

# **Multi Objective Genetic Algorithm for the Optimized Resource Usage and the Prioritization of the Constraints in the Software Project Planning**

D.Sundar  
Lecturer

Thiagarajar School of management  
Madurai, India

B.Umadevi  
Asst. Professor

Raja Dorai Singam Govt. College  
Sivagangai, India

Dr.K.Alagarsamy  
Associate Professor

Computer centre M.K.University  
Madurai, India

## **ABSTRACT**

Capability Maturity Model Integration (CMMI) helps organizations to improve software development processes. But it contains very little information on process dynamics. It doesn't address the following issues like Specific tools, methods and technologies to be followed, Issues in Human resource management, People management methodology and cost attached to people manager. In this paper we focus on the real world problem in the systems development life cycle such as Organizations understaffed, Separation of duties. In order to overcome the problem we have suggested a Multi objective genetic algorithm (MOGA) for minimization of the human resources used. So in turn MOGA has been used to minimize the cost associated, minimize the time involved and to maximize the efficiency by proper usage. The Prioritization of the various constraints involved in the project is also done by the MOGA, which shows a good result over the manual allocation. The results compared with the manual assignment and the comparative results are reported and discussed, which shows the effectiveness of the proposed approach for the project planning.

## **Categories and Subject Descriptors**

D.3.3 [Software Engineering]: Project planning, CMMI, Human resource management

## **General Terms**

Software Engineering, Software Project management

## **Keywords**

CMMI, MOGA, HRM, etc.

## **1. INTRODUCTION**

Software standards help an organization to adopt a uniform approach in designing, developing and maintaining software. There are a number of standards for software quality and software quality assurance. Once an organization decides to establish a software quality assurance process, standards may be followed to establish and operate different software development activities and support activities. A number of organizations have developed standards on quality in general and software quality in specific.

Challenged by the changes during the past 10 years in the importance and utilization of software by businesses, enterprises

now view the cost and quality of software development differently. Whereas IS organizations formerly minimized the significance of development engineering costs as non-recurring, the current critical nature of software in a product or service offering has elevated the importance of the development process. Thus, it is essential to maximize the use of any available tools or models to increase the efficiency and effectiveness of software development.

Software Engineering Institute (SEI) has developed, what is called a 'Capability Maturity Model' (CMMI) now called the Capability Maturity Model Integration (CMMI) to help organizations to improve software development processes. The CMMI contains very little information on process dynamics. . It doesn't address the following issues like Specific tools, methods and technologies to be followed, Issues in Human resource management, People management methodology and cost attached to people manager.

To solve these type of problems, scheduling problem with worker allocation was presented in [1] where workers skills level to each job is all the same. However in fact each worker has a different skill level for each machine. So another approach was proposed in [2], a new model consisting of the following three new procedures: shortening of ideal time, modifying infeasible solution to feasible solution and a new selection method of GA. In recent years, modified scheduling problem with worker allocation is presented by [3], in this proposal, a concept of Module Type GA (MTGA) was proposed, and a design of MTGA is proposed for solving modified scheduling problems with the worker allocation.

This paper introduces a Multi Objective Genetic Algorithm based Resource usage and the Prioritization of the Constraints in the Software project planning. This is used for the minimization of the resources for the usage in the project development and to fix the precedence of the constraints.

This paper is organized as follows. In Section 2 the background study has been done for such kind of problems. Section 3 gives the definition of the problem. Section 4 describes the genetic algorithm proposed and discusses the representation of the individuals. The experimental study and results are presented in Section 5. Finally, conclusions and future work are outlined in Section 6.

## 2. BACKGROUND STUDY

The Human resource allocation for the project planning can be thought as the resource allocation problem. During the last couple of years many heuristic procedures have been developed for this problem, but still these procedures often fail in finding near-optimal solutions. This problem falls to the class of NP hard optimization problems.

Recent classification and survey can be found in [4] and [5]. The survey provided [5] presents more than eighty models and algorithms for complex scheduling problems and discusses the more recent work is due to [6],[7],[8],[9]. Planning a large scale software project involves a set of activities and an allocation of programmers to teams and teams to work packages. Human resource is the most important asset for a company to be competitive.

The prioritization of the constraints in the human resources and allocation and the minimizing the human resources are the major area we have to concentrate on. As the complexity of projects increases, the requirement of an organized planning and scheduling process is enhanced [10]. The company must be able to handle a number of projects. Most of the projects have been built in condition of multi-project [11].

For developing, there is a need to employ some developers who have skills which are necessary for the project. The different projects not only isolated but also related in many ways, for example, resources competition of projects and conflict of development period. The relations make the management of human resources for the projects complex, especially the different projects of resources competition. The management of multi-projects is very important for management of projects when resources are limited.

The traditional methods of project management like Crux Path Method and Plan Estimate Review Technology can resolve the problem of management of one project effectively [12]. However, the traditional management methods can not effectively handle of the multi-projects which happen at same time.

## 3. THE DEFINITION OF THE PROBLEM

In this paper we have the objective to meet out the prioritization of the constraints in the human resources and allocation and the minimizing the human resources. The problem can be formulated as finding a schedule of the activities, taking into account the resources and the precedence of the constraints, Let  $R_i$  represent the Resources used for the process  $i$ . A schedule can be represented by a vector of resources of the processes as

$(R_1, \dots, R_m, \dots, R_{n+1})$ .

So the conceptual model can be drafted based on the following objective functions

1. Min  $R_{n+1}$
- subject to
2.  $R_k \leq R_i - D_i \quad i = 1, \dots, n+1, k \in P_i$
  3.  $R_i \geq 0 \quad i = 1, \dots, n+1$

The objective function (1) minimizes the resource required for the process  $n+1$ . (2) impose the precedence relations between

process and constraints (3) forces the resource to be non-negative.

Another problem to be concentrated is the fixing of the precedence of the constraints. The main problem in the project planning is how to handle the constrains in an order so that it requires the minimum thought through process and to lead to the satisfaction of the other constraints.

The order of the processing of the various constraints can be viewed as an optimization problem. The constraints can be classified as the active and semi active. The constraint to be taken over to be prioritized first and can be called as the active one and other which can be solved a little more later can be called as semi active , but this semi active constraints are also important but not immediately to be solved.

Such problems are instances of the ‘bin packing problem’, the solution of which is NP-hard [13] and for which, search based techniques are known to be effective.

## 4. THE PROPOSED GENETIC ALGORITHM

Generally speaking, the resource allocation problem is NP hard, which means there are no known algorithms for finding optimal solutions. Exhaustive search methods can be used to solve scheduling problems, but they require forbiddingly long execution times as the problem size increases. In this paper, an optimized method is proposed, using genetic algorithm to solve above allocation problem.

**4.1 Genetic algorithm** Genetic Algorithms are stochastic methods that can be used to solve a very broad class of optimization problems. They are known to solve problems in a heuristic way under consideration of the problem’s environment. Therefore, it is useful to apply Genetic Algorithms to improve and manage allocation problem [14].

GAs is a robust general-purpose search program based on the mechanism of natural selection and natural genetics [15]. Genes and chromosomes are the fundamental elements in GAs. A chromosome is a string of genes. In a real problem, genes are the variables that are considered influential in controlling the process being optimized, and a chromosome is a solution to the problem. Genetic Algorithms search for the optimal solution from populations of chromosomes. The representation chosen for the genome is pivotal to the performance of GA [16].

In many optimization methods, we move gingerly from a single solution in the decision space to the next using some transition rule to determine the next solution. This solution-to-solution method is dangerous because it is a perfect prescription for locating false peaks in multimodal search spaces. By contrast, GAs work from a rich database of solutions simultaneously (a population of chromosomes), climbing many peaks in parallel; thus the probability of finding a false peak is reduced over methods that go solution to solution.

## 4.2 Multi Objective Genetic Algorithms (MOGA)

Multi-objective optimization deals with solving optimization problems which involve multiple objectives. Most real-world

search and optimization problems involve multiple objectives (such as minimizing fabrication cost and maximize product reliability and others) and should be ideally formulated and solved as a multi-objective optimization problem. However, the task of multi-objective optimization is different from that of single-objective optimization in that in multi-objective optimization, there is usually no single solution which is optimum with respect to all objectives [17]. Classical search and optimization methods usually work with a point-by-point principle and thus are required to be applied many times, each time finding one Pareto-optimal solution. Moreover, the efficacy of classical methods largely depends on the shape of the Pareto-optimal region, discreteness of the search space, presence of constraints, and others [18]. Over the past decade, population-based evolutionary algorithms (EAs) (genetic algorithms (GAs) and evolution strategies (ESs)) have been found to be quite useful in solving multi-objective optimization problems, simply because of their ability to find multiple optimal solutions in a single simulation run.

The pseudo code for the MOGA is

```

Initialize Population
Evaluate Objective Values
Assign Rank based on Pareto Dominance
Computer Niche Count
Assign Linearly Scaled Fitness
Assign shared Fitness
For i=1 to number of Generations
Selection Crossover Mutation
Evaluate Objective values
Assign Rank based on Pareto Dominance
Computer Niche Count
Assign Linearly Scaled Fitness
Assign shared Fitness
End Loop

```

### 4.3 Proposed Methodology

In this paper we propose Multi Objective Genetic Algorithm (MOGA) which overcomes the disadvantages of the previous algorithms. The processing steps of MOGA are as follows

- Step 1: Generate the initial population P.
- Step 2: Subdivide the population into ‘m’ subpopulation according to the number of objectives (m)
- Step 3: For each subpopulation ‘S’ do the following steps.
  - Step 3.1: Evaluate the fitness based on the objectives assigned to each subpopulation.
  - Step 3.2: Select the best chromosome ‘X’ from ‘S’.
  - Step 3.3: Select two chromosomes ‘P1’ and ‘P2’ from ‘S’.
  - Step 3.4: Apply crossover between ‘P1’ & ‘P2’. Let O1 be the best offspring. Step 3.5: Apply crossover between O1 and

‘X’. Let O2 be the best offspring. Step 3.6: if O2 is better than ‘X’ then replace O2 with ‘X’.

Step 3.7: Iterate steps (3.1 – 3.5) until all chromosomes are considered.

Step 4: The best chromosome in each subpopulation is compared for best solutions. Step 5: Iterate steps 3 & 4 until best Pareto optimal solutions are obtained.

### 4.1 References and Citations

Footnotes should be Times New Roman 9-point, and justified to the full width of the column.

The references are also in 9 pt., but that section (see Section 7) is ragged right. References should be published materials accessible to the public. Internal technical reports may be cited only if they are easily accessible (i.e. you can give the address to obtain the report within your citation) and may be obtained by any reader. Proprietary information may not be cited. Private communications should be acknowledged, not referenced (e.g., “[Robertson, personal communication]”).

## 5 EXPERIMENTAL STUDY AND RESULTS

The data for the study is collected from a software firm located in Madurai. The pilot survey is made by the interaction with the Project Head and the Human Resource Manager of the firm. The existing allocation of resources is done manually. The constraints is based on the system requirement analysis document

In this paper we have chosen problem a set of 5 jobs. The expected resource spent is given in table 1. The expected time for development based on the given constraints is given in table 2. The number of given constraints were fixed to five.

Table 1: Resources Used for the Project

Resources Used /No. of the Job	Based on Manual Allocation method	Based on the proposed method
1	22	20
2	48	44
3	96	92
4	108	99
5	223	218

Table 2: Expected Time for development

Expected Time(days)/No. of the Job	Based Manual Allocation	Based the proposed method
1	60	56
2	72	68
3	75	72
4	120	112
5	190	186

Fig1: Comparative chart for resources used

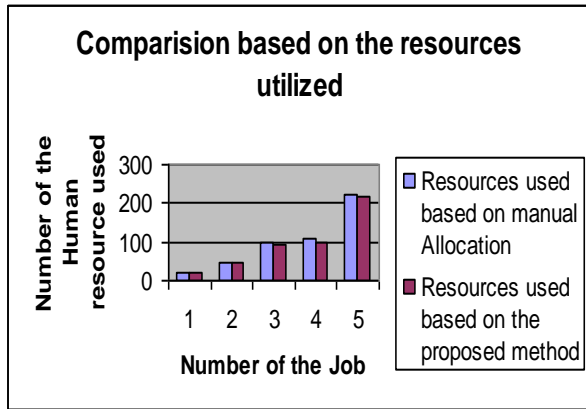
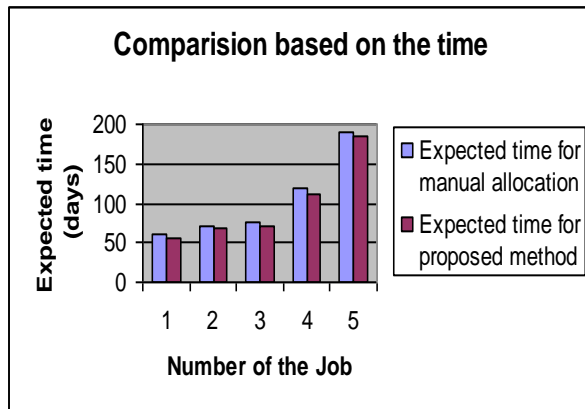


Fig2: Comparative chart for prioritizing the constraints



## 6. CONCLUSIONS

The paper discuss about organizations to improve software development processes based on the Multi Objective Genetic Algorithm for the optimized Resource usage and the Prioritization. The Prioritization of the various constraints involved in the project is also done by the MOGA, which shows a good result over the manual allocation. Based on the pilot survey the results are compared with the manual assignment and the comparative result are reported and discussed, which shows the effectiveness of the proposed approach for the project planning. In the future we shall enhance the research based on the optimistic usage of the resources as well as the reduction of the impact of the change management.

## 7. REFERENCES

[1] H.Iima ,N.Sannomiya, “Proposition of Module Type Genetic Algorithm and Its Application to Modified Scheduling Problems with Worker Allocation” IEEE Japan ,Vol.122-C, pp.409-416,2002

[2] A.Osawa , K. Ida ,“Scheduling Problem with Worker Allocation using Genetic Algorithm” , Japan-Australia workshop on intelligent and evolutionary systems, pp.1- 8 ,2005

[3] H. Iima ,N.Sannomiya, “Module Type Genetic Algorithm for Modified Scheduling Problems with Worker Allocation” Proceedings of the American Control Conference Arlington, VA, 2001.

[4] P. Brucker, A Drexl, R. Mohring, K. Neumann, E.Pesch, Resource-constrained project scheduling:Notation, classification, models and methods, *European Journal of Operational Research*, Vol.112 (1), 1999, pp. 3-41.

[5] R. Kolisch and S. Hartmann, Experimental investigation of heuristics for resource-constrained project scheduling: an update, *European Journal of Operational Research*, Vol.174 (1), 2006, pp. 23-37.

[6] D. Debels, B. De Reyck, R.Leus and M.Vanhoucke, A Hybrid Scatter Search/Electromagnetism Meta- Heuristic for Project Sheduling, *European Journal of Operational Research*, Vol. 169, 2006, pp. 638-653.

[7] D. Debels and M. Vanhoucke. *A Decomposition-Based Heuristic for the Resource-Constrained Project Scheduling Problem*. Working Paper 2005/293, Faculty of Economics and Business Administration, University of Ghent, Ghent, Belgium, 2005.

[8] J.J.M. Mendes, J.F. Gonçalves and M.G.C. Resende, A random key based genetic algorithm for the resource constrained project scheduling problem, *Computers & Operations Research*, Vol. 36, 2009, pp. 92-109

[9] K. Fleszar and K.S. Hindi, Solving the resource constrained project scheduling problem by a variable neighbourhood search, *European Journal of Operational Research*, Vol. 155, 2004, pp. 402-413.

[10] J. Magalhaes-Mendes, “Project scheduling under multiple resources constraints using a genetic algorithm”, WSEAS TRANSACTIONS on BUSINESS and ECONOMICS, Issue 11, Volume 5, November 2008.

[11] Turner J R. The hand book of project based management [M].London: Mc-Graw Hill,1993.

[12] MaoYi Hua. Application for network optimization technique in construction claim management .

[13] Garey M., Johnson D., *Computers and Intractability: a Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.

[14] Milena Karova, Julka Petkova, Vassil Smarkov, “ A Genetic Algorithm for Project Planning Problem”, International Scientific Conference Computer Science’2008.

[15] Holland J., *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, 1975.

[16] D. A. Coley, “An Introduction to Genetic Algorithms for Scientists and Engineers”, New Jersey, 1999.

[17] E. Zitzler, K. Deb, L. Thiele, Comparison of multi-objective evolutionary algorithms: empirical results, *Evolutionary Computation* 8 (2000) 125–148.

[18] Ghosh, A. and S. Dehuri, 2004. Evolutionary algorithms for multi-criterion optimization: A survey. *Intl. J. Comp. Inform. Sci.*, 2: 38–57.