2010 36th EUROMICRO Conference on Software Engineering and Advanced Applications

Investigating Tabu Search for Web Effort Estimation

F. Ferrucci, C. Gravino, R. Oliveto, F. Sarro
University of Salerno
Via Ponte don Melillo, 84084s
Fisciano (SA), Italy
{fferrucci, gravino, roliveto, fsarro}@unisa.it

E. Mendes
University of Auckland
Private Bag 92019
Auckland, New Zeland
emilia@cs.auckland.ac.nz

Abstract

Tabu Search is a meta-heuristic approach successfully used to address optimization problems in several contexts. This paper reports the results of an empirical study carried out to investigate the effectiveness of Tabu Search in estimating Web application development effort. The dataset employed in this investigation is part of the Tukutuku database. This database has been used in several studies to assess the effectiveness of various effort estimation techniques, such as Linear Regression and Case-Based Reasoning. Our results are encouraging given that Tabu Search outperformed all the other estimation techniques against which it has been compared.

1. Introduction

Recently some researchers have investigated the use of Genetic Programming (GP) to estimate software development effort [2][12]. GP is a search-based approach inspired by evolutionary biology, and used to address optimization problems. There exist other search-based techniques that have been found to be very effective in solving numerous optimization problems. In particular, Tabu Search (TS) has been successfully applied within the context of software engineering (e.g., [5][11][13]). To the best of our knowledge only one case study to date has assessed the usefulness of TS for estimating software development effort [7], using a publicity available dataset (the Desharnais dataset). However, it is widely recognized that many software and Web companies for various reasons do not have their own single-company datasets [15]. Therefore investigating the effectiveness of TS using a cross-company dataset seems a reasonable research question to investigate. In addition, the datasets employed in previous studies did not include data on Web projects. Thus, the contribution of this paper is to investigate the effectiveness of TS for effort estimation, by employing data on Web projects from the Tukutuku database [16], which has been widely used in the field of Web effort estimation (see e.g., [15][16]). Furthermore, as benchmark, we used Manual StepWise Regression (MSWR) and Case-Based Reasoning (CBR) due to their frequent use in Web & software effort estimation studies (e.g., [1][17][19]), Bayesian Networks (BN) as used in [16], and the mean effort and the median effort of the training sets.

The results of our empirical analysis revealed that the choice of the objective function may influence the accuracy of the achieved estimate. Indeed, in our case study we employed two objective functions based on widely used indicators of effort prediction accuracy (i.e., Mean of Magnitude of Relative Error (MMRE) and Median of Magnitude of Relative Error (MdMRE) [4]). MMRE has been previously used as objective function of search-based methods for effort estimation [2][7], while MdMRE was never used before. Our results suggest that the use of MdMRE as objective function could be a better choice since its use allowed us to obtain significantly better estimates than the ones achieved with MMRE.

The remainder of the paper is organized as follows. Section 2 provides a brief description of TS conceived for estimating software development effort. Section 3 presents the experimental method we employed while the results of the empirical analysis are summarized in Section 4. Section 5 discusses the validity of the study we carried out. Related work are reported in Section 6. Final remarks and future work conclude the paper.

2. The estimation method

Tabu Search (TS) is an optimization method proposed originally by Glover to overcome some limitations of Local Search [8]. It is a meta-heuristic method that relies on adaptive memory and responsive exploration of the search space. To apply TS we have to perform the following steps:

- define a representation of possible solutions;

978-0-7695-4170-9/10 $26.00 © 2010 IEEE
DOI 10.1109/SEAA.2010.59

- define the neighborhood;
- choose an objective function to evaluate solutions;
- define the tabu list, the aspiration criteria, and the termination criteria.

First of all, observe that an effort estimation process can be formulated as an optimization problem, where we have to search for the model that provides the most accurate estimates, i.e. the ones that minimize the difference between estimate and actual effort. Thus, a solution consists of a model described by an equation that combines several factors:

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

where f_i and c_i represent the values of the i^{th} factor and its coefficient, respectively, C represents a constant, while op_i represents the i^{th} operator. We took into account the set { + , - , *} for our analysis.

The search space of TS is represented by all the possible equations that can be generated assigning the values for c_i, C, and op_i. The initial solution is randomly generated. Next, starting from the current solution, at each iteration the method applies local transformations (moves), defining a set of 70 neighboring solutions. In particular, each neighbor of a solution S is obtained applying a move as follows:

1. change each coefficient c_i of $expr_S$ with probability ½; the new coefficient is calculated by applying an arithmetic operator, chosen randomly in the range {+, *, - , /}, to c_i and a number r, chosen randomly in the range]0,1];
2. change the constant factor C of $expr_S$ with probability ½ in the same way coefficients are changed;
3. change each operators op_i of $expr_S$ with probability ½ by selecting another operator in {+,−, ·}.

The current solution is compared with its neighboring solutions, to decide whether or not a move to a better neighboring solution has to be performed. This is performed by exploiting an objective function able to evaluate the accuracy of the solution (i.e., the estimation model). Several accuracy measures are usually taken into account to compare effort estimation models. All are based on the residual, i.e. the difference between the predicted and actual effort. Among them, we used as objective function: MMRE and MdMRE [4], whose definitions are reported in the next section.

Thus, when the objective function value achieved by a neighboring solution is less than the one achieved by the original solution, the latter is replaced and the neighboring solution is used in the next iteration to explore a new neighborhood. Otherwise, the search continues by generating other moves starting from the original solution. To avoid loops and to guide the search far from already visited portions of the search space, the recently visited solutions are marked as tabu and stored in a tabu list. Since only a fixed and fairly limited quantity of information is usually recorded in the tabu list [8], at each iteration the tabu list contains at most seven tabu equations. In order to allow one to perform tabu move, we employed the most commonly used aspiration criterion, namely we permit a tabu move if it results in a solution with an objective function value better than the one of the current best solution. The search is stopped after a fixed number of iterations (i.e., 2000) or after some number of iterations (i.e., 250) that do not provide an improvement in the objective function value.

3. Case study design

This section presents the design of the case study we carried out to assess the effectiveness of the proposed TS in estimating software development effort. The goals of the empirical investigation were:

- assessing the effectiveness of TS in estimating Web application development effort;
- comparing the estimates achieved by applying TS with the estimates obtained with widely and successfully employed estimation methods, such as MSWR and CBR, and BN, as used in [16].

3.1 Data set description

The data used in this paper came from the Tukutuku database [16]. This database is part of the Tukutuku project, which aims to gather data from completed Web projects, to develop Web cost estimation models and to benchmark productivity across and within Web Companies. The Tukutuku database includes information on Web hypermedia systems and Web applications. The former are characterized by the authoring of information using nodes (chunks of information), links (relations between nodes), anchors, access structures (for navigation) and its delivery over the Web. In addition, typical developers are writers, artists, and organizations that wish to publish information on the Web and/or CD-ROMs without the need to use programming languages such as Java. Conversely, the latter represent software applications that depend on the Web or use the Web's infrastructure for execution and are characterized by functionality affecting the state of the underlying business logic. Web applications usually include tools suited to handle persistent data, such as local file system, (remote) databases, or Web Services. Typical developers are young programmers fresh from a Computer Science or

Software Engineering degree, managed by more senior staff. This database has data on 195 projects, where:

- Projects came mostly from 10 different countries, mainly New Zealand (47%), Italy (17%), Spain (16%), Brazil (10%), United States (4%), England (2%), and Canada (2%).
- Project types are new developments (65.6%) or enhancement projects (34.4%).
- About dynamic technologies, PHP is used in 42.6% of the projects, ASP (VBScript or .Net) in 13.8%, Perl in 11.8%, J2EE in 9.2%, while 9.2% of the projects used other solutions. The remaining projects used only HTML and/or Javascript.
- Each Web project was characterized by process and product variables as described in Table 1.

Table 1. Variables for the Tukutuku database

Var. Name	Scale	Description
COMPANY DATA		
Country	Categorical	Country company belongs to.
Established	Interval	Year when company was established.
nPeopleWD	Ratio	Number of people who work on Web design and development.
PROJECT DATA		
TypeProj	Categorical	Type of project (new or enhancement).
nLang	Ratio	Number of different development languages used
DocProc	Categorical	If project followed defined and documented process.
ProImpr	Categorical	If project team involved in a process improvement programme.
Metrics	Categorical	If project team part of a software metrics programme.
DevTeam	Ratio	Size of a project's development team.
TeamExp	Ratio	Average team experience with the development language(s) employed.
TotEff	Ratio	Actual total effort used to develop the Web application.
EstEff	Ratio	Estimated total effort necessary to develop the Web application.
Accuracy	Categorical	Procedure used to record effort data.
WEB APPLICATION		
TypeApp	Categorical	Type of Web application developed.
TotWP	Ratio	Total number of Web pages (new and reused).
NewWP	Ratio	Total number of new Web pages.
TotImg	Ratio	Total number of images (new and reused).
NewImg	Ratio	Total number of new images created.
Fots	Ratio	Number of features reused without any adaptation.
HFotsA	Ratio	Number of reused high-effort features/functions adapted.
Hnew	Ratio	Number of new high-effort features/functions.
TotHigh	Ratio	Total number of high-effort features/functions
FotsA	Ratio	Number of reused low-effort features adapted.
New	Ratio	Number of new low-effort features/functions.
TotNHigh	Ratio	Total number of low-effort features/functions

Within the context of the Tukutuku project, a new high-effort feature/function requires at least 15 hours to be developed by one experienced developer, and a high-effort adapted feature/function requires at least 4 hours to be adapted by one experienced developer. These values are based on collected data.

Summary statistics for the numerical variables are given in Table 2, while Table 3 summarizes the number and percentages of projects for the categorical variables. Note that all categorical variables are binary, and for our analysis their values have been coded using positive integers, namely 1 (for "new" and "no") and 2 (for "enhancement" and "yes").

Table 2. Summary Statistics for numerical variables

Variable	Mean	Median	Std. Dev	Min	Max
nlang	3.9	4	1.4	1	8
DevTeam	2.6	2	2.4	1	23
TeamExp	3.8	4	2.0	1	10
TotEff	468.1	88	938.5	1.1	5,000
TotWP	69.5	26	185.7	1	2,000
NewWP	49.5	10	179.1	0	1,980
TotImg	98.6	40	218.4	0	1,820
NewImg	38.3	1	125.5	0	1,000
Fots	3.2	1	6.2	0	63
HFotsA	12.0	0	59.9	0	611
Hnew	2.1	0	4.7	0	27
totHigh	1	59.6	0.0	611	611
FotsA	2.2	0	4.5	0	38
New	4.2	1	9.7	0	99
totNHigh	6.5	4	13.2	0	137

Table 3. Summary of number of projects and percentages for categorical variables

Variable	Level	Num. Projects	% Projects
TypeProj	Enhancement	128	65.6
	New	67	34.4
DocProc	No	104	53.3
	Yes	91	46.7
ProImpr	No	105	53.8
	Yes	90	46.2
Metrics	No	130	66.7
	Yes	65	33.3

3.2 Validation Method and Evaluation Criteria

A validation process is required to verify whether the investigated estimation techniques give useful estimations of the actual development efforts. To this aim, we performed a "hold-out validation" approach [9]. This approach splits the original dataset into two completely distinct datasets, namely the training and validation sets, where the former is used to train the estimation techniques and the latter to test them by assessing the evaluation parameters. In particular, we used two different splits of the original Tukutuku dataset to build estimation models using TS and to evaluate their goodness. Note that we used the same splits employed by Mendes *et al.* [16] to compare the results obtained using TS with those obtained with MSWR, CBR, and BN. In particular, each training set was obtained by randomly selecting 130 observations from the original 195 projects of the Tukutuku database, while the remaining 65 observations were included in the validation set. A set of 195

observations is considered to be good, given that most studies in Web effort estimation use sets with no more than 20 entries.

Concerning the evaluation of the estimation methods, we performed a preliminary analysis by using some summary measures, namely MMRE, MdMRE, and Pred(25) [4]. They are based on the evaluation of the Magnitude of Relative Error [4] which is defined as

$$MRE = (|EFreal-EFpred|)/EFreal \quad (2)$$

where EFreal and EFpred are the actual and the predicted efforts, respectively. MRE has to be calculated for each observation in the validation set. All the MRE values are aggregated across all the observations using the mean and the median, giving rise to MMRE and MdMRE, where the latter is less sensitive to extreme values.

The Prediction at level l [4] is defined as

$$Pred(l) = k/n \quad (3)$$

where k is the number of observations whose MRE is less than or equal to l, and n is the total number of observations in the validation set. Generally, a value of 25 for the level l is chosen. In other words, Pred(25) is a quantification of the predictions whose error is less than 25%. According to [4], a good effort prediction model should have a MMRE≤0.25 and Pred(25)≥0.75, meaning that at least 75% of the predicted values should fall within 25% of their actual values.

We also used the Magnitude of Relative Error relative to the Estimate (EMRE) as suggested by Kitchenham *et al.* [9]. The EMRE has the same form of MRE, but the denominator is the estimate, giving thus a stronger penalty to under-estimates:

$$EMRE = (|EFreal-EFpred|)/EFpred \quad (4)$$

As with the MRE, we can also calculate the mean EMRE (MEMRE) and Median EMRE (MdEMRE).

Moreover, we exploited boxplots of absolute residuals (|EFreal – EFpred|) since they can give a good indication of the distribution of residuals and can help to explain summary statistics (i.e., MMRE and Pred(l)). Indeed, boxplots provide a quick visual representation to summarize the data, using five values: median, upper and lower quartiles, minimum and maximum values, and outliers [9].

In order to verify whether the estimates obtained with TS are characterized by significantly better accuracy than the considered benchmarks (i.e., MSWR, CBR, and BN) we statistically analyzed the absolute residuals, as suggested in [9]. Since (i) the absolute residuals for all the analyzed estimation methods were not normally distributed (as confirmed by the Shapiro test [18] for non-normality), and (ii) the data was naturally paired, we employed a non-parametric test - the Wilcoxon test [3] - setting α=0.05. This test checks whether pairs of data come from the same population.

Following the suggestion of Mendes and Kitchenham [15], we also compared the results obtained with TS with those achieved using two models based on the mean effort (i.e., MeanEffort) and median effort (i.e., MedianEffort) as predicted values, respectively. The aim was to have a benchmark to assess whether the estimates obtained with a prediction technique are significantly better than estimates based on the mean or median effort. Indeed, if it is not the case the technique cannot be transferred to industry since there would be no value in dealing with complex computations compared to simply using as estimate the mean or the median effort.

4. Results and discussion

4.1 Estimates obtained with Tabu Search

The results in terms of MMRE, MdMRE, Pred(25), MEMRE, and MdEMRE obtained with TS are shown in Table 4 and Table 5, where TS1 and TS2 denotes the application of TS using MMRE and MdMRE, as objective function. We can observe that they do not fit the thresholds suggested by Conte *et al*. [4].

Moreover, we can note that TS with MdMRE as objective function (i.e., TS2) provided the best results in terms of summary measures (except for MMRE values) for both validation sets. These results are confirmed by the boxplots in Figure 1. Indeed, concerning Split 1 TS2 has a median more close to zero and box length and tails less skewed than the others. As for Split 2, the box length and tails of TS1 are slightly less skewed than the ones of TS1 and the median of TS2 is very close to the one of TS1.

In order to verify if the differences observed using summary measures and boxplots of absolute residuals were legitimate or due to chance, we also checked the statistical significance of the results by applying the Wilcoxon test on the absolute residuals. The results for the first and the second validation set are reported in Table 6, respectively, where “Yes” in a cell means that the technique indicated on the row is significantly superior to the one indicated on the column. These results showed that the absolute residuals achieved based on TS2 were significantly better than those obtained using TS1 (p-values < 0.002).

Thus, in the next section we perform the comparison with the other estimation techniques by taking into account the results obtained with TS2.

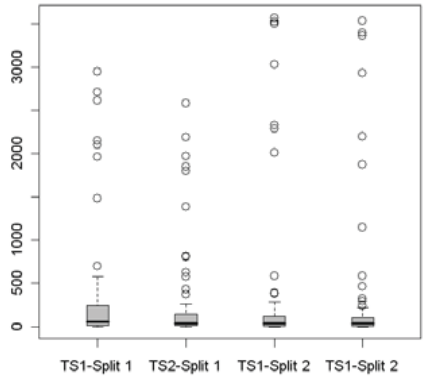

Figure 1. Boxplots of absolute residuals

Table 4. Accuracy measures on the first validation set

	MMRE	MdMRE	Pred(25)	MEMRE	MdEMRE
TS1	0.75	0.76	0.14	6.31	1.54
TS2	1.37	0.68	0.29	1.77	0.51

Table 5. Accuracy measures on the second validation set

	MMRE	MdMRE	Pred(25)	MEMRE	MdEMRE
TS1	0.80	0.66	0.18	4.72	1.42
TS2	0.99	0.49	0.31	1.81	0.54

Table 6. Results of Wilcoxon test (p-value between brackets)

	First validation set		Second validation set	
	TS1	TS2	TS1	TS2
TS1	-	No (0.998)	-	No (0.999)
TS2	Yes (0.002)	-	Yes (0.001)	-

4.2 Comparison with other techniques

In this section, we compare the results achieved using TS2 with those obtained by Mendes in [16] when applying MSWR, CBR, and BN, using the same validation sets. Table 7 (last page of the paper) reports the MMRE, MdMRE, Pred(25), MEMRE, and MdEMRE obtained in [16] We adopted the following acronyms: BNAuHu (BN automatically generated using the Hugin Bayesian Network tool), BNHyHu (BN Hybrid model using Hugin), MSWR, CBR1(CBR using one analogy), CBR2 (CBR using two analogies), CBR3 (CBR using three analogies), MeanEffort (Mean value of the effort in the dataset), MedianEffort (Median value of the effort in the dataset). Details on how these models were created can be found in [16].

The summary measures MMRE, MdMRE, Pred(25), MEMRE, and MdEMRE suggest that, for both validation sets, TS2 provided better results than the other techniques, even though they do not fit the thresholds suggested by Conte *et al*. [4]. Moreover, even if all the techniques outperform MeanEffort, only TS2 and MSWR are characterized by better prediction accuracy than MedianEffort.

Figure 2 shows boxplots of absolute residuals, obtained for the first validation set. The analysis of these boxplots confirms the patterns above mentioned. Indeed, even though the boxplots of absolute residuals of TS2 and MSWR are very similar and their medians are very close, the distribution of TS2 is less skewed than MSWR's distribution. The boxplots of absolute residuals also suggest that the estimations provided by CBR1, CBR2, CB3, and BNAuHu are worse that those obtained by using TS2, BNHyHu, and MSWR.

As for the second validation set, the boxplots of absolute residuals are shown in Figure 3 The analysis of these boxplots confirms the results provided above. Indeed, the median of the boxplots of absolute residuals for TS2 and MedianEffort are very similar and their distributions are less skewed than those of the other boxplots. Differently from the first validation set, the boxplots of absolute residuals also revealed that the results obtained with BNHyHu are worse than those obtained with the other techniques.

In order to verify if the differences observed using summary measures and boxplots are legitimate or due to chance, we again, we applied the Wilcoxon test (α = 0.05). The results, reported in Tables 8 and 9 (last page of the paper), lead us to the following observations:

1. The predictions of TS2 are significantly superior to those obtained with all the other techniques on the second validation set. With the first one, the predictions of TS2 are significantly superior to all the techniques but MSWR;
2. All the estimations obtained using the analyzed techniques (except for BNAuHu for the first validation set) are significantly superior to those using MeanEffort. Thus, for a company, it is better to use a prediction technique instead of using the simple mean effort of the previous projects.
3. Only TS2 and MSWR provided significantly superior accuracy to MedianEffort.

Thus, to summarize, regarding the research goals, the results of our empirical analysis revealed that TS did not provide estimate accuracy fitting the thresholds suggested by Conte *et al.* in [4]. However, TS has achieved significant better results than CBR for both validation sets. Moreover, TS provided significant better estimations than MSWR for the second validation set and better summary measures for the first validation set. These results are interesting since CBR and MSWR are two widely employed estimation techniques. Furthermore, TS has provided significant better results than BN, previously employed on the considered validation sets. Finally, we observed that the choice of the objective function to be employed with TS affected the accuracy of the obtained estimates. Indeed, the analysis revealed that the use of

MdMRE as objective function provided significant better estimations than the use of MMRE.

5. Threats to validity

Several factors can bias the validity of empirical studies. Here we consider three types of validity threats: Construct validity, related to the agreement between a theoretical concept and a specific measuring device or procedure; Conclusion validity, related to the ability to draw statistically correct conclusions; External validity, related to the ability to generalize the achieved results.

As highlighted by Kitchenham *et al*. [10], in order to satisfy construct validity a study has "to establish correct operational measures for the concepts being studied". The size measures and cost drivers used in the Tukutuku database and therefore in our study have been obtained from the results of a survey investigation [17], using data from 133 on-line Web forms aimed at giving quotes on Web development projects. In addition, these measures and cost drivers have also been confirmed by an established Web company and a second survey involving 33 Web companies in New Zealand. Consequently, it is our belief that the variables identified are measures that are meaningful to Web companies and are constructed from information their customers can provide at a very early stage in the project development.

In relation to the conclusion validity we carefully applied the statistical tests, verifying all the required assumptions. Moreover, we used a medium size dataset in order to mitigate the threats related to the number of observations composing the dataset. Nevertheless the use of only two splits can introduce some biases that could be overcome replicating the study using a k-fold cross validation with $k > 2$.

The dataset used in this investigation comprises data on projects volunteered by individual companies, and therefore it does not represent a random sample of projects from a defined population. This means that the results of this study only apply to the companies that volunteered data to the Tukutuku database, and other companies that develop Web applications with similar characteristics to those used herein. This represents an important external validity threat that can be mitigated only replicating the study taking into account data from other companies. Indeed, this is the only way to get a generalization of the results. However, we believe that Web companies that develop projects with similar characteristics to those used in this paper may be able to apply our results to their Web projects.

6. Related work

To the best of our knowledge, only one case study was performed to assess the use of TS for estimating software development effort while some empirical investigations were performed to assess the effectiveness of GP. In particular, we carried out a preliminary case study by applying TS on the Desharnais dataset. Differently from this work the employed dataset was single company and did not contain Web applications. Moreover, only MMRE was employed as objective function. The obtained results showed that TS had comparable performances with respect to SWR and CBR and motivated us to further investigate TS. The present case study extends the results of the original investigation showing that TS can provide superior results than SWR and CBR when applied on We applications.

As for evolutionary algorithm proposals, GP was employed by Dolado [6] in order to automatically derive equations alternative to multiple linear regression. The aim was to compare the linear equations with those obtained automatically. GP was run a minimum of 15 times and each run had an initial population of 25 equations. Even if in each run the number of generations varied, the best results were obtained with three to five generations and by using Mean Squared Error (MSE) [4] as objective function. As dataset, 46 projects developed by academic students were exploited. It is worth noting that the main goal of Dolado work was not the assessment of GP but the validation of the component-based method for software sizing. However, he observed that GP provided similar or better values than regression equations. Burgess and Lefley [2] assessed the use of GP for estimating software development effort in a case study that employed the Desharnais dataset [4]. The main parameters they used for GP were: an initial population of 1000, 500 generations, 10 run, and an objective function designed to minimize MMRE. Successively, Shepperd and Lefley [12] also assessed the effectiveness of GP and compared it with several estimation techniques such as LR (Linear Regression), ANN (Artificial Neural Networks), and CBR. As for the GP setting they applied the same choice of Burgess and Lefley [2], using a cross-company dataset (i.e., Finnish Dataset [12]). Their results showed that no single method provided superior estimates; however, GP performed consistently well.

An evolutionary computation method, named Grammar Guided Genetic Programming (GGGP), was proposed in [20] to fit models, and aiming to improve effort estimation accuracy. Data of 423 software

projects from the ISBSG database were used to build the estimation models using GGGP and LR. The objective function was designed to minimize MSE, an initial population of 1000 was chosen, the maximum number of generations was 200 and the number of run was 5. The results revealed that GPPP performed better than LR in terms of MMRE and Pred(25).

7. Conclusions and Future Work

Our analysis has shown that for Web effort estimation using cross company dataset TS can provide superior results to SWR and CBR, which are to date widely used estimation techniques. Moreover, it has revealed that the choice of the objective function may influence the accuracy of the achieved estimate. Indeed, using MdMRE as objective function allowed us to obtain significantly better estimates than the ones achieved with MMRE.

The study can be seen as a starting point for further investigations to be carried out (possibly with also other datasets) by taking into account several different objective functions. Furthermore, it could be interesting to compare the accuracy of TS with other search-based techniques (e.g., Genetic Programming [2]) since they have many similarities, but also distinguishing features. Finally, it could be 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 [14].

8. References

[1] L. Briand, T. Langley, I. Wieczorek, "A replicated assessment of common software cost estimation techniques", *Procs. Conf. on Software Engineering*, 2000, pp.377-386.

[2] C. Burgess, M. Lefley, "Can genetic programming improve software effort estimation: a comparative evaluation", *Inform. and Softw. Technology* 43(14) (2001), pp. 863–873.

[3] J. Cohen, "Statistical power analysis for the behavioral science", Lawrence Erlbaum Hillsdale, New Jersey, 1998.

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

[5] E. Diaz, J. Tuya, R. Bianco, J. J. Dolado, "A tabu search algorithm for structural software testing", *Computer and Operations Research,* 35(10), 2008, pp. 3052-3072.

[6] J. J. Dolado, "A validation of the component-based method for software size estimation", *IEEE Trans. on Soft. Engineering* 26 (10), 2000, pp. 1006-1021.

[7] Ferrucci, F., Gravino, C., Oliveto, R., and Sarro, F. "Using Tabu Search to Estimate Software Development Effort", *Procs. Inter. Conf. on Software Process and Product Measurement*, 2009.LNCS 5891.

[8] F. Glover, M. Laguna, *Tabu Search*, Kluwer Academic Publishers, Boston, 1997.

[9] B. Kitchenham, L. M. Pickard, S. G. MacDonell, M. J. Shepperd, "What accuracy statistics really measure", *IEEE Proceedings Software* 148(3), 2001, pp. 81–85.

[10] B. Kitchenham, L. Pickard, S. Peeger, "Case studies for method and tool evaluation", *IEEE Software* 12(4), 1995, pp. 52-62.

[11] L. Lanying, M. Shi, "Software-Hardware Partitioning Strategy Using Hybrid Genetic and Tabu Search" *Procs. Conf. Computer Science and Software Engineering* (*4)*, 2008, pp. 83-86.

[12] M. Lefley, M. J. Shepperd, "Using genetic programming to improve software effort estimation based on general data sets", *Procs. Genetic and Evolut. Computation Conference*, 2003, pp. 2477–2487.

[13] A. Mahmood, T.S.K. Homeed, "A Tabu Search Algorithm for Object Replication in Distributed Web Server System", *Studies in Infor. and Control*, 14(2), 2005, pp. 85-98.

[14] C. Mair, M. Shepperd, "The Consistency of Empirical Comparisons of Regression and Analogy-Based Software Project Cost Prediction", *Inter. Symposium on Empirical Software Engineering*, 2005, pp. 509-518.

[15] E. Mendes, B.A. Kitchenham, "Further Comparison of Cross-company and Within-company Effort Estimation Models for Web Applications", *Procs. Software Metrics Symposium*, 2004, pp. 348-357.

[16] E. Mendes, N. Mosley "Bayesian Network Models for Web Effort Prediction: A Comparative Study", *IEEE Trans. on Soft. Engineering* 34 (6), 2008, pp. 723–737.

[17] Mendes, E., Mosley, N., Counsell, S., "Investigating early web size measures for web cost estimation", *Procs. Eval. and Assess. in Software Engineering*, 2003., Keele. pp. 1–22

[18] P. Royston, "An extension of Shapiro and Wilks Test for Normality to Large Samples", *Applied Statistics* 31(2) (1982), pp. 115–124.

[19] M. Shepperd, C. Schofield, "Estimating software project effort using analogies", *IEEE Trans. on Soft. Engineering* 23(11), 2000, pp. 736–743.

[20] Y. Shan, R. I. Mckay, C. J. Lokan, D. L. Essam, "Software project effort estimation using genetic programming", in *Procs. Inter. Conf. on Comm. Circuits and Systems,* 2002, pp. 1108-1112.

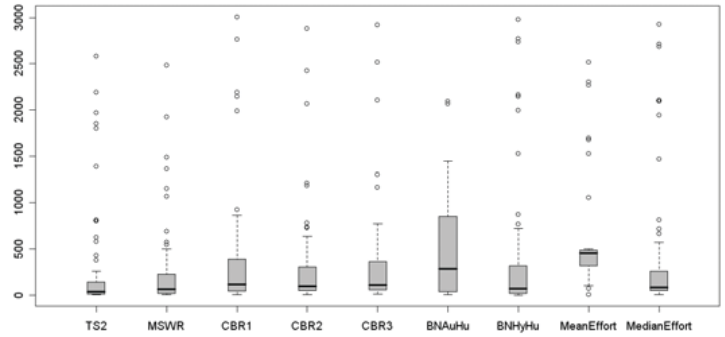

Figure 2. Boxplots of absolute residuals for the first validation set

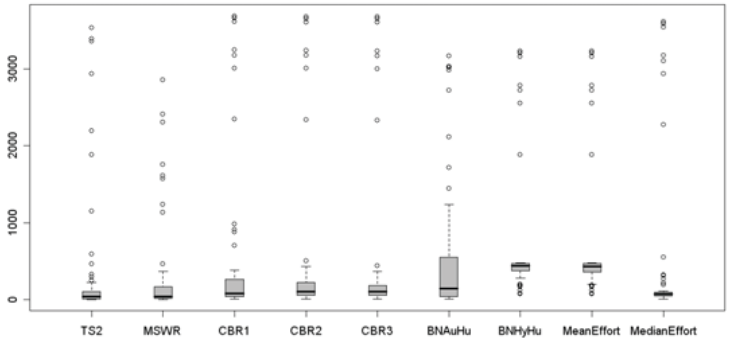

Figure 3. Boxplots of absolute residuals for the second validation set

Table 7. Accuracy measures obtained with the employed estimation techniques

	First validation set					Second validation set				
	MMRE	**MdMRE**	**Pred(25)**	**MEMRE**	**MdEMRE**	**MMRE**	**MdMRE**	**Pred(25)**	**MEMRE**	**MdEMRE**
TS2	1.37	0.68	0.29	1.77	0.51	0.99	0.49	0.31	1.81	0.54
MSWR	1.50	0.64	0.23	1.36	0.64	0.73	0.66	0.11	2.86	1.21
CBR1	5.27	0.97	0.08	31.70	3.43	4.46	0.92	0.08	21.81	0.95
CBR2	5.06	0.87	0.11	3.59	0.81	6.73	0.89	0.15	15.65	0.90
CBR3	5.63	0.97	0.09	4.17	0.88	6.09	0.84	0.09	13.26	0.89
BNAuHu	7.65	1.67	7.69	1.07	0.76	4.09	0.96	0.02	7.90	0.93
BNHyHu	1.90	0.86	0.15	13.06	2.38	27.95	5.31	0.09	1.34	0.90
MedianEffort	5.02	0.93	0.09	4.43	0.94	4.95	0.89	0.15	4.62	0.78
MeanEffort	30.35	3.99	15.38	1.07	0.91	27.94	5.31	0.03	1.34	0.90

Table 8. Results for the first validation set using the Wilcoxon test

<	MSWR	CBR1	CBR2	CBR3	BNAuHu	BNHyHu	Mean Effort	Median Effort
TS2	No (0.240)	**Yes (0.000)**	**Yes (0.000)**	**Yes (0.000)**	**Yes (0.000)**	**Yes (0.000)**	**Yes (0.000)**	**Yes (0.000)**
MSWR	-	**Yes (0.000)**	**Yes (0.002)**	**Yes(0.000)**	**Yes (0.000)**	**Yes (0.000)**	**Yes (0.000)**	**Yes (0.000)**
CBR1		-	Yes (0.052)	No (0.398)	No (0.288)		**Yes (0.022)**	
CBR2			-	No (0.227)	**Yes (0.022)**	No (0.229)	**Yes (0.000)**	No (0.335)
CBR3				-	**Yes (0.023)**	No (0.288)	**Yes (0.000)**	No (0.422)
BNAuHu					-		No (0.223)	No (0.822)
BNHyHu		**Yes (0.000)**			**Yes (0.038)**	-	**Yes (0.022)**	
Mean Effort							-	
Median Effort		**Yes (0.003)**				**Yes (0.042)**	**Yes (0.002)**	-

Table 9. Results for the second validation set using the Wilcoxon test

<	MSWR	CBR1	CBR2	CBR3	BNAuHu	BNHyHu	Mean Effort	Median Effort
TS2	**Yes (0.045)**	**Yes (0.000)**	**Yes (0.000)**	**Yes (0.000)**	**Yes (0.000)**	**Yes (0.000)**	**Yes (0.000)**	**Yes (0.000)**
MSWR	-	**Yes (0.000)**	**Yes (0.000)**	**Yes (0.001)**	**Yes (0.000)**	**Yes (0.000)**	**Yes (0.000)**	**Yes (0.009)**
CBR1		-	No (0.909)	No (0.893)	No (0.241)	**Yes (0.009)**	**Yes (0.009)**	
CBR2			-		No (0.380)	**Yes (0.000)**	**Yes (0.000)**	
CBR3			**Yes (0.000)**	-	No (0.264)	**Yes (0.000)**	**Yes (0.000)**	
BNAuHu					-	**Yes (0.000)**	**Yes (0.000)**	
BNHyHu						-	**Yes (0.000)**	
Mean Effort							-	
Median Effort		**Yes (0.000)**	**Yes (0.000)**	**Yes (0.000)**	**Yes (0.000)**	**Yes (0.000)**	**Yes (0.000)**	-