Search-Based Approaches for Software Development Effort Estimation

Federica Sarro
University of Salerno
Via Ponte don Melillo
8404 Fisciano (SA), Italy
+39089963323
fsarro@unisa.it

ABSTRACT

In the last years the use of Search-Based techniques has been suggested to estimate software development effort. These techniques are meta-heuristics able to find optimal or near optimal solutions to problems characterized by large space. In the context of effort estimation Search-Based approaches can be exploited to build estimation models or to enhance the effectiveness of other methods. The preliminary investigations carried out so far have provided promising results. Nevertheless, the capabilities of these approaches have not been fully explored and the empirical analyses carried out so far have not considered the more recent recommendations on how to perform this kind of empirical assessment in the effort estimation context and in Search-Based Software Engineering. The main aim of the PhD dissertation is to provide an insight on the use of Search-Based techniques for effort estimation trying to highlight strengths and weaknesses.

Categories and Subject Descriptors

D.2.8 [**Metrics**], D.2.9 [**Management**], G.1.6 [**Optimization**]

General Terms

Algorithms, Management, Measurement, Experimentation.

Keywords

Software Development Effort Estimation, Search-Based Software Engineering, Empirical Study.

1. INTRODUCTION

Effort estimation is a critical activity for planning and monitoring software project development and for delivering the product on time and within budget. Indeed, significant over or under-estimates expose a software project to several risks. As a matter of fact under-estimates could lead to addition of manpower to a late software project, making the project later (Brooks's Law), or to the cancellation of activities, such as documentation and testing, negatively impacting on software quality and maintainability. Thus, the competitiveness of a software company heavily depends on the ability of its project managers to accurately predict in advance the effort required to develop software system. However, several challenges exists in making accurate estimates, e.g., the estimation is needed early in the software lifecycle, when few information about the project are available, or several factors can impact on project effort and these factor are usually specific for different production contexts.

Several techniques have been proposed in the literature to support project manager in estimating software project development effort.

To date, expert opinion is a commonly used estimation method and is still used by software and Web companies [22]. However, relying on the expertise of the company's practitioners the results are less repeatable, being mainly based on subjective judgments [5]. Moreover, this made difficult to quantify and to determine those attributes that have been used to derive an estimate [30].

To overcome this limitation, several techniques which rely on a more formal approach have been proposed. These include the application of some algorithms to a number of factors that influence the development cost, such as the size, to produce an estimate or a model providing the estimation in an objective way. COCOMO and COCOMO II are probably the best known generic methods [5]. They are based on a regression formula, with parameters that are derived from some historical project data and current project characteristics. They are generic methods that often need to be calibrated to local data to take into account the characteristics of the specific production context. Alternatively, a software company can construct its specific model (or estimation) using an estimation technique that takes as input the information coming from past projects. Usually the employed data consist of information about some relevant factors (named cost drivers) and the effort actually spent by the company to develop prior projects. In this class of data-driven estimation techniques, we can find Linear and Stepwise Regression [5] [30] and some artificial intelligence techniques, such as Classification and Regression Tree (CART), Case-Based Reasoning (CBR), and Bayesian Networks (BN) [30].

In the last years the use of Search-Based (SB) approaches has been suggested to be employed as an effort estimation technique. These approaches include a variety of meta-heuristics, such as local search techniques (e.g., Hill Climbing, Tabu Search, Simulated Annealing) or Evolutionary Algorithms (e.g., Genetic Algorithms, Genetic Programming). They search for suitable solutions to problems characterized by large search space, using an objective function that gives an indication of how a solution is suitable for the problem under investigation.

The generic nature of these meta-heuristics let them to be fruitful for different goals and issues, simply by redefining the solution

Conference PROFES 2011, June 20-22, 2011, Torre Canne, BR, Italy.

representation and the objective function. As a matter of fact, in the last years there has been an explosion of researches on the use of SB techniques in many software engineering fields [20], giving rise to a very active field known as Search-Based Software Engineering (SBSE) [19]. The idea underlying the use of such techniques is based on the reformulation of software engineering problems as search or optimization problems whose goal is to find the most appropriate solutions which conform to some adequacy criteria (i.e., problem goals). In particular, the use of SB approaches in the context of effort estimation is twofold: they can be exploited to build effort estimation models or to enhance the use of existing effort estimation techniques. In the first case the problem of building estimation model is reformulated as an optimization problem where the SB method builds many possible models - exploiting past projects data - and tries to identify the best one, i.e., the one providing the most accurate estimates. In the second case, SB methods can be exploited in combination with other estimation techniques to improve critical step of their application (e.g., features subset selection or the identification of critical parameters) aiming to obtain better estimates.

The usage reported in the literature of SB approaches for effort estimation have provided promising results that encourage further investigations. However, they can be considered preliminary studies. As a matter of fact, the capabilities of these approaches were not fully exploited, either the employed empirical analyses did not consider the more recent recommendations on how to carry out this kind of empirical assessment in the effort estimation and in the SBSE contexts [1] [2] [26] [37]. The main aim of the PhD dissertation is to provide an insight on the use of SB techniques for the effort estimation trying to highlight strengths and weaknesses of these approaches for both the uses above mentioned.

In the following we provide a brief literature review, the goals of our research and the methodology employed to address them, and a brief report of the initial results.

2. BRIEF LITERATURE REVIEW

Some investigations have been carried out so far on the use of SB approaches for effort estimation. These studies have provided promising results that encourage further investigations. However, they can be considered preliminary studies. As a matter of fact, the capabilities of SB approaches have not been fully exploited and often the empirical analyses have not taken into account the more recent recommendations on how to carry out this kind of empirical assessment in the effort estimation and in the SBSE contexts [1] [2] [26] [37], as detailed in the follow.

Table 1 summarizes the main aspects (e.g., employed technique, dataset, validation method, and evaluation criteria) of the studies carried out so far to assess SB approaches for building effort estimation models. First of all we observe that all the previous studies [6] [11] [28] [34] employed Genetic Programming (GP) and no attempts have been reported on the use of other SB techniques (e.g., the ones based on local-search), although they have many similarities but also distinguishing features.

Moreover, each SB technique has specific design choices that may affect the performance of the method. As an example, for GP we have to choose the solution encoding, the fitness function (i.e., objective function), the strategy for creating the initial population, the operators for mating and survival selection, the crossover and mutation operators, and the stopping criteria. The choice of the objective function is common to all the SB techniques and represents one of the most critical step since such function guides the search towards suitable solutions. In the context of effort estimation this choice should be based on a measure of model accuracy. The studies carried out so far exploited two measures as fitness function, namely MMRE [6] [28] and MSE [11] [34]. However, several measures have been proposed to evaluate effort estimation accuracy and all of them could be exploited as objective function [18]. Nevertheless, the use of multiple criteria has not been investigated although there are recommendations on the use of several different accuracy measures to carry out a more reliable evaluation of estimation models. The existing studies have neither fully investigated the impact of the other design choices such as the stopping criterion and its impact on the method convergence.

Concerning the empirical analyses, all the studies employed only one dataset thus affecting their external validity. Moreover, a hold-out validation was applied, where the dataset is split into a training set used to build the estimation model and a test used to validate it. Unfortunately this procedure can be biased since the prediction performance may depend on how the dataset is split.

Regarding the evaluation criteria only summary measures were employed: in particular MMRE and Pred(25) in all the case studies, and in some cases also MSE, AMSE, BMMRE, and Pred(50).

As for the benchmarks, useful to understand the actual effectiveness of the proposed approach, all the case studies employed several estimation methods, such as Linear Regression (LR) and Case-Base Reasoning (CBR). However, often there is a lack of details about their application. As for example, studies that employed LR did not state if the underlying assumptions were verified [24] while this aspect is crucial for the internal validity of the empirical study.

Finally, little attention has been given by previous studies to the random variation in results due to the non-deterministic nature of SB techniques: indeed, very few executions were performed and often only results related to the best execution were reported, thus affecting the conclusion validity of these case studies.

Many of the above limitations can be found also in the studies that assessed the use of SB approaches to improve existing effort estimation techniques [4] [7] [21] [27] [29] [36]. As an example, as we can observe from Table 2, all the studies exploited Genetic Algorithms. For the sake of space we refer the reader to [12] where a more detailed description of prior work is provided.

3. RESEARCH GOALS AND METHODOLOGY

On the basis of the weakness highlighted in the state of the art, the research will focus on the following research goals:

RG1. How the design choices characterizing the use of SB approaches impact on the performance of these techniques?

RG2. Can the use of multi-objective approaches improve the effectiveness of SB methods?

RG3. Are there any differences in the use of different SB techniques?

RG4. Are SB techniques more effective than widely used effort estimation methods?

Table 1. Summary of the empirical studies that assessed SB approaches for building effort estimation models

Reference	Employed technique	Case study Dataset	Validation method	Evaluation Criteria	Benchmark Methods
[6]	GP with MMRE as fitness function	Desharnais	hold-out training set: 149 test set: 15	AMSE, MMRE, BMMRE, Pred(25)	ANN, LR, CBR
[11]	GP with MSE as fitness function	Academic projects	hold-out training set: 30 test set: 16	MMRE, Pred(25)	LR, ANN
[28]	GP with MMRE as fitness function	Finnish	hold-out training set: 63 test set: 18	AMSE, MMRE BMMRE, Pred(25)	ANN, LR, CBR
[34]	GP with MSE as fitness function	ISBSG	hold-out training test: 211 test set: 212	MMRE, Pred(25) Pred(50), MSE	LR

Table 2. Summary of the empirical studies that assessed the use of SB approaches in combination with existing estimation methods

Reference	Employed technique	Case study Dataset	Validation Method	Evaluation Criteria	Benchmark methods
[4]	GA+SVR with MMRE and Pred(25) as fitness function	Desharnais and NASA	leave-one-out	MMRE, Pred(25)	SVR
[7]	GA+CBR with MMRE and Pred(25) as fitness function	Canadian Financial service and IBM DP	3-fold	MMRE, MdMRE, Pred(25)	OLSR, ANN, CART
[29]	GA+CBR with MMRE as fitness function	Desharnais, Albrecht, and two artificial datasets	hold-out	MMRE, MdMRE, Pred(25)	CBR, SVR, ANN, CART
[27]	GA+CBR with MMRE as fitness function	Albrecht, COCOMO, and ER	leave-one-out	MMRE, MdMRE, Pred(25)	COCOMO, NN, LR, GRA, CBR, CART
[21]	GA+GRA with MMRE as fitness function	Albrecht and COCOMO	3-fold	MMRE, Pred(25)	CBR, ANN, CART
[36]	GA+NN with MSE as fitness function	78 software projects	hold-out (n times) training sets: 63 test sets: 15	Student's t-test	Regression Tree NN. Back-Propagation NN, Quick Propagation NN

RG5. Are SB techniques effective to improve the accuracy of other data-driven effort estimation techniques?

To address research goal *RG1*, special attention will be given to the role played by the use of different objective functions since this is the most important design choice to be made in the use of any SB technique. In particular, we plan to experiment several objective functions based on both single and combined evaluation measures and to assess how the accuracy of GP is affected by this choice. The combination of accuracy measures will allow us also to experiment simple forms of multi-objective SB approaches. Other more sophisticated methods, such as the ones based on Pareto optimality, will be also investigated to address research goal *RG2*.

To address *RG3* we will analyze the use of different SB techniques and compare them in terms of accuracy and cost-effectiveness.

Concerning *RG4*, to understand the actual effectiveness of SB effort estimation approaches, we will compare them with both baseline methods such the mean and median of effort and several widely used effort estimation techniques, such as Manual Stepwise Regression (MSWR) [30] and Case-Based Reasoning (CBR) [35]. Indeed, if the investigated estimation method does not outperform the results achieved with these baseline methods it cannot be transferred to industry [30]. Moreover, it will be also interesting to compare SB effort estimation approaches against the estimates provided by human judgment [22]. Nevertheless, this requires the availability of datasets containing both project data and human estimation data.

As for *RG5*, we plan to use SB techniques to configure other estimation techniques, such as Support Vector Regression (SVR) that is a new generation of Machine Learning algorithms that have turned out to be effective for development effort estimation. Nevertheless, its prediction accuracy is heavily influenced by its parameter setting [9] and no general guidelines are available to select these parameters. Thus, we will investigate the use of SB techniques in combination with SVR to select the parameters of SVR to be employed for effort estimation.

The research will be organized to verify the effectiveness of proposed techniques in a quantitative and reproducible way carrying out several empirical studies carefully taking into account the biases that might affect the obtained results (i.e., threats to validity). To this end we will perform empirical research following the guidelines proposed in [1] [26] [37]. In the following we discuss the main aspects of our empirical study design.

Techniques specification

In order to allow for replication and comparisons with other techniques and future studies, the design choices made to tailor SB techniques will be always described and justified. Also the application of other estimation methods will be described in details, clearly reporting all the assumptions underlying the employed techniques.

Dataset Selection

Collecting appropriate datasets for empirical experimentation is a crucial task, since whether the results can be generalized depends on whether the data under investigation are representative. To this end we will employ datasets of different size and characteristics, containing data about industrial software projects. In particular, the datasets contained in the PROMISE repository [33] should be adequate to our scope. Indeed, they contain data about industrial software projects developed in different languages and for many different application domains, ranging from telecommunications to commercial information systems. The number of observations ranges from tens to hundreds, the features were mostly based on projects characteristics available at prediction time (e.g. team experience and size, employed tools, employed languages), and almost all datasets contained Function Point as size measure. We also plan to employ datasets containing data about development of Web projects, such as the Tukutuku dataset [32], since such data could exhibit different characteristics with respect to the development of desktop applications. As an example, the size measure can be the number of features/functions instead of Functions Points. It is worth noting that the above datasets contains both single- and cross-company data and have been previously employed in other research works to evaluate effort estimation methods.

Validation Method and Evaluation Criteria

To verify whether or not a method gives useful estimations of the actual development effort we will perform a multiple-fold cross validation, partitioning the whole dataset into training sets, for model building, and test sets, for model evaluation [5]. We will made publicly available the folds employed in the validation process when public datasets are used to allow for replications of our studies.

To provide a more reliable accuracy evaluation we will use several evaluation tools [17] [25] such as widely used summary measures (i.e., MMRE, MdMRE, Pred(25), MEMRE, and MdEMRE [8] [24]) and boxplot of absolute residuals [25]. Moreover, to establish if one of the prediction methods provides significantly better estimates than the others we also test the statistical significance of the absolute residuals [24] and to have also an indication of the practical/managerial significance of the results we verify the effect size [23].

Threats to validity

Several factors can bias the construct, internal, conclusion, and external validity of empirical studies.

The choice of the features and how they are collected represent crucial aspects for construct validity. We will try to mitigate such threats by employing publicly available and industrial datasets. Moreover, we will consider only information that would be available at the early stages of the software development process avoiding the use of Lines of Code (LOC) that could create a false impression as to the efficacy of the prediction method [35].

The internal validity could be affected by the selection of data with certain characteristics; to mitigate such threat we will perform a multiple-fold cross validation.

As for the conclusion validity, we will carefully apply the statistical tests, verifying all the required assumptions. Moreover, the randomness in results obtained with SB approaches will be taken into account executing several runs (at least 10) and considering average results and measures, such as standard deviation, that indicate the level of uncertainty associated with the results [1].

Finally, to assure the external validity we will exploit many datasets, both single and cross-company, containing data collected in industrial contexts. Indeed, taking into account a huge amount of data collected from different companies let us to be more confident in the generalization of the achieved results.

4. INITIAL RESULTS

During the first PhD year some investigations have been carried out and the research goals have been partially addressed.

Concerning the use of SB to build effort estimation models two techniques were defined and empirically assessed. In particular, we employed a global search technique, namely Genetic Programming (GP) previously used for effort estimation (see Section 2), and a local search technique, Tabu Search (TS), never employed before to find optimal estimation models.

Both the SB techniques relying on solutions represented by models described by an equation of this type:

$$Effort = c_1\ op_1\ f_1\ op_2\ ...\ op_{2n-2}\ c_n\ op_{2n-1}\ f_n\ op_{2n}\ C \qquad (1)$$

where f_i represents the values of the i^{th} project feature and c_i its coefficient, C represents a constant, while op_i represents the i^{th} mathematical operator of the model. Thus, the search space of GP (TS) is represented by all the possible equations that can be generated assigning the values for c_i, C, and op_i and provide positive values for *Effort*. It is worth noting that the encoding of such a solution differs for GP [16] and TS [15] as for the design choices which are specific to each technique, such as evolutionary process for GP or moves and tabu list definition for TS. On the other hand, the employed objective function and stopping criteria can be the same.

The above techniques were experimented in several empirical studies meant to address research goals *RG1*, RG3 and *RG4*. In the following we briefly report the obtained results.

Concerning *RG1* (i.e., analyzing how the design choices made in the use of SB approaches impact on the performance of these techniques) interesting results were obtained analyzing the impact of different fitness functions on the accuracy of the estimation models identified by GP. A preliminary analysis [16] was carried out to analyze GP with single and combined evaluation measures as fitness functions on a publicly available dataset (i.e., Desharnais [33]) by exploiting a 3-fold cross validation. The obtained results showed that the choice of fitness function significantly impacts on the effort estimation accuracy of models identified by GP. In particular, some fitness functions negatively affect the overall estimation accuracy, while others - mainly the ones based on combinations of evaluation measures - behave significantly better. In a subsequent study (still not published) we replicated the analysis using more datasets and fitness functions and confirmed our preliminary findings. The analysis of the impact of other design choices in the definition of SB approaches and the investigation of more sophisticated multi-objective optimization approaches (*RG2*) are part of our agenda of future work.

Regarding *RG3* (i.e., investigating whether there are differences in the use of different SB techniques) and *RG4* (i.e., investigating if SB techniques are more effective than widely used effort estimation methods) the empirical studies performed exploiting GP and TS on single and cross-company datasets with hold-out and 3-fold cross validation, showed that both techniques were effective as MSWR and CBR for single-company datasets (i.e., Desharnais [33] and NASA [3]) [13] [14] [16], while TS provided superior results on a cross-company dataset of Web applications (i.e., Tukutuku [32]) [15]. These studies can be seen as a starting point for further investigations to be carried out possibly with other data. First of all, it could be interesting to compare TS and GP with other SB techniques (e.g., Simulated Annealing). Moreover, it could be also interesting to investigate the conditions (such as type of Web applications and type of methodology employed) and/or the characteristics of the dataset so that an estimation technique can provide better results than another.

As for the use of SB to enhance existing effort estimation techniques, the combination of Tabu Search with Support Vector Regression (SVR) was proposed. In particular, since the performance of SVR for effort estimation is influenced by its setting [9], we exploited TS to search for optimal configurations. In this case, the solution consists in SVR configuration and the search space is composed by several possible settings. A preliminary empirical study [10] was carried out on the Tukutuku dataset by exploiting multiple hold-out validation addressing *RG5*. The obtained results highlighted that TS can be effectively used to configure SVR, allowing us to achieve the best performance ever obtained on this dataset up to now. An extension of this work (still not published), devoted to generalize the obtained results using more data, highlighted that the proposed approach is very effective allowing us to obtain significantly superior estimates respect to MSWR and CBR.

5. ACKNOWLEDGMENTS

I am deeply grateful to my supervisor Prof. Filomena Ferrucci for her continuous and invaluable advice, encouragement, and support. I also thank Dr. Carmine Gravino for his precious collaboration.

6. REFERENCES

[1] Arcuri, A., Briand, L., A Practical Guide for Using Statistical Tests to Assess Randomized Algorithms in Software Engineering, in Procs of International Conference on Software Engineering, (2011).

[2] Basili, V.R., The Role of Experimentation in Software Engineering: Past, Current and Future, in Procs of International Conference on Software Engineering, (1996).

[3] Bailey, J.W., Basili, V.R., A Meta Model for Software Development Resource Expenditure, in Procs of Conference on Software Engineering, pp. 107-115, (1981).

[4] Braga, P.L., Oliveira, A.L.I., Meira, S.R.L., A GA-based Feature Selection and Parameters Optimization for Support Vector Regression Applied to Software Effort Estimation, in Procs of the ACM Symposium on Applied computing, pp. 1788-1792, (2008).

[5] Briand, L., Wieczorek, I., Software Resource Estimation, Encyclopedia of Software Engineering, pp. 1160–1196, (2002).

[6] Burgess C. J., Lefley M., Can genetic programming improve software effort estimation? A comparative evaluation, Information and Software Technology, pp.863–873, 43, (2001).

[7] Chiu, N.H., Huang, S., The adjusted analogy-based software effort estimation based on similarity distances, Journal of Systems and Software, pp. 628–640, 80(4) (2007).

[8] Conte, D., Dunsmore, H., Shen, V., Software engineering metrics and models, The Benjamin/Cummings Publishing Company, Inc., (1986).

[9] Corazza A., Di Martino S., Ferrucci F., Gravino C., Mendes E., Investigating the use of Support Vector Regression for web effort estimation, Empirical Software Engineering, pp. 211-243,16(2), (2011).

[10] Corazza, A., Di Martino, S., Ferrucci, F., Gravino, C., Sarro, F., Mendes, E., How Effective is Tabu Search to Configure Support Vector Regression for Effort Estimation? in Procs of PROMISE 2010: 4.

[11] Dolado, J. J. , A validation of the component-based method for software size estimation, IEEE TSE, pp. 1006–1021, 26(10), (2000).

[12] Ferrucci, F., Gravino, C., Oliveto, R., Sarro, F., Using Evolutionary Based Approaches to Estimate Software Development Effort, in Evolutionary Computation and Optimization Algorithms in Software Engineering: Applications and Techniques, M. Chis, IGI Global, ISBN13: 9781615208098, (2010).

[13] Ferrucci, F. , Gravino, C., Oliveto, R., Sarro, F., Using Tabu Search to Estimate Software Development Effort, in Procs of IWSM/MENSURA 2009. LNCS, Springer, vol. 5891, pp. 307-320, (2009).

[14] Ferrucci, F. , Gravino, C., Oliveto, R., Sarro, F., Estimating Software Development Effort Using Tabu Search,, in Procs of the 12th International Conference on Enterprise Information Systems, pp. 236-241, 1, (2010).

[15] F. Ferrucci, C. Gravino, E. Mendes, R. Oliveto, F. Sarro, Investigating Tabu Search for Web Effort Estimation, in Procs of the 36th EUROMICRO Conference on Software Engineering and Advanced Applications, IEEE Computer Society, pp.350-357, (2010).

[16] F. Ferrucci, C. Gravino, R. Oliveto, F. Sarro, Genetic Programming for Effort Estimation: an Analysis of the Impact of Different Fitness Functions, in Procs of the 2nd International Symposium on SBSE, IEEE Computer Society, pp. 89-98, (2010).

[17] Foss, T., Stensrud, E., Kitchenham, B., Myrtveit, I., A simulation study of the model evaluation criterion MMRE, IEEE TSE, pp. 985-995, 29 (11), (2003).

[18] Harman, M, Clark, J.A., Metrics Are Fitness Functions Too, IEEE METRICS, pp. 58-69, 2004.

[19] Harman M., Jones B. F., Search based software engineering, Information and Software Technology, pp. 833–839, 43(14), (2001).

[20] Harman, M., The Current State and Future of Search Based Software Engineering, in Procs of Future of Software Engineering, pp. 342-357, (2007).

[21] Huang, S.J., Chiu, N.H., Chen, L.W., Integration of the grey relational analysis with genetic algorithm for software effort estimation, European Journal of Operational Research, pp. 898-909, 188(3), (2008).

[22] Jørgensen, M., A review of studies on expert estimation of software development effort, Journal of Systems and Software, pp. 37-60, 70(1-2), (2004)

[23] Kampenes, V., Dybå, T., Hannay, J.E., Sjøberg, D.I.K. , A Systematic Review of Effect Size in Software Engineering Experiments, Information and Software Technology, pp.1073-1086, 4(11-12), (2007).

[24] Kitchenham, B., Mendes, M., Why comparative effort prediction studies may be invalid, in Procs of PROMISE 2009: 4.

[25] Kitchenham, B., Pickard, L. M., MacDonell, S. G., Shepperd, M. J., What accuracy statistics really measure, IEE Procs Software, pp. 81-85, 148 (3), (2001).

[26] Kitchenham, B.A., Pfleeger, S.L., Pickard, L.M., Jones, P.W., Hoaglin, D.C, El Emam, K., Rosenberg, J., Preliminary Guidelines for Empirical Research in Software Engineering, IEEE TSE, pp. 721-734, 8 (2002).

[27] Koch, S., Mitlöhner, J., Software project effort estimation with voting rules, Decision Support Systems, pp. 895- 901, 46(4), (2009).

[28] Lefley M., Shepperd M. J., Using genetic programming to improve software effort estimation based on general data sets, in Procs of GECCO, pp. 2477–2487, (2003).

[29] Li, Y.F., Xie, M., Goh, T.N., A study of project selection and feature weighting for analogy based software cost estimation, Journal of Systems and Software, pp.241-252, 82(2),(2009).

[30] Mendes, E. Web Cost Estimation and Productivity Benchmarking, ISSSE, pp. 194-222, (2008).

[31] Mendes, E., Kitchenham, B. Further comparison of cross-company and within-company effort estimation models for web applications, in: Procs of International Software Metrics Symposium, IEEE press, pp. 348-357, 10 (47), (2004).

[32] Mendes, E., Mosley, N., Counsell, S., Investigating Web Size Metrics for Early Web Cost Estimation, Journal of Systems and Software, pp. 157-172, 77 (2), (2005).

[33] PROMISE Repository of empirical software engineering data, http://promisedata.org/repository.

[34] Shan, Y., Mckay, R. I., Lokan, C. J., Essam, D. L., Software project effort estimation using genetic programming, in Procs of International Conference on Communications Circuits and Systems, IEEE press, pp. 1108–1112, (2002).

[35] Shepperd, M., Schofield, C., Estimating software project effort using analogies, IEEE TSE, pp. 736-743, 23(11), (2000).

[36] Shukla, K. K., Neuro-genetic prediction of software development effort, Information and Software Technology, pp. 701–713, 42 (10), (2000).

[37] Wohlin, C., Runeson, P., Host, M., Regnell, B., Wesslen, A., Experimentation in Software Engineering-An Introduction, Kluwer Academic Publishers Norwell, MA, USA, (2000).