

# A Bayesian Network Model of the Particle Swarm Optimization for Software Effort Estimation

Germanjit Singh Sandhu  
Student, Department of CSE  
Lovely Professional University  
Phagwara, Punjab (India) – 144411

Dalwinder Singh Salaria  
Assistant professor, Department of CSE  
Lovely Professional University  
Phagwara, Punjab (India) – 144411

## ABSTRACT

Rapid growth of software industry leads to need of new technologies. Software effort estimation is one of the areas that need more attention. Exact estimation is always a challenging task. Effort Estimation techniques are broadly classified into algorithmic and non-algorithmic techniques. An algorithmic model provides a mathematical equation for estimation which is based upon the analysis of data gathered from previously developed projects and Non-algorithmic techniques are based on new approaches, such as Soft Computing Techniques. Effective handling of cost is a basic need for any Software Organization. This paper presents the new hybrid Bayesian Network model of PSO for effort estimation. we have developed a tool in MATLAB and at last proved that Bayesian Network with PSO gives more accurate results than other existing techniques. For sake of ease, we use NASA 93 datasets to verify the model and also compare the proposed model with COCOMO and Bayesian Regulation Neural Network Model and it is found that the developed model provides better estimation.

## General Terms

Particle Swarm Optimization, Neural network, Bayesian Network.

## Keywords

COCOMO, PSO, Bayesian Network, Effort Estimation

## 1. INTRODUCTION

Software effort estimate is one of the noticeable & mind catching field. But since it was started, it is challenging factor for software industry and Academia to realize the exact estimation of software development. In today's fast changing world, success in managing projects is a critical factor for the success of the entire organization. Estimation that either overestimated or underestimated both is very critical. In case of Overestimating time and effort (or budget), due to a presumed lack of resources or because the projected completion is too late, can convince management not to approve projects that may otherwise contribute to the organization. On the other hand, underestimation may result in approval of projects that will fail to deliver the expected product within the time and budget available. There are many factors that influence the Software estimation, some of them are: uncertainty, level of detail of preparing the project plan, managerial factors, lack of past data, pressure to lower estimation and estimator experience [11]. In spite of the critical role of accuracy, examples of incorrect estimation abound, especially in IT projects, resulting in enormous waste of time and money. Some techniques which were used in the past are not in use during present time, like SLIM [3], checkpoint [2], Seer [1]. In all the way of work time, many of new advance roads have been suggested for effort estimation

like Genetic programming [17], Fuzzy logic [12], Neural Network [8], data mining [5], etc. One cannot state that one model give better accuracy above all. Each and every give different level of accuracy in different Environment. But in recent days, soft computing techniques like PSO [7] gains main attention. The main focus of this paper is to investigate the accuracy of estimation using Bayesian network approach based on PSO and this has been done with the help of tool generated by us in MATLAB. This paper comprises as follow: section 2 describes the some former effort estimation models and review of related work to PSO, section 3 includes introduction of Bayesian Network and PSO, in section 4 problem is stated, section 5 describes methodology used, section 6 includes experimental results and comparisons. In last conclusion and future scope is given.

## 2. REVIEW OF LITERATURE

The period of Effort Estimation was started from the expert judgments, which is based on the experiences of experts. But it is only proceed as pillar when current project & pertinent Past projects are similar. Choices of effort estimation techniques footstep from COCOMO [2] to AI approaches [3]. In 1970, Larry Putnam developed the method SLIM [1], based on the Rayleigh function and the influence used to Rayleigh curve was Manpower Buildup Index (MBI and Productivity Factor (PF) [2]. Linear programming was key work to drive effort estimation in SLIM [3] and depend upon the source line of code. In 1981 developer Barry Boehm developed COCOMO as constructive cost model [21]. Which is one of an easy going & understandable model, could call the effort & time period of project. Due to some problems and some misses found in COCOMO, later on Barry Bohem developed the advance road of this model i.e. COCOMO 2.0 [3]. As growth of software industry rising tremendously and previous version was not up to need. After that, Howard Rubin proposed the ESTIMAC model to estimate effort at conception stage [21]. Equations used in this model are not available, because it was a proprietary model. ESTIMAC is high level model but doesn't provide accurate solution [10]. Six critical estimation dimensions identified by Rubin for this model are: effort hour, staff size, cost, hardware resource requirement, risk, portfolio impact [2]. But these methods (COCOMO, SLIM, ESTIMAC) are based on Line of code (LOC). The main problems in Line of Code methods are: lack of universally accepted definition for exactly what line code really is? Other side line of code is language dependence. So, in 1979 at IBM, developer Allan Albrecht developed measurement method called Function point [10] in order to reduce the issues related with LOC methods. Function point defines the complexity of software system in terms of functions that system delivers to user. It comprise linear combination of five basic software components (input, output, master files, interfaces, inquiries) consider to be low, average,

high [10]. In 1990, GC Low and DR. Jeffery also concluded in their paper that Function point method is more consistent than the line of code measure [9]. But on the other side, function point method is unable to deal with Uncertain,

imprecise and incomplete data. Many researcher collaborate PSO with different types of techniques for effort estimation. Some of PSO collaborate models are given below:

**Table 1. Summary of PSO Effort Estimation Approaches with Their Pros and Cons**

Author's Name	Year	MODEL name	dataset	Comparison With	Conference/ Journal	advantages	disadvantages
Alaa Sheta et al.,	2008	COCOMO- PSO	COCOMO 81	Fuzzy Logic and Halstead	IEEE	accuracy	Don't handle missing data
Prasad Reddy	2010	PSO with Fuzzy	COCOMO 81	COCOMO -PSO and Fuzzy	IJSE	Uses linguistic Variable	Rules are not very direct
Prasad Reddy et al.,	2011	PSO with Inertia Weight	NASA	Not mention	IJSE	Better tuned	Difficult to compute
Srinivasa Rao T. et al.,	2013	PSO with K-Mean	COCOMO 81	COCOMO and Fuzzy	IJSE	faster	Fails for nonlinear data sets
Zhang Dan	2013	PSO-ANN-COCOMO II	COCOMO I and NASA93	ANN-COCOMO	IEEE	Powerful tool	complex
Rao et al.,	2014	MPSO	COCOMO 81	COCOMO	SPRINGER	Better results	Poor to handle missing datasets

#### Artificial Neural Network Model Based on PSO

Sheta, Alaa proposed a collaboration of artificial neural network (ANN) and Constructive Cost Model (COCOMO)[20] [3], which expanded by Particle Swarm Optimization (PSO). PSO-ANN-COCOMO II model accurately estimate the cost of Software development. This revised model not only raised the speed of artificial neural network but also resolve the problem of dependency of initial weight in learning ability of artificial neural network. With keeping the advantages of COCOMO model, this model get better the learning capability of original model. PSO-ANN-COCOMO II has an progress of 3.27% in software effort estimation precision than the original artificial neural network Constructive Cost Model (ANN-COCOMO II), this

consequence is supported by two data set ( COCOMO I and NASA93).There are three critical factors in software development process: Time, Cost and Manpower. By appraising these parameters at early phase of project, we can get an effective development process. As original ANN-COCOMO II model provide good solution to vagueness of data and uncertainty found in software attributes, but every model have some cons also, low speed convergence and high dependency on network for initial weight are some short-coming in ANN-COCOMO II model. Particle swarm optimization algorithm solves these problems.

#### PSO Fuzzy Software Cost Estimation Models

In 1969, PVGD et.al is presented Particle Swarm Optimization Algorithm (PSOA) to fine tune the fuzzy estimate for the development of software projects [13][6][18].

Efficiency is tested by using three datasets: 10 NASA software projects, 18 NASA projects and COCOMO 81 database, which was based on a range of criteria for assessment of software cost estimation. By comparison of all the models, discussed in this paper, it proved that model developed by using particle swarm optimization algorithm is better than all.

#### The PSO-tuned COCOMO model

Bohem developed the simple COCOMO model in 1981 [1]. For the better software effort estimation Sheta et.al used the PSO model along with simple COCOMO model in 2008. Generalization of computation is achieved for all the projects by using PSO model [22].

#### PSO with Inertia Weight model

In 2010 PVGD et.al used PSO with initial weights in order to tune the parameters. For the continue moving of particles in the same direction a nonzero inertia weight is introduced. Decreasing the inertia over time introduces a shift from the exploratory (global search) to the exploitative (local search) mode. [14]

#### Multi Objective Particle Swarm Optimization for software Cost Estimation

Recently 2014 Rao ,et al. proposed a model for software cost estimation using Multi Objective (MO) Particle Swarm Optimization. By considering two objectives- Mean Absolute Relative Error and Prediction, parameters of model tuned by using MOPSO [15]. For testing COCOMO dataset is considered. From the comparison, it proved that model developed using MO gives better results than standard COCOMO. Also found that by giving more classification among training data gives efficient results. At the deficiency part of this model, it returns set of solution rather than returning a single solution.

#### The PSO with K-mean Model

In 2013, Rao et al. proposed Particle Swarm Optimization technique is proposed which drive on data sets which are clustered using the K-means clustering algorithm [16]. PSO has been employed to generate parameters of the COCOMO model for each cluster of data values. Back Propagation technique is used to train the Neural Network. COCOMO 81 dataset is used for testing and also the results have been compared with standard COCOMO model and as well as the neuro fuzzy model. It is concluded from the results that the neural networks with efficient tuning of parameters by PSO operating on clusters, can generate better results and hence it can function efficiently on ever larger data sets.

### **3. INTRODUCTION TO BAYESIAN NETWORK AND PSO**

BN are a strong modeling technique which increases some different unique characteristics. A powerful feature of BN is the possibility, through application of probability theory, to model uncertainty or subjectivity. that allows the integration

of objective evaluations, learned from data, with subjective evaluations estimated by experts [19]. Also another feature is the possibility to carry out what-if analyses, by giving the model with variations in input values, referred to as evidences, and monitoring the effect on the output. The main advantage of BN is that it can deal well with missing data, which happens number of times in all software projects datasets. BNs are named after and founded on the Bayes theorem, which models the relationship between two variables. The Bayes theorem basically builds the relationship between the prior and posterior probabilities. The prior probability taking some specified value regardless of other variables, that is, it is the default probability. Posterior probability is variable taking some value, given some evidence, i.e. it is the updated probability after new information is input. BNs is basically the graph structure that models the causal relationships between the variables, and a quantitative part made up of node probability tables (NPT's) which contain the probability distributions. The directed acyclic graph (DAG) encodes the dependencies between the variables. The nodes represent the relevant variables (factors) in the domain being modeled, and each directed arc depicts the dependencies between these factors [19]. The NPT's contain the prior and conditional probabilities. BNs can be modeled fully based on data, through a hybrid approach, i.e. integrating data modeling and experts knowledge or fully expert-based. Expert Knowledge can be coded by means of subjective or qualitative variables and also in the network topology by defining the relationships between the variables.

PSO is a population related search technique. In the population, there are different set of particles which signifies the solution for problem. These particles are generally initialized randomly in most evolutionary computation methods. At the time of starting process, each particle based on some evaluation, changes their personal position with positive speed [7]. The speed is calculated based on experience of the particle itself and all of the population. This changed process is repetitive for different generations. This process stops either when the task is completed or when the maximum number of generations is reached.

PSO was developed in 1995 by James Kennedy and Russell Eberhart. It utilizes a number of particles that comprise a swarm moving around for the best results [7]. Every particle is treated as a point in a N-dimensional space which alters its "flying" according to its own flying experience and the flying experience of other particles also. PSO is a population-based optimization tool, which could be imposed and applied easily to solve different function optimization problems. PSO is a robust stochastic optimization technique based on the movement of intelligent swarms. The basic concept of PSO lies in accelerating every particle towards its Pbest and Gbest locations with a random weighted acceleration every time. PSO is one of the optimization methods and a kind of evolutionary computation technique.

**Table 2. Cost Drivers for COCOMO**

Attribute	Type	Description
RELY	Product	Required system reliability
CPLX	Product	Complexity of system modules
DOCU	Product	Extent of documentation required
DATA	Product	Size of database used
RUSE	Product	Required percentage of reusable components
TIME	Computer	Execution time constraint
PVOL	Computer	Volatility of development platform
STOR	Computer	Memory constraints
ACAP	Personnel	Capability of project analysts
PCON	Personnel	Personnel continuity
PCAP	Personnel	Programmer capability
PEXP	Personnel	Programmer experience in project domain
AEXP	Personnel	Analyst experience in project domain
LTEX	Personnel	Language and tool experience
TOOL	Project	Use of software tools
SCED	Project	Development schedule compression
SITE	Project	Extent of multisite working and quality of inter-site communications

#### 4. PROBLEM STATEMENT

The main aim of any software development organizations is to finish the project within acceptable or customary schedule and budget. Budget is mainly driven by labor cost and time and together they form a measure called effort. From quality point of view estimating effort is one of the major important factors. Because estimation either it be over estimate or under estimate, produces worst results. In case of over estimation of time and effort project completion is too late due to lack of resources, which refuses the management to approve that favored project. On the other hand, under estimation may result in approval of projects that will fail to deliver the expected product within the time and budget available [11]. So, there is a need of accurate estimation effort technique at early Stages of software development. In this research, the main aim is to improve software effort estimation by using Bayesian network with PSO. The main reason for using such a learning system for this problem is to keep the estimation process up-to-date by incorporating up-to-date project data. At last Comparison is drawn between training algorithms used in

this research to state that Bayesian Network with PSO gives much accurate estimation. One algorithmic approach, COCOMO is also compared with Bayesian Regulation of neural network model.

#### 5. PROPOSED METHODOLOGY

Following are the steps used for Effort Estimation:

##### 5.1 Data Collection

The dataset used in this work is NASA93 (<http://promisedata.googlecode.com>) a public available data set consisting of a total of 93 projects at the time of this study.

##### 5.2 Division of Data

Data set is divided into two parts: Training and Testing. For our work we divide the data into 85-15% ratio i.e. 80 rows for training and 13 for testing. These 13 rows are randomly chosen by formula ( $\text{ceil}(1 + (93-1) * \text{rand}(13,1))$ ), available in MATLAB. From this, for testing row number 15,40,92,74,91,94,5,80,59,64,71,63, 38 are chosen.

##### 5.3 Cost Drivers

Cost drivers for this work choose from the cost drivers designed for COCOMO II. Table 2 represents, Cost drivers for COCOMO.

##### 5.4 Normalization of data

The first step in training to normalize the data , because in Bayesian network the data is only pass to the network when it is in normalize .so the formula for normalization is

$$I1 = (I1 - \min(I1)) / (\max(I1) - \min(I1)) - 1;$$

Where I1 is the data which are input by radio button.

##### 5.5 Initializing parameters to PSO algorithm

After the normalization of data, the next step is to initializing the parameters to PSO i.e., initialize the swarm, velocity and positions of the particles

##### 5.6 Creation of Bayesian network

After the second step then the creation of Bayes net is started. In this step the Bayes net is created and also the Dags are created.

## 5.7 Training

Once the network weights and biases are initialized, the network is ready for training.

## 5.8 Performance Criteria

Mean Magnitude Relative Error: MMRE is frequently used to evaluate the performance of any estimation technique. It seems obvious that the purpose of MMRE is to assist us to select the best estimation approach. It measures the percentage of the absolute values of the relative errors, averaged over the N items in the "Test" set and can be written as [4]:

$$MRE = \frac{|\text{actual effort} - \text{estimated effort}|}{\text{actual effort}}$$

$$MMRE = \frac{1}{N} (\sum MRE_i)$$

$$MMRE \text{ of COCOMO} = \frac{6.92}{13}$$

$$= 0.51$$

$$MMRE \text{ of BR} = \frac{6.446}{13}$$

$$= 0.49$$

$$MMRE \text{ of Bayesian Network with PSO} = \frac{0.85}{13}$$

$$= 0.065$$

## 6. EXPERIMENTAL RESULTS AND COMPARISON

Table 4 summarizes the result obtained by COCOMO model Bayesian Regulation Neural Network Model and Bayesian Network Model of PSO. In the testing phase the calculated efforts and errors using different training algorithms and COCOMO is shown in table 3 and table 4 respectively.

**Table 3. Comparison between different training algorithms**

Performance criteria	COCOMO	BR	Bayesian with PSO
MMRE	0.51	0.49	0.065

Figure 1 clearly present Bayesian Network with PSO is more accurate than others. As evident from the table 3, the predicted values of the Bayesian Network with PSO efforts is very close to the expected or actual values as compare to Bayesian Regulation Neural Network and COCOMO.

## 7. CONCLUSION

Effort Estimation is one of the crucial tasks in software project management. This simulation with NASA93 dataset has been carried out using tool created with the help of MATLAB. Bayesian Network is generated using PSO. The result from our simulation shows that Bayesian Network Model of PSO gives the best performance, among the other training algorithms. We have experimented with 15 attributes of the COCOMO and further investigation can be done with other attributes and also concentration needed for process maturity.

**Table 4. Effort Estimation by using COCOMO, BR and Bayesian Network with PSO**

Row no.	Expected	COCOMO	BR	Bayesian Network with PSO
15	48	85.9557	61.9294	54
40	114	66.9477	121.206	126
92	240	85.9557	85.847	232
74	4178.2	1649.24	4058.46	4150
91	1772.5	539.26	2902.12	1794
94	1924.5	393.61	1201.62	1921
5	25.2	38.2213	83.0016	32.7
80	703	904.279	562.929	693
59	4560	6718.84	4471.23	4584
64	150	115.445	61.3017	157
71	72	155.732	106.606	79
63	160	270.499	61.7749	144
38	444	463.311	338.233	456

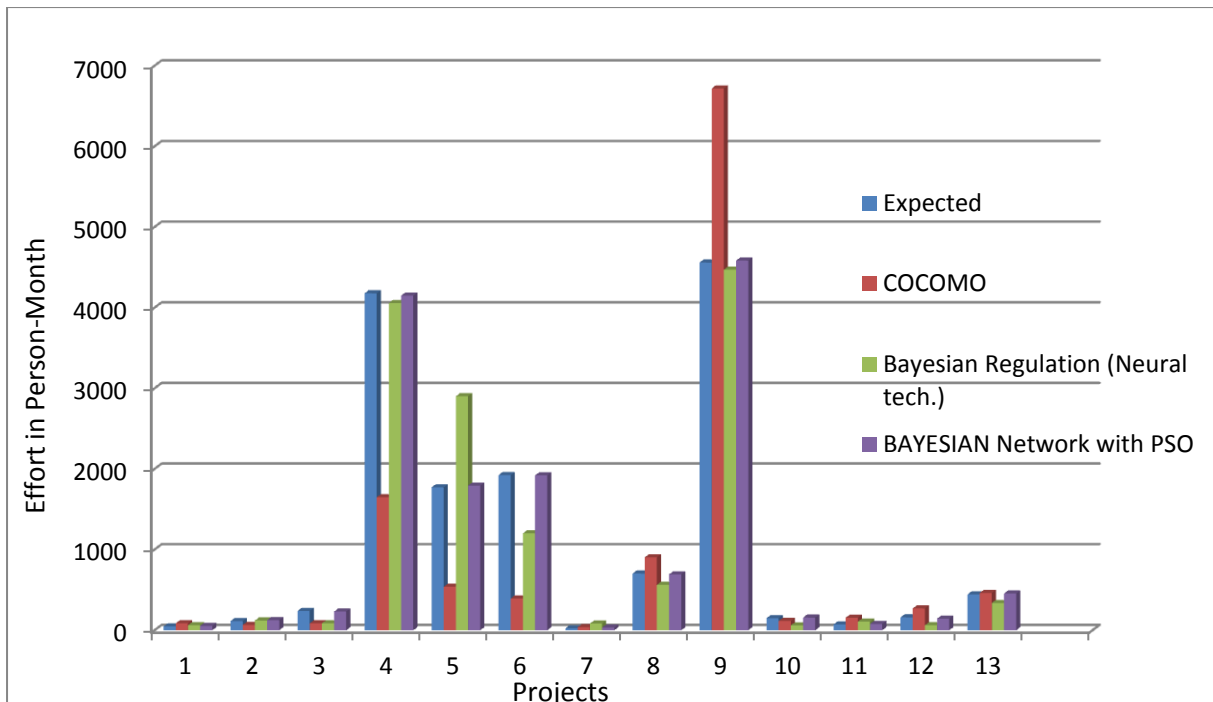


Fig 1: Column chart for effort estimation

## 8. REFERENCES

- [1] Boehm, B.W. Software Engineering Economics. rentice-Hall, Englewood Cliffs, N.J. 1981.
- [2] Boehm, Barry, Chris Abts, and Sunita Chulani. Software development cost estimation approaches—A survey." Annals of Software Engineering 10.1-4 (2000): 177-205.
- [3] Boehm, Barry, et al. "Cost models for future software life cycle processes: COCOMO 2.0." Annals of software engineering 1.1 (1995): 57-94.
- [4] de Barcelos Tronto, Iris Fabiana, Jose Demisio Simoes da Silva, and Nilson Sant'Anna. "Comparison of artificial neural network and regression models in software effort estimation." Neural Networks, 2007. IJCNN 2007. International Joint Conference on. IEEE, 2007.
- [5] Dejaeger, Karel, et al. "Data mining techniques for software effort estimation: A comparative study." Software Engineering, IEEE Transactions on 38.2 (2012): 375-397.
- [6] E. C. Laskari, K. E. Parsopoulos and M.N. Vrahatis, Particle Swarm Optimization for Minimax Problems , Evolutionary Computation, In: (Eds.) CEC '02 Proceedings of the 2002 Congress On, 2, 2002, pp. 1576 -158.
- [7] Eberhart, Russ C., and James Kennedy. "A new optimizer using particle swarm theory." Proceedings of the sixth international symposium on micro machine and human science. Vol. 1. 1995.
- [8] Haykin, Simon. Neural networks: a comprehensive foundation. Prentice Hall PTR, 1994.Kemerer, Chris F. "An empirical validation of software cost estimation models." Communications of the ACM 30.5 (1987): 416-429.
- [9] Low, Graham C., and D. Ross Jeffery. "Function points in the estimation and evaluation of the software process." Software Engineering, IEEE Transactions on 16.1 (1990): 64-71.
- [10] Matson, Jack E., Bruce E. Barrett, and Joseph M. Mellichamp. "Software development cost estimation using function points." Software Engineering, IEEE Transactions on 20.4 (1994): 275-287.
- [11] Morgenshtern, Ofer, Tzvi Raz, and Dov Dvir. "Factors affecting duration and effort estimation errors in software development projects." Information and Software Technology 49.8 (2007): 827-837.
- [12] Nisar, M. Wasif, Yong-Ji Wang, and Manzoor Elahi. "Software development effort estimation using fuzzy logic-A survey." Fuzzy Systems and Knowledge Discovery, 2008. FSKD'08. Fifth International Conference on. Vol. 1. IEEE, 2008.
- [13] PVGD, Prasad Reddy, and CH VMK Hari. "Software Effort Estimation Using Particle Swarm Optimization with Inertia Weight." Global Journal of Computer Science and Technology 11.18 (1969).
- [14] PVGD, Prasad Reddy. "Particle swarm optimization in the fine-tuning of fuzzy software cost estimation models." International Journal of Software Engineering (IJSE) 1.2 (2010): 12-23.
- [15] Rao, G. Sivanageswara, Ch V. Phani Krishna, and K. Rajasekhara Rao. "Multi Objective Particle Swarm Optimization for Software Cost Estimation." ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India-Vol I. Springer International Publishing, 2014.
- [16] Rao, Srinivasa, C. H. Hari, and Prasad Reddy PVGD. "Predictive and Stochastic Approach for Software Effort Estimation." Int. J. of Software Engineering, IJSE 6.1 (2013).

- [17] Shan, Yin, et al. "Software project effort estimation using genetic programming." *Communications, Circuits and Systems and West Sino Expositions, IEEE 2002 International Conference on*. Vol. 2. IEEE, 2002.
- [18] Suresh Chandra Satapathy, J.V.R. Murthy, P.V.G.D. Prasad Reddy, B.B. Misra, P.K. Dash and G. Panda, Particle swarm optimized multiple regression linear model for data classification *Applied Soft Computing* , 9, ( 2), (2009), Pages 470-476.
- [19] Tierno, Ivan AP, and Daltro J. Nunes. "Assessment of Automatically Built Bayesian Networks in Software Effort Prediction." *CibSE*. 2012.
- [20] Zhang. "Improving the accuracy in software effort estimation: Using artificial neural network model based on particle swarm optimization." *Service Operations and Logistics, and Informatics (SOLI), 2013 IEEE International Conference on*. IEEE, 2013.
- [21] Kemerer, Chris F. "An empirical validation of software cost estimation models." *Communications of the ACM* 30, no. 5 (1987): 416-429. [22] Sheta, Alaa, David Rine, and Aladdin Ayeshe. "Development of software effort and schedule estimation models using soft computing techniques." *Evolutionary Computation*, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on. IEEE, 2008.