An Optimization Model And Genetic Algorithm Solution For Software Projects

Yücel Dil[1]
Mustafa Cem Kasapbaşı [2]

Abstract
Many optimization techniques which are inspired by the nature are used in optimization problems. Genetic Algorithms (GA) is an optimization algorithm, tries to mimic the natural process of livings. Allowing to survive better generation therefore inheriting the better qualifications to next generations. In this study genetic algorithm is used to find the optimal cost for a software project. In order to evaluate results of the genetic algorithm, a test system based on linear programming is established. The results indicates that designed genetic algorithm optimization model successfully calculated the cost of software project very close to deterministic costs.

Keywords: *Software Project Management, Genetic Algorithm, Optimization*

I. Introduction

Software project management aims to achieve all the project goals and objectives while working within the constraints posed by project environment and stakeholders. These constraints include (but not limited to) time, scope, resources, resource allocation and optimization etc. [1].

Software project management (SPM) is the art and science of planning and leading software projects [2]. According to [4] a survey conducted in the industry only about a quarter of software projects are regarded as successful therefore billions of dollars are lost annually due to the project failures or unsatisfactory project deliveries. Many problems can cause such results but it is mainly because of failing to understand and manage software project risks [5], also not having a proper quantitative cost calculation tool therefore letting the project being guided by subjective decisions of project manager. Unable to comprehend project entirely may lead problems like cost schedule overruns, unmet user requirements.

Software management can be defined as keeping team together on the same purpose, distributing tasks while keeping healthy intercommunication between team members,

[1] *Computer Engineering Department, Istanbul Commerce University, Istanbul, Turkey, yuceldil@gmail.com*
[2] *Computer Engineering Department, Istanbul Commerce University, Istanbul, Turkey, mckasapbasi@ticaret.edu.tr*

at the end of each task evaluating results properly to asses overall progress [6].

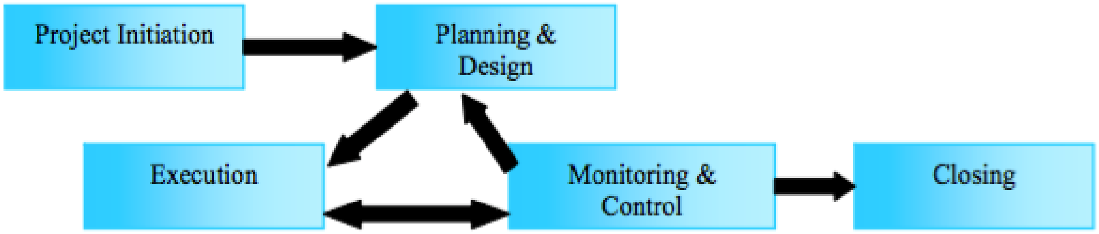

Fig. 1. Traditional Project Management Stages [1]

II. Background Study and Methodology

A. Literature Review

Since Software Project Management is considered a subclass of Project Management there are many researches, tools and methodology could be used to asses software projects. However software project managements are highly depended on manager's subjective judgments. Therefore some means of quantitative tools and decision helpers are needed for varıous stages of Software Projects.

One of such stage is Risk Control of a software project management. In [3] Capability Maturity Model based (CMM) risk assessment system is proposed. In that study previous similar project results are used as database in obtaining the problem solution via a dynamic programming method. An other Risk Control optimization model is proposed in [5] utilizing particle swarm optimization methodology to represent some means of quantitative data to software project manager.

An other approach is to determine the software metrics and modeling them to assist the software project manager. A successful fuzzy model for software metrics is presented to better analyze the time vs. performance vs. information vs. cost tradeoffs that are entailed in software project management [7].

III. Methodology And Application

Same software project management experiment scenario is conducted for two different methods namely Linear Programming and Genetic Algorithm, which are used to optimize the cost of the project. At the end, these two methods are compared weather they are consistent with each other.

Conducted Experiment Scenario:
In the software project 10 people are going to be allocated 6 of them are process analysts, 4 of them are software developers. Qualifications and experience of these people also the cost of them are known from previous projects in which they have involved.

TABLE I. MAN HOURS DISTRIBUTION AND COST OF THE WORKERS, VARIABLE NAMES OF COST FUCNTION

Worker	**Work (man hour)**												**Wages per Hours (TL)**
	Analysis			***Coding***			***Testing***			***Activating***			
	Cost	***LP Cost***	***Variable name***	***Cost***	***LP Cost***	***Variable name***	***Cost***	***LP Cost***	***Variable name***	***Cost***	***LP Cost***	***Variable name***	
D_1	-	-	-	84	3192	X_1	78	2964	X_2	66	2508	X_3	38
D_2	-	-	-	82	2296	X_4	80	2240	X_5	64	1792	X_6	28
D_3	-	-	-	102	2448	X_7	78	1872	X_8	68	1632	X_9	24
D_4	-	-	-	106	2756	X_{10}	76	1976	X_{11}	68	1768	X_{12}	26
A_1	70	1960	X_{13}	-	-	-	66	1848	X_{14}	54	1512	X_{15}	28
A_2	94	2256	X_{16}	-	-	-	52	1248	X_{17}	56	1344	X_{18}	24
A_3	78	2964	X_{19}	-	-	-	58	2204	X_{20}	62	2356	X_{21}	38
A_4	86	3096	X_{22}	-	-	-	70	2520	X_{23}	60	2160	X_{24}	36
A_5	112	2240	X_{25}	-	-	-	64	1280	X_{26}	52	1040	X_{27}	20
A_6	102	2244	X_{28}	-	-	-	72	1584	X_{29}	50	1100	X_{30}	22

This software project consists of 4 main phases namely analyzing (Planning & Designing), writing codes (Execution), testing (Monitoring & Control), and activating the project (Closing). These phases comply with the generic project management process that can be seen from Fig. 1. In Table 1 according to job assignments of workers, unit costs are given Turkish Lira and man hours are depicted accordingly. D stands for Developer, A stands for Analyst. As can be inferred from the Table 1 that some analysts and developers cost more then others since they have different experience levels.

As in every project this project has constrains that are guaranteed by the contract signed by parties. Complying these constraints is one of the responsibilities of the software project manager.

- It is expected that every developer must work at least 8 hours and every analyst must spent at least 16 hours on the project.
- Maximum 86 hours for coding, 100 hours for analysis, 80 hours for tests, and for activating project 64 hours must be separated.
- According to accepted quality assurance standard of the company at least 82 hours for coding, 80 hours for analyzing, 56 hours for testing and 60 hours for activating the project must be separated.

It is expected from and responsibility of the software project manager that while being coherent with the constraints obtaining the minimum cost with the optimum resource planning.

Project is regarded and designed as a minimization focused optimization problem. This problem is solved using with Linear Programming (LP) and Genetic Algorithms (GA) then results are compared with each other.

For linear Programming Eq.1 is utilized as cost function and minimum cost is obtained accordingly.

$$\text{Min } C = 3192\,X_1 + 2964\,X_2 + 2508\,X_3 + 2296\,X_4 + 2240\,X_5 + 1792\,X_6 + 2448\,X_7 + 1872\,X_8 + 1632\,X_9 + 2756\,X_{10} + 1976\,X_{11} + 1768\,X_{12} + 1960\,X_{13} + 1848\,X_{14} + 1512\,X_{15} + 2256\,X_{16} + 1248\,X_{17} + 1344\,X_{18} + 2964\,X_{19} + 2204\,X_{20} + 2356\,X_{21} + 3096\,X_{22} + 2520\,X_{23} + 2160\,X_{24} + 2240\,X_{25} + 1280\,X_{26} + 1040\,X_{27} + 2244\,X_{28} + 1584\,X_{29} + 1100\,X_{30} \qquad (1)$$

When we have presented this project as mathematically we presented 30 variables and 18 constraints. MATLAB program is utilized to solve the same problem with Genetic Algorithm.

TABLE II. RESULTS OF BOTH LP AND GA OPTIMIZATION

Workers and Stage Steps	Working Time Constraints	Lineear Programming		Genetic Algorithm	
		Work Duration	Deviation (%)	Work Duration	Deviation (%)
D1	$X_1 + X_2 + X_3 >= 8$	8	0	8	0
D2	$X_4 + X_5 + X_6 >= 8$	58	0,5	68,2	0,6
D3	$X_7 + X_8 + X_9 >= 8$	8	0	8	0
D4	$X_{10} + X_{11} + X_{12} >= 8$	8	0	8	0
A1	$X_{13} + X_{14} + X_{15} >= 16$	80	0,64	77,64	0,62
A2	$X_{16} + X_{17} + X_{18} >= 16$	40	0,24	43,98	0,28
A3	$X_{19} + X_{20} + X_{21} >= 16$	16	0	16	0

A4	$X_{22} + X_{23} + X_{24} >= 16$	16	0	16	0
A5	$X_{25} + X_{26} + X_{27} >= 16$	28	0,12	16	0
A6	$X_{28} + X_{29} + X_{30} >= 16$	16	0	16,17	0
Coding	$82 <= X_1 + X_4 + X_7 + X_{10} <= 86$	82	0	82	0
Analysis	$80 <= X_{13} + X_{16} + X_{19} + X_{22} + X_{25} + X_{28} <= 100$	80	0	80	0
Test	$56 <= X_2 + X_5 + X_8 + X_{11} + X_{14} + X_{17} + X_{20} + X_{23} + X_{26} + X_{29} <= 80$	56	0	56	0
Activating	$60 <= X_3 + X_6 + X_9 + X_{12} + X_{15} + X_{18} + X_{21} + X_{24} + X_{27} + X_{30} <= 64$	60	0	60	0
Project Cost		523.600 TL		533.919 TL	
Total Deviation		1,46		1,5	
Solution Time		0,301884 sn		2,029242 sn	

After 200 generation desired solution has obtained for GA. While calculating with GA Linear generation function is preferred for generating generations. Two point crossover method is chosen for crossover function. For selection process Tournament method is preferred. Same constraints and fitness function is used for evaluation as use in LP. Results are depicted in Table II. As one can understand from the Table II that cost calculation of LP and GA are both similar.

A. *After Project Non linear Programming*

After the project minimum cost is applied and being used. Due to quality control standards requirements each cost of application used in the project can be calculated in a certain tolerance of error. If this rate of error goes beyond the pre determined tolerance new version of cost calculation is carried out.

Error rate that will be experienced according to usage amount (number of steps) is chosen as $2X_1^{0,1}$. This error rate corresponds and composed of due to misusage and application. In Eq. 2 X_1 represents number of steps, X_2 represents number of errors due to misusage. Relation of number of steps and error due to misusage is depicted in Eq. 2.

$$F(X) = X_1^{0,3} + 2X_2^{0,2} \qquad (2)$$

It is desired from project manager to assign and work with maximum number of users in the project without publishing e new version of the cost calculation and schedules. Since these calculations always alter the initial plan therefore modifies it.

The constraints that should be take in to consideration by the project manager are as fallows:

- In the case errors due to application rises over 75 a new application release should be calculated and presented. This constraint is illustrated in Eq. 3.

$$2X_1^{0,1} - X_2 <= 75 \qquad (3)$$

- Total number of errors must be less then 100. This constraint is illustrated in Eq. 4.

$$2X_1^{0,1} <= 100 \qquad (4)$$

- All of the errors can not be resulted from misusage. This constraint is illustrated in Eq. 5 and Eq. 6.

$$2X_1^{0,1} - X_2 >= 0 \qquad (5)$$

$$X_1 >= 0, X_2 >= 0 \qquad (6)$$

Even in this case what is expected from the project manager is also considered and investigated as an optimization problem. Designed problem is non linear and have non linear constraints Cases of which are regarded as hard solution problems. In order to solve the problem MATLAB program is utilized

IV. CONCLUSIONS

In this study cost optimization problem of software project is analyzed using GA applied to LP and Non LP. Sample constraints and cases that can be encountered in a software project are chosen. As a result it has been emphasized that software development projects can be designed as an optimization problem and a solution can be proposed to the problems may be encountered while activating the project. Another contribution is the use of Genetic Algorithm approach in software project management processes.

REFERENCES

[1] Ramzan, M.; Iqbal, M.A.; Jaffar, M.A.; Rauf, A.; Anwar, S.; Shahid, A.A., "Project Scheduling Conflict Identification and Resolution Using Genetic Algorithms," in *Information Science and Applications (ICISA), 2010 International Conference on* , vol., no., pp.1-6, 21-23 April 2010 doi: 10.1109/ICISA.2010.5480400

[2] Stellman, Andrew; Greene, Jennifer. “Applied Software Project Management”. O'Reilly Media, 2005

[3] Xu Ruzhi; Qian leqiu; Jing Xinhai, "CMM-based software risk control optimization," in Information Reuse

and Integration, 2003. IRI 2003. IEEE International Conference on , vol., no., pp.499-503, 27-29 Oct. 2003 doi: 10.1109/IRI.2003.1251457

[4] Kwak, Y. H., & Stoddard, J. (2004). Project risk management: Lessons learned from software development environment. Technovation, 24(11), 915–920.

[5] Ping Cao; FuJi Chen, "A Risk Control Optimization Model for Software Project," in Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on , vol., no., pp.1-4, 11-13 Dec. 2009 doi: 10.1109/CISE.2009.5362886

[6] Güney D. D.,Yönetim ve Organizasyon. Nobel Publishing Dağıtım 2007.

[7] Mirseidova, S.; Atymtayeva, L., "Definition of software metrics for software project development by using fuzzy sets and logic,", Joint 6th International Conference on Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS), 2012 vol., no., pp.272-276, 20-24 Nov. 2012

[8] doi: 10.1109/SCIS-ISIS.2012.6505336