minig	Emp18	Java	34	3400	325	9	5

Figure-3. Show scheduling the tasks to the available employee.

Construction of employee allocation matrix using ACO and event based scheduler:
The employee allocation matrix specifies the originally planned working hours of employees to tasks. Figure-4 shows the Allocation of resources based on the task list.

Tid	Eid	Tname	Ename	SkillSet	Duration	Cost	MaxH	Join Time	Leave Time
1	21	Network Management	Emp21	Network	30	4200	345	9	6
2	3	Programming	Emp3	Oops	40	5200	360	9	6
3	20	Programming	Emp20	DotNet	20	3500	360	9	7
4	20	Web Services	Emp20	DotNet	25	4000	360	9	7
5	7	Image Project	Emp7	MatLab	36	4200	365	9	6
6	18	Data Mining	Emp18	Java Documentation	78	7500	325	9	5
7	9	Research Paper	Emp9	Analysist	45	6000	265	9	7
8	9	Research Paper	Emp9	Analysis	15	2600	265	9	7
9	18	Programming	Emp18	Java Documentation	25	4000	325	9	5
10	14	Data Mining	Emp11	Java	34	3400	236	9	7

Figure-4. Shows the allocation of resources based on the task list.

Task representation: Tasks have the following attributes that are taken into consideration for simulation:

- 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 simulation 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 simulation 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

VOL. 10, NO. 6, APRIL 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

resource is not operating at full capacity. For now, we use a simple cumulative Gaussian distribution.

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, we have an initial estimate of the work required for completion. Figure-5 shows the handling the resource uncertainty.

Tid	Eid	Tname	Ename	SkillSet	Duration	Cost	MaxH	Join Time	Leave Time
1	4	NetworkManagement	Emp4	Network	3039	5232	380	9	9
2	3	Programming	Emp3	Oops	2165	6775	360	9	6
3	5	Programming	Emp5	DotNet	2862	4880	360	9	9
4	5	Web Services	Emp5	DotNet	4844	5715	360	9	9
5	7	Image Project	Emp7	MatLab	3902	4913	365	9	6
6	2	Data Mining	Emp2	Java Documentation	4532	8789	355	9	6
7	9	Research Paper	Emp9	Analysist	3128	7734	266	9	7
8	9	Research Paper	Emp9	Analysis	2212	3957	265	9	7
9	2	Programming	Emp2	Java Documentation	4997	5690	366	9	6
10	11	Data Mining	Emp11	Java	3526	3937	236	9	7

Figure-5. Shows the handling the resource uncertainty.

PERFORMANCE EVALUATION

In Figure-6 and Figure-7 we compare the results of both existing and proposed algorithm with parameters Costs and Time. Time taken to schedule and resource allocation in the project is very low in the proposed system compared to EBS and ACO, which is shown in Figure-6. Cost Performance taken to schedule and resource allocation in the project is very low in the proposed system compared to EBS and ACO, which is shown in Figure-7.

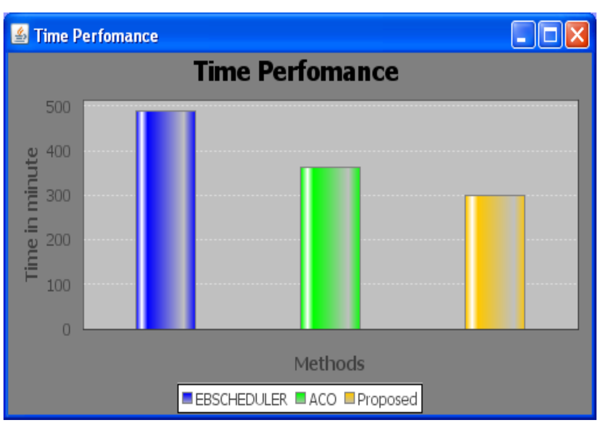

Figure-6. Comparison of results of both existing and proposed algorithm with parameter Time.

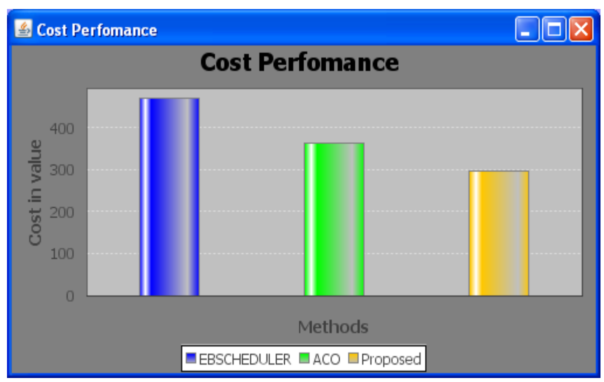

Figure-7. Comparison of results of both existing and proposed algorithm with parameter cost.

CONCLUSIONS

A new method for solving the software project planning problem has been developed. The main characteristics of the proposed method are in two aspects. First, the method introduces an event-based scheduler. Second, the method takes advantage of ACO to solve the complicated planning problem. Better insights into the criticality of tasks can help software development managers cope with the uncertainties that they face in project planning. We believe that simulation, with sufficiently accurate models, can do much better 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. We recognize that some uncertainties are more benign than others, and we hope to provide managers with a tool that would help them to focus on the uncertainties that matter most.

REFERENCES

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

[2] C.W. Chiang, Y.C. Lee, C.N. Lee and T.Y. Chou "Ant Colony optimization for task matching and scheduling", IEEE Proc. Comput. Digit. Tech. Vol. 153, No. 6, November 2006.

[3] Simos Gerasimou, Constantinos Stylianou and Andreas S. Andreou, "An Investigation of Optimal Project Scheduling and Team Staffing in Software Development using Particle Swarm Optimization", 2012.

[4] J. Clarke *et al*., "Reformulating Software Engineering as a Search Problem," Proc. IEE Software. vol. 150, no. 3, pp. 161-175, 2003.

[5] M. Harman, S.A. Mansouri, and Y. Zhang, "Search Based Software Engineering: A Comprehensive Analysis and Review of Trends Techniques and Applications," Technical Report TR-09- 03, Apr. 2009.

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

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

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

[9] Lai-Hsi Lee Robust Preemptive Resource Assignment for Multiple Software Projects Using Parameter Design"International journal of applied Science and engineering. 2007.

[10] Cedric Pessan `Multiskill Project Scheduling Problem and total Productive Maintenance" MISTA 2007.

[11] A. Barreto, M. de O. Barros, C.M.L. Werner, "Staffing a Software Project: A Constraint Satisfaction and Optimization-Based approach," Computers and Operations Research. vol. 35, pp. 3073-3089, 2008.

[12] 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.

[13] R. Ramaswamy, 'How to staff business critical maintenance projects', IEEE Software. 7(7), 90-95, (May 2000).

[14] Podnar and B. Mikac, 'Software maintenance process analysis using discrete-event simulation', in European Conference on Software Maintenance and Reengineering, pp. 192-195, Lisbon, Portugal, (Mar 2001). IEEE Society Press.

[15] A. Bertolino, E. Marchetti, and R. Mirandola, 'Performance measures for supporting project manager decisions', Software Process: Improvement and Practice. 12(2), 141-164, (2007).

[16] W. Gutjahr, S. Katzensteiner, P. Reiter, C. Stummer, and M. Denk, 'Competence-driven project portfolio selection, scheduling and staff assignment', Central European Journal of Operations Research. 16(3), 281-306, (September 2008).

[17] Alba, E. and Chicano, J.F., 2007. Software project management with GAs. Inform. Sciences. 177(11), pp. 2380-2401.

[18] 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.

[19] 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.

[20] Callegari, D.A. and Bastos, R.M. 2009. A multi-criteria resource selection method for software projects using fuzzy logic. In 11th International Conference on Enterprise Information Systems. Milan, Italy, 6-10 May 2009. Berlin, Germany: Springer-Verlag.

ARPN Journal of Engineering and Applied Sciences

[21] Barreto, A., Barros, M.d.O. and Werner, C.M.L. 2008. Staffing a software project: A constraint satisfaction and optimization-based approach. Comput. Oper. Res. 35(10), pp. 3073-3089.

[22] 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.