# Segmented software cost estimation models based on fuzzy clustering

Javier Aroba, Juan J. Cuadrado-Gallego, Miguel-Ángel Sicilia, Isabel Ramos,

Elena García-Barriocanal

## Abstract

Parametric software cost estimation models are based on mathematical relations, obtained from the study of historical software projects databases, that intend to be useful to estimate the effort and time required to develop a software product. Those databases often integrate data coming from projects of a heterogeneous nature. This entails that it is difficult to obtain a reasonably reliable single parametric model for the range of diverging project sizes and characteristics. A solution proposed elsewhere for that problem was the use of *segmented models* in which several models combined into a single one contribute to the estimates depending on the concrete characteristic of the inputs. However, a second problem arises with the use of segmented models, since the belonging of concrete projects to segments or clusters is subject to a degree of fuzziness, i.e. a given project can be considered to belong to several segments with different degrees.

This paper reports the first exploration of a possible solution for both problems together, using a segmented model based on fuzzy clusters of the project space. The use of fuzzy clustering allows obtaining different mathematical models for each cluster and also allows the items of a project database to contribute to more than one cluster, while preserving constant time execution of the estimation process. The results of an evaluation of a concrete model using the ISBSG 8 project database are reported, yielding better figures of adjustment than its crisp counterpart.

*Keywords:* Software effort estimation; Parametric software estimation models; Fuzzy clustering

## 1. Introduction

Parametric software cost estimation models are based on mathematical relations that intend to be useful to esti-mate the effort and time required to develop a software product. From the fifties of the past century, a large amount of parametric models have been reported (Farr and Zagorski, 1965; Herd et al., 1977; Putnam, 1978; Bay-lei and Basili, 1981; Boehm, 1981; Jensen, 1983; Rubin, 1983; Putnam and Mayers, 1992; Boehm et al., 2000), and they have provided universal mathematical equations that attempted to deal with estimations for any kind of projects.

Recent research (Dolado, 2001) has showed the large difficulty or perhaps the impossibility to obtain that kind of universal mathematical expression. Consequently some other alternatives to the universal mathematical equation like analogy-based estimation have been explored (Walkerden and Jeffery, 1999). In any case, an effort in building estimation models is required due to the importance of the time dimension of project management in organizations (Soderlund, 2005).

One of the alternatives to traditional regression is the use of segmented models (Cuadrado-Gallego et al., 2006; Cuadrado-Gallego and Sicilia, 2007). Segments can be obtained through clustering procedures that consider project distributions and divergences in variance. This method

**Nomenclature**

| | | | |
|---|---|---|---|
| **ISBSG:** | International Software Benchmarking Standards Group | **RE:** | relative error |
| **MMRE:** | mean magnitude of relative error | **EM:** | expectation maximization |
| **MMER:** | mean magnitude of error relative | **FCM:** | fuzzy C-means |

is especially useful when practitioners deal with project databases that often collect data from projects of a widely heterogeneous nature. This problem is specially relevant for databases that collect data from disparate organizations, as the ISBSG (www.isbsg.org). Such diversity entails that building a single parametric model for the whole database usually leads to poor overall adjustment. For example, if we adjust with conventional regression techniques the resulting model yields the measures of mean magnitude of relative error (MMRE) = 1.18[1] and PRED(.30) = 25.6%, which are very far from being acceptable. The ISBSG provides a tool called Reality targeted at software cost estimations for any kind of projects. An extended definition of mean magnitude of error relative to the estimate (MMER) (Foss et al., 2003), MMRE and PRED is presented in ANNEX A.

A possible solution to enhance the results obtained for the ISBSG database problem is that of using different estimation mathematical models for different parts of the project database. Such consideration results in somewhat matching the project under estimation with the "most appropriate model", which in turn would have been obtained from a concrete class of "similar" projects. This raises the need for computing mathematical devices that consider different parametric models for different inputs, ideally in a way in that the computation can be automated by an algorithm.

Clustering algorithms can help in automatically determining the classes of projects that a database records at a particular moment in time. As reported elsewhere (Cuadrado-Gallego et al., 2006; Cuadrado-Gallego and Sicilia, 2007), the use of the EM clustering algorithm to estimate the software development effort, using the data (software projects) of the ISBSG database, and considering the size and effort attributes, has demonstrated to be appropriate in obtaining segmented parametric estimation models with fairly good adjustment properties.

However, considering clusters as conventional, crisp sets leads to a problem of uncertainty for projects that are somewhere "in the middle" of several clusters (i.e. those not significantly close to the center of any of the clusters). In other words, there are cases in which a project under estimation cannot be assigned to a segment in a sharp way, so that it can be hypothetized if considering the contribution of the models of more than one cluster could improve the overall results as a way of "using more information".

Classical clustering algorithms assigns each object to one and only one of the clusters, with a degree of membership equal to one, assuming well-defined boundaries between the clusters. This model often does not reflect the description of real data, where boundaries between clusters might be fuzzy, and where a more nuanced description of the project's affinity to the specific cluster is required. However, cluster analysis is based on partitioning a collection of data points into a number of clusters, where the objects inside a cluster have a certain degree of similarity, so that in principle it seems reasonable that a project could be considered to belong (to some extent) to more than one segment.

These considerations were the rationale for experimenting with fuzzy clustering techniques as a flexible approach to produce segmented parametric models of software effort estimation. The use of clustering techniques to construct models have been used in previous works like (Yoshinari et al., 1993). Other approximations using fuzzy logic in the field of software effort estimation can be found, e.g. in Sicilia et al. (2005) fuzzy set elicitation techniques are used as a tool to model vague categories expressing cost driver quantities, and in Xu et al. (2004) the authors present a fuzzy identification cost estimation modeling technique to deal with linguistic data and automatically generate fuzzy membership functions and rules.

The research presented here is based on a model (Aroba, 2003) based on the algorithm FCM and the methodology proposed by Sugeno and Yasukawa (1993) for fuzzy clustering as the procedure to obtain the parametric models. The quality of the model is evaluated empirically by using the ISBSG database to compare the results, and also a comparison with a "crisp segment" based technique is provided.

The resulting fuzzy cluster model provides a significant advantage: several (or even all) membership functions associated to segments are used to compute every single estimation. The flexibility to consider several parametric models for a single project under estimation potentially allows the compensation of the divergences between models for clusters to which the project belongs partially.

The proposed fuzzy clustering approach, the same as any clustering algorithm, can be applied to several kinds of data sets. In this paper, the information provided by the proposed fuzzy clustering algorithm is used as a basis to develop a fuzzy segmented parametric estimation model for software development effort.

[1] The database of projects used in the Reality tool was the ISBSG release 8.

The rest of this paper is structured as follows: Section 2 provides the definition and structure of segmented model. In Section 3, the details of the experiment of the fuzzy model with the ISBSG 8 database are provided. Finally, Section 4 provides conclusions and some directions for further research.

## 2. Fuzzy segmented parametric estimation models for software development effort

This section describes the model used for the prediction of effort, as a generalization of existing segmented models (Cuadrado-Gallego et al., 2006; Cuadrado-Gallego and Sicilia, 2007) in which the predictions of development effort are obtained by combining potentially more than one of the models obtained for each segment.

### *2.1. Method to obtain the software development effort based on the fuzzy approach*

Classical effort estimation models provide a single mathematical model in the form $e = f(x_i)$ to derive effort from a number of $n$ cost drivers $x_i$ that are known to have influence in the project.

Segmented models for parametric estimation provide a number of mathematical models that are used differently for different types of inputs, which is expressed in a generic form in expression (1), where $S$ is the combination scheme used to actually compute the estimation from the inputs, each $c_j$ represents one of the $m$ inputs (clusters) and each $f_i$ is a partial model targeted at some concrete cluster (in general, the number of clusters and models is the same, so $m = n$). The same range of cost drivers is considered for each $f_i$, even though the model could be extended to deal with different inputs for each sub-model.

$$e(p \in c_j) = S(f_1(x_i) \dots f_n(x_i), c_1, \dots, c_m) \tag{1}$$

Several approaches can be used to devise the combination approach, resulting in segmented models with different predictive properties. For example, the crisp segmented model described in Cuadrado-Gallego et al. (2006), partitions the input space (for the concrete case, considering only size measured in function points) in a number of clusters obtained from the EM algorithm and then uses expression (2) to compute the effort. Essentially, the scheme partitions the input space and produce a parametric model for each partition. Consequently, the model to which an input point belongs is the only model used to derive the estimate. The following expression puts this in formal terms:

$$e(p) = f_j(p), \quad \text{where} \quad (p \in c_j) \Lambda \forall c_k \in (C - (c_j)) \rightarrow \neg(p \in c_k) \tag{2}$$

This approach has the obvious inconvenience of having a degree of possible error for inputs "between" two or more segments, since clusters are always of an approximate nature. A variant of this scheme could consider clusters with a fuzzy nature in an attempt to alleviate that potential source of error. Concretely, in the model described in this paper, each project inside the database is considered to belong to all the clusters with a membership grade, with $\mu_{c_j}(p)$ being the membership of the project $p$ to cluster $c_j$. Such belonging is determined from its proximity to cluster centers obtained from fuzzy clustering algorithms. Given that we have a parametric estimation model for each cluster, the combination of the contribution of the model associated to each cluster is aggregated by considering the extent to which the input belongs to each cluster, as expressed in (3).

$$e(p) = \mathscr{A}^{n}_{i=1}[\mu_{c_i} \cdot f_i(p)] \tag{3}$$

The technique used to compute the partial models obviously depends on the final combination scheme, in which a concrete $\mathscr{A}$ is used for the aggregation of partial contributions, and a concrete · operator is used weight the contribution from each segment. A straightforward concrete model – that is the one used in this paper from here on – is provided in (3b), where · is the standard product.

$$e(p) = \sum_{i=1}^{n} \mu_{c_i} \cdot f_i(x) \tag{3b}$$

As pointed out in Sicilia et al. (2005), it is often necessary to apply the clustering algorithm "recursively" to some of the clusters initially obtained to discriminate more sub-types of projects in case of large clusters. This recursive process in some cases does not change the compositional scheme (e.g. in expression (3), the actual origin of the final clusters is not relevant to the final outcome), but in other cases, it may be useful to consider a refinement of the original scheme.

At this point, it is necessary to describe the methodology used to obtain the membership grade of each project $i$ to each cluster $c(\mu_{c_i})$ used in (3b). For this purpose, we will use an unsupervised fuzzy clustering algorithm. In this case, there are no initial conditions on the location of cluster centroids, and classification prototypes are identified during a process of unsupervised learning.

### *2.2. Obtaining the membership grade of a project to a fuzzy cluster $\mu_{c_i}$*

Classical clustering algorithms generate a partition of the population in a way that each case is assigned to a cluster. These algorithms use the so-called "rigid partition" derived from the classical sets theory: the elements of the partition matrix obtained from the data matrix can only contain values 0 or 1; with zero indicating null membership and one indicating whole membership. That is, the elements must fulfill

$$\begin{aligned}&\text{(a)}\ \ \mu_{ik} \in \{0,1\}, \quad 1 \leqslant i \leqslant c, \quad 1 \leqslant k \leqslant n\\ &\text{(b)}\ \ \sum_{i=1}^{c} \mu_{ik} = 1, \quad 1 \leqslant k \leqslant n\\ &\text{(c)}\ \ 0 \leqslant \sum_{i=1}^{c} \mu_{ik} \leqslant n, \ 1 \leqslant i \leqslant c\end{aligned} \tag{4}$$

Fuzzy partition is a generalization of the previous one, so that it holds the same conditions and restraints for its elements, except that in this case real values between zero and one are allowed (partial membership grade). Therefore, samples may belong to more than one group, so that the selecting and clustering capacity of the samples increases. From this it can be deduced that the elements of a fuzzy partition fulfill the conditions given in Eq. (4), except that condition (a) will now be written as

$$\mu_{ik} \in [0,1], \quad 1 \leqslant i \leqslant c, \quad 1 \leqslant k \leqslant n \tag{5}$$

The best known general-purpose fuzzy clustering algorithm is the so-called fuzzy C-means (FCM) (Bezdek, 1981). It is based on the minimization of distances between two points (data) and the prototypes of cluster centres (C-means). For this purpose, the following cost function is used:

$$J(x;u;v) = \sum_{i=1}^{c} \sum_{k=1}^{n} (\mu_{ik})^m \|x_k - v_i\|_A^2, \tag{6}$$

where $U$ is a fuzzy partition matrix of $X$, $V = (v_1, v_2, \ldots, v_n)$ is a vector of cluster center prototypes which must be determined and $m \in [1, \infty]$ is a weighting exponent which determines the degree of fuzziness of the resulting clusters. Finally,

$$D_{ikA}^2 = \|x_k - v_i\|_A^2 = (x_k - v_i)^{\mathrm{T}} A (x_k - v_i) \tag{7}$$

is the norm used for measuring distances (matrix $A$ induces the rule to be used – provided that it is the unit matrix, which is very frequent, that is, the Euclidean norm). The described algorithm was used (Sugeno and Yasukawa, 1993) to build a fuzzy model based on the rules of the form

$$R' : \text{IF } x \in A' \text{ THEN } y \in B', \tag{8}$$

where $x = (x_1, x_2, \ldots, x_n) \in R$ are input variables, $A' = (A_1, A_2, \ldots A_n)$ are $n$ fuzzy sets, $y \in R$ is the output variable and $B'$ is the fuzzy set for this variable.

The used computed model is based on the previously described methodology (Sugeno and Yasukawa, 1993) and represented by Eq. (8). This initial methodology has been adapted and improved in the following aspects:

- It allows working with quantitative databases with $n$ input and $m$ output parameters.
- The different variables object of study can be weighted by assigning them weights for the calculation of distances between points of the space being partitioned.
- The achieved fuzzy clusters are processed by another algorithm to obtain graphic rules trapeziums (Fig. 1).

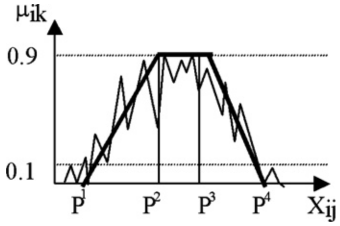

Fig. 1. Trapezoid Approximation.

- An algorithm processes and solves cases of multiple projections in the input space (mounds).
- The output provided in the original method has been improved with a graphic interface showing the graphic of the achieved rules.
- An algorithm automatically provides the interpretation of the fuzzy graphic rules in natural language.

The model was implemented in the tool PreFuRGe (Aroba, 2003), which has been used to carry out this research.

Fig. 2a and b show two examples of rules generated by means of the PreFuRGe tool.

In the rule of Fig. 2a, the fuzzy set assigned to each parameter is represented by a polyhedron. The parameter values are represented on the $x$ axis of each fuzzy set, and the value of membership to a cluster on the $y$ axis. This fuzzy rule would be interpreted as follows:

IF $A_1$ is small and $A_2$ is bigger or equal to average THEN $S$ is very small.

When applying the fuzzy clustering algorithm (Aroba, 2003) to the generated databases, it is possible to obtain multiple projections in the input parameters (mountain). In the fuzzy rule of Fig. 2b, a multiple projection (mountain) is represented in the input parameter $A_1$. In this case, how the parameter $A_1$ can take different types of values for a certain kind of output is observed. This fuzzy rule can be interpreted as follows:

IF $A_1$ is small or big and $A_2$ is average THEN $S$ is very small.

## 3. Empirical validation with the ISBSG database

The basic model expressed in (3) was the point of departure for the empirical analysis of the fuzzy model. The project database used was the ISBSG (International Software Benchmarking Standard Group) version 8, and the selected subset of data was the one used in the Reality tool of the ISBSG product, which includes the projects that are considered more reliable inside the database.

The mathematical models were obtained through standard least square method regression using the classical $e = a_0 fp^{a_1}$ model, and considering the membership value of each point to each cluster as a weight for the regression procedure. As can be seen in the first row of Table 1, the global MMER (Foss et al., 2003), MMRE and PRED

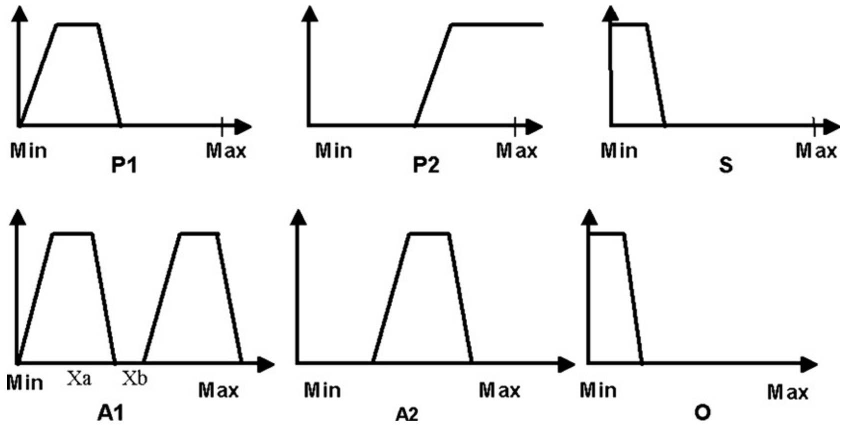

Fig. 2. Fuzzy Rules Example.

Table 1
Quality of adjustment for fuzzy models

| Fuzzy clusters | Global MMRE (%) | Global PRED(.30) (%) | Global MMER (%) |
|---|---|---|---|
| 11 | 58.76 | 51.7 | 38.57 |
| 15 | 51.28 | 61.4 | 34.11 |
| 20 | 47.5 | 59.3 | 14.29 |

are much better for eleven clusters than those that could be obtained by a single model (MMRE[SingleModel] = 1.18, Pred[SingleModel](.30) = 25.6%), but there is still room for improvement till reaching reasonably good parameters for MMER, MMRE and PRED(.30).

The process was repeated for growing numbers of clusters in search for better adjustment. Table 1 also provides the global results for 15 and 20 clusters. It can be appreciated that MMER and MMRE improves with the number of clusters, but PRED fluctuates, which manifest that simply increasing the number of required clusters is not a guarantee for increasingly better models.

One of the major difficulties encountered during fuzzy clustering of real data is that the number of clusters cannot always be defined a priori, and it is necessary to find a cluster validity criterion (Bezdek, 1981) to determine the optimal number of clusters present in the data. There are many studies on this issue (Dunn, 1974; Fukuyama and Sugeno, 1989; Gath and Geva, 1989). In the present study, we use the criterion described in Fukuyama and Sugeno (1989) for this purpose

$$S(c) = \sum_{i=1}^{c} \sum_{k=1}^{n} (\mu_{ik})^{m} (\|x_k - v_i\|^2 - \|v_i - \bar{x}\|^2), \qquad (9)$$

where $n$ is the number of data to be clustered, $c$ is the number of clusters ($c \leqslant 2$), $X_k$ $k$th data (usually vector), *Xmedia* the average of data $x_1, x_2, \ldots, x_n$, $v_i$ is the vector expressing the center of the $i$th cluster, *norma* is the norm, $\mu_{ik}$ is the grade of the $k$th data belonging to the $i$th cluster and $m$ is the adjustable weight (usually $m = 1.5$–$3$).

The number of clusters $c$, is determined so that $S(c)$ reaches a minimum as $c$ increases: it is supposed to be a local minimum as usual. As seen in (9), the first term of the right-hand side is the variance of the data in a cluster and the second term is that of the clusters themselves. Therefore, the optimal clustering is considered to minimize the variance in each cluster and to maximize the variance between the clusters.

Thus, once obtained c is we have the optimal number of projects's partitions in the database we are using, solving in this way the appearance of a possible problem of excessive segmentation that is difficult to avoid when using crisp clustering.

## 4. Conclusions

The use of parametric models is an extensively used method in the software effort estimation field. Over the years, most of the researches in parametric software effort estimation models have been centered on finding a generic expression that allows the practitioners to estimate the effort to be spent in any kind of projects.

Some recent research has pointed out the large difficulty or perhaps the impossibility to obtain that kind of universal mathematical expression. For this reason, the research described here has tried to find an alternative to universal expressions. One of the alternatives recently explored in single-relation models, especially for input domains that are large and heterogenous, is the use of segmented models. Segments can be obtained through clustering procedures that consider project distributions and divergences in variance.

Although encouraging results have been found using the segmented-based parametric software effort estimation some problems could still be found with their use. The use of fuzzy clusters in segmented models of parametric software estimation provide a convenient approach to overcome some of that problems i.e. the potential discrimination errors that arise in segmented models with crisp clusters. This paper has described a concrete approach for deriving such models, along with a case study of its application using the ISBSG 8 database.

The results point out that the fuzzy variant may present better predictive capabilities to crisp versions, providing at the same time estimates aggregated from different components coming from partial models. This alleviates the problem of points in the frontier of two clusters and also provides higher explicative capabilities.

Further work is ongoing in experimenting other fuzzy clustering algorithms for the same problem. Concretely, algorithms that attempt to extract relationships characterizing clusters like the Gath–Geva or Gusftanson–Kessel algorithms could provide additional insight in obtaining realistic and interpretable shapes for project fuzzy clusters.

## Acknowledgement

This work has been supported by the project MCYT TIN 2004-06689-C03 (IN2GESOFT).

## Annex A

When the project is finished it is possible to compare the actual values obtained with the estimated values. To do that some indicators could be used. If $V_e$ is the estimated value and $V_a$ is the actual value, the relative error (RE) and the error relative (ER) (Foss et al., 2003) to the estimates are

$$\mathrm{RE} = \frac{V_a - V_e}{V_a}$$

$$\mathrm{ER} = \frac{V_a - V_e}{V_e}$$

Frequently, it is necessary to know the relative error for a set of estimators, for example, usually it is desirable to know if the effort estimations done are accurate for a set of developed projects. The medium relative error (similar for the medium error relative) for a set of projects is

$$\overline{\mathrm{RE}} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{RE}_i$$

Also it is possible to calculate the value of these same indicators considering the absolute value $\mathrm{MMRE} = \overline{|\mathrm{RE}_i|}$, in this case, for $n$ projects the expression is

$$\mathrm{MMRE} = \frac{1}{n}\sum_{i=1}^{n} |\mathrm{RE}_i|$$

If the medium of the relative error is little, then our predictions are right. For the error relative a similar expression is obtained:

$$\mathrm{MMER} = \frac{1}{n}\sum_{i=1}^{n} |\mathrm{ER}_i|$$

These concepts are used to define a measure for the prediction quality. For a set of $n$ projects, $i$ is the number of those whose medium relative error value is less or equal to $q$:

$$\mathrm{PRED}(q) = \frac{i}{n}$$

The prediction of level $q$, $\mathrm{PRED}(q)$, gives an indication of the adjustment degree for a data set, based on the value of the RE obtained for each datum. For example if $\mathrm{PRED}(0.3) = 0.4$ the 40% of the projects have a medium relative error below 25%. To evaluate the performance of a given model, a model whose $\mathrm{MMRE} \leqslant 0.25$ and $\mathrm{PRED}(0.25) \geqslant 0.75$ (Conte et al., 1986) is considered to be a good one.

## References

Aroba, J., 2003. Ph.D. Thesis: Advances in the Decision Making in Software Development Projects, University of Sevilla, Spain.

Baylei, J., Basili, V., 1981. A meta-model for software development resource expenditures. In: Proceedings of the Fifth International Conference on Software Engineering, pp. 107–116.

Bezdek, J., 1981. Pattern Recognition with Fuzzy Objective Function Algorithm. Plenum Press, New York.

Boehm, B., 1981. Software Engineering Economics. Prentice Hall.

Boehm, B., Abts, C., Brown, W., Chulani, S., Clark, B., Horowitz, E., Madachy, R., Reifer, D., Steece, B., 2000. Software Cost Estimation with Cocomo II. Prentice Hall.

Conte, S., Dunsmore, H., Shen, V., 1986. Software engineering metrics and models. Benjamin Cummings.

Cuadrado-Gallego, J.J., Sicilia, M.A., Rodríguez, D., Garre, M., 2006. An empirical study of process-related attributes in segmented software cost-estimation relationships. Journal of Systems and Software 79 (3), 353–361.

Cuadrado-Gallego, J.J., Sicilia, M.A., 2007. An algorithm for the generation of segmented parametric software estimation models and its empirical evaluation. Computing and Informatics 26 (1), 1–15.

Dolado, J., 2001. On the problem of the software cost function. Information and Software Technology 43, 61–72.

Dunn, J., 1974. Well-separated cluster and optimal fuzzy partition. Journal Cybern 4, 95–104.

Farr, L., Zagorski, H., 1965. Quantitative analysis of programming cost factors: a progress report. In: Proceedings of the ICC Symposium on Economics of Automatic Data Processing, Amsterdam, Holland.

Fukuyama, Y., Sugeno, M., 1989. A new method of choosing the number of clusters for fuzzy c-means method. In: Proceedings of the Fifth Fuzzy System Symposium (in Japanese), pp. 247–250.

Gath, I., Geva, A., 1989. Unsupervised optimal fuzzy clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 7, 773–780.

Herd, J., Postak, J., Russell, W., Stewart, K., 1977. Software cost estimation study–study results. Final technical report, RADC-TR-77–220. Doty Associates.

Jensen, R., 1983. An improved macrolevel software development resource estimation model. In: Proceedings of the Fifth ISPA Conference, pp. 88–92.

Foss, T., Stensrud, E., Kitchenham, B., Myrtveit, I., 2003. A simulation study of the model evaluation criterion MMRE. IEEE Transactions on Software Engineering 29 (11), 985–995.

Putnam, L., 1978. A General empirical solution to the macro software sizing and estimation problem. IEEE Transactions on Software Engineering 4, 345–361.

Putnam, L., Mayers, W., 1992. Measures for Excellence: Reliable Software on Time within Budget. Yourdon Press.

Sicilia, M.A., Cuadrado-Gallego, J.J., Crespo, J., García-Barriocanal, E., 2005. Software cost estimation with fuzzy inputs: fuzzy modelling and aggregation of cost drivers. Kybernetika 41, 249–264.

Soderlund, J., 2005. What project management really is about: alternative perspectives on the role and practice of project management. International Journal of Technology Management 32 (3/4), 371–387.

Rubin, H., 1983. Macroestimation of software development parameters: the estimacs system. In: Proceedings of the SOFTFAIR Conference Development Tools, Techniques and Alternatives. IEEE Press, Arlington.

Sugeno, M., Yasukawa, A., 1993. A fuzzy-logic based approach to qualitative modeling. IEEE Transactions on Fuzzy Systems 1, 7–31.

Walkerden, F., Jeffery, R., 1999. An empirical study of analogy-based software effort estimation. Empirical Software Engineering 4, 135–158.

Xu, Z., Taghi, T., Khoshgoftaar, M., 2004. Identification of fuzzy models of software cost estimation. Fuzzy Sets and Systems 145, 141–163.

Yoshinari, Y., Pedrycz, W., Hirota, K., 1993. Construction off fuzzy models through clustering techniques. Fuzzy Sets and Systems 54, 157–165.