ISSN:0975-9646

Jagjit Kaur et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 4463-4465

Parameter Estimation of COCOMO II using Tabu Search

Jagjit Kaur[1], Ritu Sindhu[2]

[1]*Research Scholar,* [2]*Associate Professor,*
Department of Computer Science
SGT Institute of Engineering & Technology, Gurgaon,
Haryana,India

Abstract--- **Estimating the cost and effort of software product is one of the biggest challenges. With the growth of technology and changing frequent user requirements it becomes very difficult to make accurate software cost estimation. But the accurate results are required for proper project planning because any error results in huge losses. The objective of software cost estimation method is to estimate the cost and effort required for software production. The cost estimation is usually dependent upon the size estimate of the project. There are several different techniques for performing software cost estimation among which COCOMO II is commonly used because of its transparency and simplicity. There are several soft computing techniques to optimize the effort estimated. Various optimization techniques such as Genetic algorithm, neural network, simulated annealing and many more were used earlier but in proposed model the author used the best technique called Tabu Search. The proposed model will tend to reduce the uncertainty of COCOMO II post architecture model coefficients i.e. a, b, c and d. Tabu search provides the optimized results in comparison with all other methods.**

Keywords--- **COCOMO II, Tabu Search, Scale factors, Effort multipliers, Size**

I. INTRODUCTION

Today Softwares are becoming increasingly very important. The most important thing in software for its users is the cost which is very difficult to estimate. This software cost would affect the entire software management process which includes project planning, scheduling and resource allocation. Effort is usually estimated in person month and it can be translated into actual cost. So calculating the cost and effort is the major task which in turn calculates the overall employees required for the project. Among many models COCOMO II is the best used model in calculating the effort for the software project but there are various techniques which are used to optimize the coefficients of COCOMO II model to get the estimated effort equivalent to actual effort and better than current COCOMO II current coefficient effort. These techniques are like such as Genetic algorithm, neural network, simulated annealing and many more were used earlier [5] but in proposed model the author used the best technique Tabu search which provides the optimized results in comparison with all other methods.

- Genetic Algorithm:- The main disadvantage of GA is of premature convergence and it does not consider the worst solution
- Neural network:- The main limitation of neural network, it makes a particular decision in a difficult task due to the shortcomings of black boxes..
- Simulated Annealing (SA):- The limitation of SA in comparison with other techniques such as it is easier to code even for complex problems but due to its meta-heuristic approach it needs a lot of choices to turn it into an actual algorithm.

Tabu search will increase the efficiency of COCOMO II model. Tabu search is a Meta heuristic local search algorithm that uses the concept of flexible memory both long term and short term to avoid local minima and cycling. Tabu technique is directly applied on continuous functions. Tabu search has been applied to a number of problems like instruction scheduling [6], job shop scheduling [5] employee scheduling, time table management**.** [6]

II. COCOMO II MODEL

COnstructive COst MOdel (COCOMO 81) is the parametric software cost estimation model which was developed by Barry W. Boehm. It uses an algorithmic method to evaluate the cost of the software. It is based on waterfall lifecycle model [3]. It is a transparent model which provides all the details like assumptions, definitions and cost estimates of the model. Besides all these it suffers from some limitations like:

- For each phase of project life cycle separate estimation model should be used.
- Not a realistic model as assumptions and requirements vary with time.
- Not suitable for non sequential, reuse case models and object oriented models.

COCOMO II, a successor of COCOMO 81, which was developed in 1995 overcome all these limitations faced by its predecessor [1]. It is more accurate model. It takes qualitative inputs and produce quantitative results. COCOMO II has three sub models application composition model, early design model and post architecture model [1]. Of all these COCOMO II uses post architecture model which is a detailed model and it has been prepared after the architecture has been designed. COCOMO II post architecture takes cost drivers, scale factors and size as three most important inputs. The values of the cost drivers and scale factors depend upon the model being used.

Cost drivers are the characteristics of the software that influence the effort in carrying out a certain project. COCOMO II has as an input a set of seventeen Effort Multipliers (EM) or cost drivers which are used to adjust the nominal effort (PM) to reflect the software product being developed and five scale factors which have rating levels are Very Low (VL), Low (L), Nominal (N), High (H), Very High (VH) and Extra High (XH)

TABLE I
COCOMO II SCALE FACTORS [8]

Scale factor	Description
PREC	Precedentedness
FLEX	Development flexibility
RESL	Risk resolution
TEAM	Team cohesion
PMAT	Process maturity

TABLE II
COCOMO II COST DRIVERS [8]

Cost Driver	Description
SCED	Schedule
SITE	Multi site development
RUSE	Reusability
DOCU	Documentation needs
CPLX	Product complexity
TIME	Execution time
STOR	Storage
PVOL	Platform volatility
ACAP	Analyst capability
APEX	Application experience
PLEX	Platform experience
LTEX	Language and tool experience
PCON	Personnel continuity
TOOL	Software tools used
PCAP	Programmer capability
DATA	database size
RELY	Reliability

COCOMO II calculates the effort in person month by using the following equation:
Effort (PM) = A x $SIZE^E$ x $\prod_i EM_i$ [2][8]
Where A is multiplicative constant having a value of 2.94
As we know size is the most important factor in calculating the effort of the software project and it is measured in Kilo Source Line Of Code (KSLOC), so E can be calculated as

E= B + 0.01 x $\sum_i$ SFi [2][8]
Where B is a constant= 0.91.
In this way effort is calculated in person month.
But the software companies are also more interested in calculating the duration the project lasts. So it can be calculated from effort as

T (development time) = C x $(PM)^F$ [2][8]
Where C is constant = 3.67
F can be calculated as
F = D + 0.2 x 0.01 x $\sum SF_i$ [2][8]
Where D is constant = 0.28.

III. TABU SEARCH

Tabu search is a Meta heuristic search technique created by Fred W. Glover in 1986 for solving the complex reliability problems. It continues iterate in a loop until it gets an accurate result. It makes the use of short term, long term and working memory [7]. It is basically a local search method to escape from local minima. Tabu uses the concept of tabu list which is also known as short term memory which avoids the cycling because all the recent history of previously visited solutions is stored in the tabu list and it forbids the same moves again.
Tabu search algorithm works by initially selecting a random number from a solution space and then put that number in the tabu list as well as to the current best solution. Whenever a new neighbour is chosen from a set of individuals then it is compared with the tabu list. If that is already in the tabu list then it is avoided, if not then it is compared with the individual stored in the current best solution. If its value is better than that then the value gets replaced and tabu list gets populated with the new individual. At each iteration, a steepest-descent solution is chosen. Whenever tabu list gets completely occupied then it frees its memory by using the concept of first come first out. In this way tabu search first performs diversification to choose the neighbourhood area of the solution space and intensification of search with the help of flexible short term memory [10]. Tabu generally also considers the worst solution as shown below:

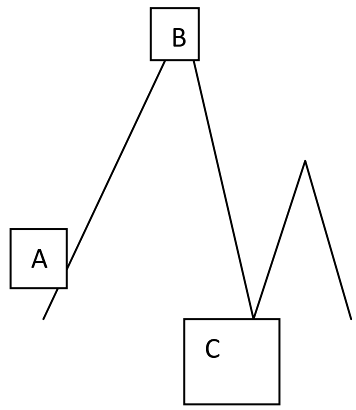

Fig 1. Consideration of worst solution in Tabu

If a tabu search gets a best solution on going from A to B then it also traverse a path from B to C in the hope of getting a more better solution. In this way it considers the worst solution if any from going through B to C.

Pseudo code for Tabu search
Step 1: Choose an initial solution in search space
Step 2: Create a set of neighbor solutions
Step 3: Initialize a Tabu list.
Step 4: Evaluate the neighboring solutions in a search space.
Step 5: Choose the best admissible solution
Step 6: Update the tabu list and aspiration criteria.
Step 7: If a stopping condition is met then stop. Else go to Step 2. [7]

IV. Experiment And Results

The main objective of the Experiment performed is to reduce the uncertainty of current COCOMO II post architecture coefficients i.e. a, b, c and d and get the best software effort estimation results equivalent to actual effort using tabu search algorithm. The proposed algorithm is tested on Turkish and industry dataset on 15 different projects. This dataset consist of size of each project in kilo source line of code, actual effort and the COCOMO II current coefficient effort in Person Month (PM) in Table III.

Table III
Data Sets With Their Size And Effort Values [9]

P.No	Size	Actual effort	COCOMO II predicted effort
1	003.0	001.20	003.60
2	019.90	074.60	092.70
3	004.05	002.0	002.30
4	004.25	004.50	009.30
5	015.00	004.0	063.20
6	010.00	003.00	036.20
7	131.00	619.90	745.20
8	031.85	005.00	147.10
9	001.61	002.10	002.00
10	001.37	001.00	000.90
11	023.11	004.00	063.20
12	064.10	332.00	256.70
13	114.28	018.00	294.00
14	002.00	002.00	002.90
15	040.53	022.00	028.60

The seventeen cost drivers/ effort multipliers and five scale factors are taken from [9].
The working is implemented on NetBeans IDE 8.0.
Current COCOMO II PA coefficients areas:
a= 2.94, b= 0.91, c= 3.67, d= 0.28.
The tabu search undergoes many iteration and optimize these COCOMO II PA coefficient values. The resulting optimized COCOMO II PA coefficients by tabu search are as: a=2.51, b=0.62, c=3.78, d=0.01.
The efforts of first four projects are predicted and are compared with actual effort in table IV.

Table IV Predicted Values Of Effort Using Tabu Search

P.No	Size (KSLOC)	Actual effort (PM)	COCMO II predicted effort	Calculated Effort using coefficients optimized by Tabu search
1	003.0	001.20	003.60	1.33
2	019.90	074.60	092.70	82.0
3	004.05	002.0	002.30	1.7
4	004.25	004.50	009.30	4.8

It can be seen from the table that effort which is calculated by optimizing the coefficients of COCOMO II by using Tabu search technique provides the results which are much closer to actual effort and better than current COCOMO II PA predicted effort.

V. Conclusion

Accurate software cost estimation is an important factor in project planning. It has been seen that Tabu search aims to follows close to Glover's basic approach. It optimizes the predicted effort of COCOMO II as close to real effort. It helps in developing software within time and budget.
Hence the algorithm has effectively solved the complex optimization problem and achieves the more accurate results by optimizing the coefficients of COCOMO II.

References

[1] Barry Boehm, Chris Abts, Sunita Chulani, "Software development cost estimation approaches –A survey" Annals of Software Engineering 10, 2000, pp.182-190
[2] Bogdan Stępień, "Software Development Cost Estimation Methods and Research Trends" Computer Science, Vol. 5, 2003, pp. 68-82
[3] Nancy Merlo-schett, "seminar on software cost estimation" Requirement Engineering research group, Department of Computer Science, WS 2002/2003, pp. 3-19.
[4] Kavita Choudhary, " GA Based Optimization of Software Development Effort Estimation " International Journal Of Computer and Technology, Vol. 1, No. 1, Sep.2010, ISSN: 0976-8491, pp. 38-40.
[5] B.T. Eck. Good solutions to job shop scheduling problems via tabu search. Technical report, Department of Industrial Engineering and Operations Research, Columbia University, New York, May 1989.
[6] Steven J. Beaty, "Genetic Algorithm versus Tabu Search for Instruction Scheduling"pp.14-18.
[7] Caserta M, Uribe AM. Tabu search-based metaheuristic algorithm for software system reliability problems. Computers and Operation Research (2007), doi: 10.1016/j.cor.2007.10.028.
[8] A. Yahya, Rodina Ahmad, Sai Peck Lee, "Effects of Software Process Maturity on COCOMO II's Effort Estimation from CMMI Perspective" In 2008 IEEE, Department of Software Engineering, University of Malaya, pp.255-256.
[9] Ekananta Manalif, " Fuzzy Expert – COCOMO Risk Assessment and effort contingency Model in Software Project Management" The university of Western Ontario London, Ontario, Canada.
[10] Rachid Chelouah and Patrick Siarry " A hybrid method combining continuous tabu search and Nelder- Mead simplex algorithms for the optimization of multiminima functions" European Journal of Operation Research, 2005.