International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Optimization of COCOMOII Model Coefficients using Tabu Search

Richika Chadha[1], Shakti Nagpal[2]

[1, 2] CSE Department, Geeta Engineering College, Panipat, India

Abstract: ***Software effort estimation is one of the most important step in project planning. Accurate estimates help the company in decision making process and effectively manage the software development process. Various Software cost estimation models have been proposed till now, COCOMO'81 is one of them. It became one of the most popular parametric cost estimation model in 1980's but when new life cycle processes came it experienced some difficulties in calculating cost. COCOMOII model came in 1995 to overcome these problems and became one of the most commonly used models for estimating effort. Today's models are based on various soft computing techniques such as fuzzy logic, neural network, genetic algorithm, simulated annealing, tabu search etc. Fuzzy logic and neural network techniques are very hard to use and genetic algorithm cannot produce very good results because of its problem of premature convergence. This work aims to propose Tabu Search algorithm which can increase the efficiency of output values of parameters of COCOMOII model and can evaluate predicted effort values as accurate to the real values.***

Keywords: COCOMOII, effort multipliers, scale factors, size, tabu search.

1. Introduction

Software cost estimation is a process of predicting the effort required to develop a software engineering project.[1] Software cost does not directly refer to the monetary value of the software development. It contains two main questions: "What's the effort involved?" and "How long will it takes?". The answers to these questions can be translated to monetary value. In reality, software cost consists of three elements: manpower loading, effort and duration. Manpower loading is the number of engineering and management personnel allocated to the project. Effort is the engineering and management effort required to complete a project usually measured in unit such as person-months. Duration is the amount of time required to complete the project (usually measured in months). Software cost directly depends on items such as analysis, design, coding, testing and integration. Other than these it also includes some other items such as training, customer support, installation, level of documentation, configuration management and quality assurance.[2] Many software estimation models have been proposed so far. Among them COCOMOII is widely used method because of its simplicity. In this paper, COCOMOII model is used with tabu search approach for optimizing the parameters of COCOMOII model so that it can predict accurate effort values of software project. Tabu search is a metaheuristic search technique which employs local search methods for finding optimum solution. Tabu search is created by Fred W. Glover in 1986 and formalized in 1989.

2. COCOMO II model

The COCOMO (COst COnstructive MOdel) cost and schedule estimation model was originally proposed by Dr. Barry Boehm in 1981. But COCOMO'81 experienced some difficulties in estimating cost to new life-cycle processes and capabilities. Later COCOMOII model came in 1994 to address the issues on non-sequential and rapid development process models, reengineering, reuse driven approaches, and object oriented approaches. COCOMOII has three sub models: Application Composition, Early Design and Post Architecture model.

(i) **Application Composition Model:** This model is used to estimate effort and schedule on projects which uses Integrated Computer Aided Software Engineering Tools for rapid application development. It uses object points for sizing.
(ii) **Early Design Model:** This model involves the exploration of software and system architectures and concepts of operation. It is based on function point (or lines of code when available) and contains 7 scale factors and 5 effort multipliers.
(iii) **Post Architecture Model:** This model is the detailed extension of early design model and estimates for the entire development lifecycle. It is used when top level design is complete and detailed information about the project is available. It uses source of lines of codes and/or function points for sizing, a set of 17 effort multipliers and 5 scale factors[2]

COCOMOII model describes 17 cost drivers in Product, Personnel, Computer and Project categories and also 5 scale factors .[4]

Table 1: Cost drivers for COCOMO-II PA model [4] [5]

Cost Drivers	*Description*	*Type*
RELY	Required software reliability	Product
DATA	Data base Size	Product
RUSE	Developed for Reusability	Product
DOCU	Documentation needs	Product
CPLX	Product complexity	Product
TIME	Execution Time Constraints	Computer
STOR	Main Storage Constraints	Computer
PVOL	Platform Volatility	Computer
ACAP	Analyst Capability	Personnel
PCAP	Programmer Capability	Personnel
APEX	Application Experience	Personnel
PLEX	Platform Experience	Personnel
LTEX	Language ad tool experience	Personnel
PCON	Personnel Continuity	Project
TOOL	Use of Software Tools	Project

SITE	Multi site Development	Project
SCED	Required development schedule	Project

Table 2: Scale factors for COCOMO II PA model [4][5]

Scale Factors	*Description*
PREC	Precedentedness
FLEX	Development Flexibility
RESL	Risk Resolution
TEAM	Team Cohesion
PMAT	Process Maturity

In COCOMOII, effort can be calculated by the following equation:

$$\text{Effort (PM)} = A\times (\text{SIZE})^{E}\times \Pi_i\, EM_i \text{ .... [5](1)}$$

Where A is the multiplicative constant with value 2.94 that scales the effort according to project conditions.
Size – COCOMOII expresses size of project in thousands of Source Line Of Code (SKLOC). [6]
EM_i - are Effort Multipliers where i =1, 2, 3….17.

E – is an exponent which is aggregated of five Scale Factors that describe relative economies and diseconomies of scale that are encountered for software projects of dissimilar magnitudes. [6]

$$E= B + 0.01\sum_j SF_j \text{ where j=1 to 5 ....[6] (2)}$$

B is a multiplicative constant whose value is 0.91
The development time (TDEV) of the project is derived from the equation:

$$TDEV= C*(\text{Effort})^F \text{ ....[5]} \quad (3)$$

C is a multiplicative constant whose value is 3.67 and the coefficient F can be determined by following equation:

$$F = D + 0.2*0.01* \sum_j SF_j \text{ .... [5]} \quad (4)$$

where D=0.28
Or
F= D + 0.2*(E-B)

When all the scale factors and effort multipliers are taken with their nominal values, then the equation of effort and duration are:

$$\text{Effort} = 2.94 \times (\text{Size})^{1.1} \text{ ....[5]} \quad (5)$$

$$\text{Duration: } TDEV = 3.67 \times (\text{Effort})^{3.18} \text{ ....[5]} \quad (6)$$

COCOMOII is an industry standard and having clear and effective calibration process by combining Delphi techniques with algorithmic cost estimation techniques (Bayesian approach) and having backward compatibility with Rosetta Stone. The main disadvantage of COCOMOII model is that it based on waterfall model and most of the extensions are still experimental and not fully calibrated till now. [6]

3. Comparative Study

Various optimization techniques have been used earlier in the proposed model such as genetic algorithm, particle swarm optimization, neural network and many more but they have some disadvantages associated with them, that is why they cannot produce better results than tabu search.

a) **Genetic algorithm:** the main problem of genetic algorithm is premature convergence which does not allow it to access whole solution space constraining it to converge to a local optimum.
b) **Neural Network:** the main disadvantage is, they are black box i.e. the knowledge of internal working is never known and secondly to fully implement a neural network architecture would require a lot of computational resources.
c) **Particle Swarm Optimization:** this technique suffers from partial optimism, which causes the less exact at the regulation of its speed and its direction and also the method cannot work on the problems of scattering and optimization. Advantages of Tabu Search:
   - It can be applied to both discrete and continuous solution spaces.
   - For larger and more difficult problems tabu search obtains solutions that rival and often surpass the best solutions previously found by other approaches.

4. Proposed Work

Objective: the main aim of this work is to use the concept of tabu search to optimize the COCOMO II model coefficients to achieve accurate software effort estimation. Tabu search reduces the uncertainty of COCOMO II Post Architecture model coefficients i.e. a,b,c,d so that it can produce accurate effort results near to the actual values.

Dataset Description: Experiments have been conducted on Turkish and Industry data set presented by Ekananta Manalif [8] to optimize effort. The dataset consists of three variables i.e. Size in Kilo Line of code (KLOC), Actual effort and the predicted effort using COCOMO II PA model. The dataset is given in Table 3. Effort multipliers and scale factors rating from Very Low to Extra High related to fifteen projects are taken from Appendix B of [8].

Table 3: Data sets with their size and effort values [8]

Pr. No.	Size (KLOC)	Actual Effort (PM)	COCOMO II Model Predicted Effort (PM)
1	002.00	002.00	002.90
2	114.28	018.00	294.00
3	064.10	332.00	256.70
4	023.11	004.00	063.20
5	001.37	001.00	000.90
6	001.61	002.10	002.00
7	031.85	005.00	147.10
8	131.00	619.90	745.20
9	010.00	003.00	036.20
10	015.00	004.00	063.20
11	004.25	004.50	009.30
12	004.05	002.00	002.30
13	019.90	074.60	092.70
14	003.00	001.20	003.60
15	040.53	002.00	028.60

Tabu Search based approach to optimize estimated effort: Tabu Search which is proposed by Fred Glover in 1986 is a metaheuristic search technique which employs local search methods for finding optimum solution. Local search techniques takes a potential solution of a problem and then find its neighbours which are basically similar but having a minute difference in their details. Through these neighbours we can find out an improved solution. But this technique has a problem of becoming stuck at suboptimal solutions or on plateaus where many solutions are equally fit. Tabu search overcome the problem of this technique by using memory structures that describes the visited solutions and a set of rules. The basic principle of tabu search is to pursue the search whenever a local optimum is encountered by allowing non-improving moves If any solution is previously visited or violating any rule is considered as "tabu" so that the solution will not considered repeatedly.

Tabu Search algorithm starts with an initial solution to the problem, calls it a current solution, and further create its neighbourhood (a collection of solutions which can be easily reached from current solution) and tries to find out a best solution from its neighbourhood. It then designates the best solution as the current solution and starts the search process again. The search process gets terminate when some stopping criteria has been met, for example execution time, prespecified iteration counts, solution quality etc. In order to prevent repeatedly considering a solution that has been recently visited a list has been maintained called tabu list which contains a list of neighbour generated moves that has been considered forbidden and are ignored while searching the neighbourhood of a solution. Once a move enters in tabu list, it remains there for a pre-specified number of tabu search iterations (known as tabu tenure of the move). After the completion of tabu tenure of the move, it can be reached again while searching in the neighbourhood. The list of tabu moves changes continuously during the execution of the search, making tabu search an adaptive memory search algorithm. When the stopping criteria met, we get current solution as the best solution.[7]

Proposed Algorithm: Tabu Search algorithm is proposed to optimize the COCOMOII PA model coefficients. The main steps of tabu search algorithm are:

Step 1: Set the iteration counter k=0 and randomly generate an initial solution $x_{initial.}$ Set this solution as the current solution as well as the best solution, $x_{best,}$ i.e. $x_{best} = x_{initial} = x_{current}$.

Step 2: Randomly generate a set of trial solutions x_{trial}s in the neighborhood of the current solution, i.e. create $S(x_{current})$. Sort the elements of S based on their objective function values in ascending order as the problem is a minimization one. Let us define x_{trial}^{i} as the i[th] trial solution in the sorted set, 1<= i<=num of trials. Here, x_{trial}^{i} represents the best trial solution in S in terms of objective function value associated with it.

Step 3: Set i=1. If $J(x_{trial}^{i}) > J(x_{best})$ go to step 4, else set $x_{best} = x_{trial}^{i}$ and go to step 4.

Step 4: Check the tabu status of x_{trial}^{i}. If it is not in the tabu list then put it in the tabu list, set $x_{current} = x_{trial}^{i}$ and goto step 7. If it is not in the tabu list go to step 5.

Step 5: Check the aspiration criterion of x_{trial}^{i}. if the satisfied then override the tabu restrictions, update the aspiration level, set $x_{current} = x_{trial}^{i}$ and go to step 7. If not, set i=i+1 and go to step 6.

Step 6: If i>num of trials go to step 7, else go back to step 4.
Step 7: Check the stopping criteria. If one of them is satisfied then stop, else set k=k+1 and go back to step 2. [9]

5. Result Analysis

In this experiment, Tabu Search algorithm has been applied to optimize the COCOMO II PA model coefficients a,b,c,d and it has been tested on TURKISH and INDUSTRY dataset of 15 projects. The main purpose of this experiment is to give better estimation of effort values of projects which are near to the actual values unlike using COCOMOII PA model coefficients. Current COCOMO II model coefficients are following:

a=2.94; b=0.91; c=3.67; d=0.28.

The best solution is achieved using many iterations, however in my work default number of iterations is 1000. A solution set of individuals is received from which best individual with best fitness function value is chosen. The final fittest individual value obtained is:
a= 2.66 b=0.567 c=3.606 d=0.785

The following Table 4 shows the comparison among the Actual Effort values and Estimated Effort values for the last five project dataset using the Tabu Search algorithm optimized and current COCOMO II PA model coefficients. At the same time, in the results obtained using the coefficients optimized by Tabu Search algorithm, the error is much lower, but it still persists.

Table 4: Predicted development Effort values

Project	*Calculated Estimated Effort*	*Calculated Effort By Cocomo 2 Coefficients*	*Actual Effort*
Project 1	1.7	2.90	2
Project 2	3.04	36.20	3.0
Project 3	5.51	9.30	4.5
Project 4	1.20	3.60	1.2
Project 5	2.40	28.60	2.0

The graphical representation of Table 3 is shown in following Figure 1. According to this figure it can be easily figured out that estimated effort values using tabu search are much better than the COCOMO II PA model coefficients calculated effort values. So it can be concluded that Tabu Search algorithm can offer significant improvement in accuracy and has the potential to be a valid additional tool for the software effort estimation.

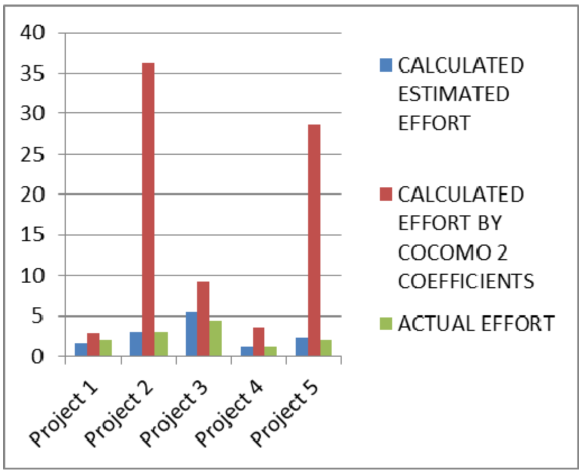

Figure 1: Graph showing comparison among Efforts

6. Conclusion

In this paper, Tabu Search algorithm is tested on TURKISH and INDUSTRY dataset and the obtained results are compared with the ones obtained using the current COCOMO II PA model coefficients. It is concluded that by comparing the results, Tabu Search based coefficients can be used to produces better results as compared to the results obtained using the current COCOMO II PA model coefficients. The application of tabu search in this work suggests the useful potential of this approach and its underlying principles.

This research indicates directions for further research. The proposed framework can be analyzed in terms of feasibility and acceptance in the industry. Trying to improve the performance of existing methods and introducing the new methods for estimation based on today's software project requirements can be future works in this area. So the research is on the way to combine different techniques for calculating the best estimate.

References

[1] Maged A. Yahya, Rodina Ahmad, Sai Peck Lee, "Effects of Software Process Maturity on COCOMOII"s Effort Estimation from CMMI Perspective" In 2008 IEEE, Department of Software Engineering, University of Malaya, pp. 255-256.

[2] Andre Ladiera, "Cost Estimation Methods for Software Engineering", Rand Africaans University ,2002.

[3] Astha Dhiman, Chander Diwakar "Optimization of COCOMO2 Effort Estimation Using Genetic Algorithm", Department of Computer Engineering, University Institute of Engineering and Technology, in 2013 AIJRSTEM ISSN: 2328-3491, pp-208-212.

[4] Darko Milicic, "*Applying COCOMO II - A case study*" Master Thesis Software Engineering, Thesis no: MSE-2004-19, Aug. 2004, School of Engineering Blekinge Institute of Technology, Sweden.

[5] Bogdan Stępień, "Software Development Cost Estimation Methods and Research Trends" Computer Science, Vol. 5, 2003, pp. 68-82.

[6] Nancy Merlo–Schett, "Seminar on Software Cost Estimation" Requirements Engineering Research Group, Department of Computer Science, WS 2002/2003, pp. 3-19.

[7] Sumanta Basu,"Tabu Search Implementation on Traveling Salesman Problem and Its Variations: A Literature Survey", Indian Institute of Management, Calcutta in2012,***American Journal of Operations Research*, pp- 163-173**

[8] Ekananta Manalif, "*Fuzzy Expert-COCOMO Risk Assessment and Effort Contingency Model in Software project Management*" For the degree of Master of Engineering Science, The School of Graduate and Postdoctoral Studies, The University of Western Ontario London, Ontario, Canada.

[9] V.Sanjeeva Rao, Dr.B.V.Shankarram, "Design of a Power System Stabilizer for asynchronous Generator Using

[10] Hybrid Intelligent Controller", JNTUH,Kukatpally, Hyderabad in 2014, International Journal Of Engineering And Computer Science ISSN:2319-7242, pp. 5524-5528.