Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2017) 25: 2897 – 2909

doi:10.3906/elk-1604-45

Hesitant fuzzy pairwise comparison for software cost estimation: a case study in Turkey

Ayfer BAŞAR*
Department of Industrial Engineering, Faculty of Management, İstanbul Technical University, İstanbul, Turkey

Received: 05.04.2016 • **Accepted/Published Online:** 11.11.2016 • **Final Version:** 30.07.2017

Abstract: Estimating the cost of software is a complex process for almost all engineering companies. Uncertainties about development method, design, estimation process, data, and processing affect the accuracy of estimation. Underestimation results in fewer resources being committed than the project really needs, an unrealistic schedule, and low quality outputs. On the other hand, overestimation wastes resources and causes loss of customer credit. Thus, choosing the appropriate cost estimation method is crucial. Studies in the literature emphasize the importance of empirical, analytical methods and expert judgement. Certain cost estimation techniques have been widely studied in the literature. However, there are limited studies using fuzzy approaches for software cost estimation. This paper presents a hesitant fuzzy pairwise comparison (HFPC) used in the hesitant fuzzy analytic hierarchy process for software cost estimation problems by using expert judgement. For this purpose, first a number of criteria are selected with the help of expert judgements from the Turkish banking sector and information technology industry. Subsequently, the HFPC method is presented to estimate the cost of software projects. In order to analyze the efficiency of the proposed approach, it is applied to a software cost estimation problem for a Turkish company. It is seen that the proposed method provides efficient estimations due to low deviation between the real effort and estimated cost. The results are also approved by experts working in the relevant software company.

Key words: Decision making, software cost estimation, hesitant fuzzy pairwise comparison, multiple experts, case study

1. Introduction

In recent years, software development projects have become a determining factor in the competitive strength of companies. Therefore, enterprises need to plan the methods and timing of high-quality software application development. Estimated costs are important for both development teams and customers, since they are used for scheduling, negotiations, performance monitoring, etc. Accuracy of estimation is crucial because development projects are classified and prioritized according to plans based on estimation. Moreover, resource assignment and usage can be specified more effectively, change requests can be better managed, and projects can easily be monitored and controlled with respect to the cost estimation [1]. Cost of developing software can be estimated by effort (in man-hours/days/months), duration (in time), and cost (in monetary value). The cost estimation method and size measurement factors (e.g., lines of code, function points, feature points) to be used in estimation affect the accuracy of the estimation [2].

This study proposes a decision support system using hesitant fuzzy pairwise comparison (HFPC) to identify the importance of the cost factors and estimate the cost of the software projects. While estimation with

*Correspondence: ayferbasar@gmail.com

function points is based on external inputs and outputs, user interactions, external interfaces, and files used by the system, object points consider the number and complexity of the screens and the reports and modules developed for components. Therefore, object points can provide an easier and fairer estimation in comparison with function points, especially in the early phases of the software development life cycle. Object points have also been used in common models such as COCOMO II for cost estimation in the literature [3]. Moreover, object points were proposed as a replacement for function points since recent software development languages are object-oriented, dealing with classes and objects [4]. Thus, cost estimation using object points was preferred in this study. To this end, first the most important objects that affect cost estimation were identified in the literature [5–8]. The number of client (i.e. screen, user interface), application (i.e. facade, entity), batch, database (i.e. stored procedure, index), and data (table, field, sequence) objects were selected as the main cost factors by experts among many alternatives including reporting, being web-based, integration with the other modules, and warning. These factors, which cover all the entities developed for a software project, are also addressed for software design and development studies in the literature [5–8].

Subsequently, the priorities of these five main objects were identified using HFPC. Software cost estimation problems can be modeled as a hierarchy containing the decision goal, which is estimating the cost of software projects and related factors. Since expert judgement is fundamental in terms of estimation, a subjective weighting technique was required to find the importance of the cost factors. Among different subjective weighting approaches, pairwise comparison was found to be a more efficient method since it focuses on only two alternatives at a time [9]. Moreover, since decision makers focus on finding the relative importance of two factors without being affected by external factors, pairwise comparison generally gives more accurate results compared to the other weighting methods [10]. On the other hand, it is clear that there may be uncertainties while evaluating the relative importance of two factors that may not be solved by traditional pairwise comparison. For instance, the decision maker may evaluate the importance of one factor over another at "about 4", "between 1 and 2", "at least 5", or "at most 3" times. In order to cope with this uncertainty in judgement, we used hesitant fuzzy sets (HFSs) with a hesitant fuzzy linguistic scale for the pairwise comparison. An ordered weighted averaging (OWA) operator was used to aggregate expert judgements.

Table 1 summarizes the cost factors and methods used or mentioned in the study (pairwise comparison, fuzzy pairwise comparison, HFPC, analytic hierarchy process (AHP), fuzzy AHP) by giving the definition of each term, related references, and where they are used.

The proposed methods were applied in estimating the cost of 1180 real software development projects of a Turkish bank that was closed in 2015 and the deviation for all the projects was calculated according to the actual development effort.

It was found that the deviation between the real effort and estimated cost obtained by HFPC method is low; thus, the proposed method provides efficient estimations. The results were also approved by the experts of a related software company and senior managers who contributed in identifying and specifying the importance of the criteria. Due to the efficiency and effectiveness of the proposed HFPC approach, top management decided to use this method in the estimation of costs for new projects. Our unique contributions to the literature can be listed as follows:

- Identification of the importance weights of the factors that affect the cost estimation in uncertain environments,
- Developing the HFPC method model to find the weights of the cost factors,

Table 1. Cost factors and methods related to the study.

		Definition	References	Used in
Cost factors	Client objects	Screen, user interface, user control	[5–8]	Application in this study
	Application objects	Facade, entity, operation	[5,6,8]	Application in this study
	Batch objects	Database job, operating system batch	[6,7]	Application in this study
	Database objects	Stored procedure, user-defined function, index	[5–8]	Application in this study
	Data objects	Table, view, sequence, dimension	[5–7]	Application in this study
Methods	Pairwise comparison	Method of comparing alternatives by focusing on only two alternatives at a time	[23,27–31]	AHP
	Fuzzy pairwise comparison	Method of comparing two alternatives at a time by a range of values instead of crisp number	[34–36]	Fuzzy AHP
	Hesitant fuzzy pairwise comparison	Method of comparing two alternatives at a time by hesitant fuzzy scale	–	Application in this study
	AHP	Method of deriving ratio scales from paired comparisons	[23,27–31]	
	Fuzzy AHP	Method of deriving ratio scales from paired comparisons by a range of values instead of a crisp number	[34–36]	

- Integrating analytical and subjective methods by using both expert judgement and HFPC for software cost estimation,
- An application of the proposed methodology for a Turkish software company.

The rest of this paper is organized as follows: Section 2 is dedicated to studies about cost estimation methods. In Section 3, the concepts of fuzzy multicriteria decision making (MCDM) approaches for pairwise comparison including fuzzy and hesitant fuzzy extensions are discussed. The application of the proposed methodology for a Turkish software company is presented in Section 4. Finally, conclusions and further research directions are provided in Section 5.

2. Cost estimation methods

There are a number of methods to estimate software development costs in the literature. Cost estimation methods differ in two main ways: being algorithmic and nonalgorithmic. Expert judgement is one of the most widely used nonalgorithmic methods involving consulting one or more experts who use their experience in the industry and on the project. However, in most cases the level of accuracy in expert judgement is quite low because of potential bias or inconsistencies of experts. Overconfidence in judgmental forecasting is also discussed in the literature [11,12]. Therefore, different methods such as the Delphi technique and pairwise comparisons can be required to resolve the inconsistencies in expert judgement [13–15]. Parkinson's principle is another nonalgorithmic method used for software cost estimation. It is used to determine the cost based on available resources and delivery time. Simply, if the delivery time is 12 months and 5 people are available, the cost of the project is estimated to be 60 person-months [16]. Estimation by analogy is also another example of the nonalgorithmic methods. This technique estimates by analogy using the actual costs of a previous project with high similarity to the current project. Since the estimation is based on actual costs rather than forecasting, analogy can be efficient. However, it is not easy to find a similar project as it is highly unlikely that two projects will share similar risks, availability of resources, size, etc. [17].

On the other hand, algorithmic methods developed for software cost estimation are different from nonalgorithmic ones since they depend on mathematical models, which are analytical or empirical and give objective results. These methods estimate the cost of the software by using a function involving cost factors, which can be shown with $f(x_1, x_2 \ldots, x_n)$ where $x_1, x_2, \ldots, x_n$ show the cost factors. Thus, decision of cost factors and the function to be used becomes fundamental in making an accurate estimation. The best known and most common factors are based on product, computer, personnel, and project. Furthermore, algorithmic models can be linear, multiplicative, or power functions. In the linear models, cost estimation is found by $a_0 + \sum_{i=1}^{n} a_i x_i$ while $x_1, x_2, \ldots, x_n$ are cost factors and $a_1, a_2, \ldots, a_n$ are importance weights of the related criteria [18]. Multiplicative models use the $a_0 \prod_{i=1}^{n} a_i^{x_i}$ formula [19]. Moreover, power function models estimate cost by the $a \times S^b$ formula. In this formula, S represents the size of the code, while a and b are the other cost factors. COCOMO (Constructive Cost Model) is the most common and best known approach among power function models [20].

COCOMO uses code size (S) as thousands of lines of code (KLOC) and effort as person-months. Obviously, estimating the lines of codes before delivering the project is a difficult process. COCOMO models are classified in three ways: simple COCOMO, intermediate COCOMO, and COCOMO II. Simple COCOMO, which estimates the effort by the formula $a\times$ (KLOC)^b, is the earliest model developed [20]. In this formula, while S refers the code size, a and b represent the complexity of the code. The values of a and b in simple COCOMO differ according to the complexity of the software: they take values of 2.4 and 1.05 if the size and complexity is simple, 3.0 and 1.15 for more complex programming activities, and 3.6 and 1.20 if the software is complex, respectively. It is clear that simple COCOMO assumes that cost depends on code size only and it does not consider the cost factors that can affect the estimation. Therefore, intermediate COCOMO is required to obtain more accurate estimation by taking into account all relevant parameters. Intermediate COCOMO uses $a\times$ (KLOC)$^b\times$ EAF, where EAF is the effort adjustment factor. In intermediate COCOMO, the values of parameter a are different, while the values of parameter b do not change in comparison with simple COCOMO. Parameter a takes a value of 3.2 for simple programming activities, 3.0 for more complex situations, and 2.8 for complex applications in intermediate COCOMO. After determining the values of cost factors, the product of all cost factors is calculated to find the overall impact EAF. The adjustment value is 1 for a cost factor that is judged as nominal. The biggest drawback of simple and intermediate COCOMO models is that they consider a software project as a single homogeneous product while large systems can involve a number of subsystems, which can have different characteristics. Therefore, COCOMO II, the latest version of COCOMO models, was introduced in the related literature [21]. In COCOMO II, parameter b changes according to 5 new cost factors, which are precedentedness, development flexibility, risk resolution, team cohesion, and process maturity.

3. Fuzzy multicriteria decision making

MCDM deals with decision making with multiple criteria or objective where the objectives are usually conflicting and it is very difficult to select one criterion over another. It is clear that the final result in MCDM problems is highly dependent on the preferences of decision makers [22]. In the literature, the pairwise comparison method is found to be an efficient approach to specify the importance of criteria for MCDM problems [23]. However, pairwise comparison is insufficient to deal with the imprecision and subjectivity in the pairwise comparison process, since decision makers generally evaluate qualitative criteria subjectively and imprecisely in the pairwise comparison method. To overcome this drawback, fuzzy pairwise comparison is presented in the literature [24].

Fuzzy pairwise comparison uses a range of values instead of a single crisp value, and thus decision makers can select the value reflecting their confidence and specify their attitude as optimistic, pessimistic, or moderate. Furthermore, the concept of HFS, which can be used in pairwise comparisons, is a relatively new introduction to the literature, with only a few studies using this technique [25,26].

3.1. Pairwise comparison method

The pairwise comparison approach, which was originally used in the AHP method, consists of pairwise comparisons and a hierarchical structure of the factors in decision making problems [23]. The AHP has been applied to many studies in different decision making problems such as project planning, software cost estimation, technology selection, and vendor evaluation [27–31]. Thus, pairwise comparison is determined as an effective method for a series of expert judgements to find the importance criteria.

The method is applied by following steps: first, the decision hierarchy, including a goal, criteria, and subcriteria (if they exist), is prepared to make judgements in pairs. A pairwise comparison matrix (A) is then built to compute the importance of different criteria and subcriteria. Each entry (a_{ij}) of matrix A shows the importance of criterion i compared to criterion j. The relative weights of the criteria are determined according to a scale of 1–9 corresponding to the linguistic comparisons that are described in Table 2 [32]. After building matrix A, relative importance weights of the criteria $w = (w_1, w_2, \ldots, w_n)$ are found by solving $Aw = nw$ as the principal right eigenvector of A [32].

Table 2. Pairwise comparison scale used in classical AHP method.

Intensity of importance	Definition
1	Equal importance
3	Moderate importance of one over another
5	Strong or essential importance
7	Very strong or demonstrated importance
9	Extreme importance
2, 4, 6, 8	Intermediate values

3.2. Fuzzy pairwise comparison method

Fuzzy set theory is proposed to deal with imprecision in MCDM problems [33]. While in classical set theory an element must either be included in a set or not included, in fuzzy set theory, an element can belong in a set to a degree k $(0 \leq k \leq 1)$, and the degree to which elements are members of an interval is known as membership function (i.e. triangular membership function).

Fuzzy pairwise comparison was originally developed in the fuzzy AHP method, which uses the concepts of fuzzy set theory and hierarchical structure analysis in MCDM problems. Fuzzy AHP elaborates the standard AHP method into a fuzzy approach by using fuzzy numbers instead of crisp values. Similar to the AHP, fuzzy AHP has also been applied to many different studies, such as supply chain planning, outsourcing, and site selection [34–36].

In order to calculate the fuzzy importance weights by allowing the decision maker to select one alternative over another where there is uncertainty, a method is introduced employing a triangular fuzzy number, $a_{ij} = (l_{ij}, m_{ij}, u_{ij})$, where l represents the lower limit value, m refers to the most promising value, and u is the upper limit value [24]. A fuzzy comparison matrix, $A = (a_{ij})$, is then built by using the values of scales and fuzziness. After constructing the fuzzy matrix, importance weight values for each criterion and for each alternative with reference to a given criterion are calculated. In order to achieve this, first the synthetic values are obtained by:

$$S_i = \sum_{j=1}^{m} N_{ci}^{j} \otimes \left[\sum\nolimits_{i=1}^{n} \sum_{j=1}^{m} N_{ci}^{j} \right]^{-1} \tag{1}$$

In Eq. (1), $N_{ci}^{j} (j = 1, 2, \ldots, n)$ demonstrates triangular fuzzy values and $\otimes$ shows fuzzy multiplication operation. The possibility of $N_1 \geq N_2$ is given by:

$$V(N_1 \otimes N_2) = sup_{x \geq y} \min(\mu_{N1}(x), \min(\mu_{N2}(y)) \tag{2}$$

If a pair (x, y) exists such that $x \geq y$ and $\mu_{N1}(x) = \mu_{N2}(y) = 1$, then $V(N_1 \geq N_2) = 1$. Moreover, $V(N_1 \geq N_2)$ is computed by:

$$V(N_1 \geq N_2) = [l_1 - u_2]/[(m_2 - l_2) - (m_1 - l_1)] \tag{3}$$

If $m(A_i) = \min V(S_i \geq S_k)$, for $k = 1, 2, ..., n$ and $k \neq i$, then the weight vector W_p is shown as $W_A = m(A_1)$, $m(A_2), \ldots, m(A_n)\ )^T$ where $A_i (i = 1, 2, \ldots, n)$ are n elements. Finally, in order to find the importance weight vector, W_A is normalized by:

$$W_A = W^T / \left(\sum W^T \right) \tag{4}$$

3.3. Hesitant fuzzy sets

It is clear that determining the membership degree of an element is difficult since there may be possible values that make decision makers hesitate while making judgements about one criterion over another. Addressing this problem, HFS was introduced in the literature [37]. The most important details of HFS are given as follows:

Definition 1 *If X is a fixed set, the HFS on X returns a subset of [0, 1] by:*

$$E = (<x, hE(x)> | x \in X; \tag{5}$$

where $h_E(x)$ denotes the possible membership degrees of element $x \in X$ to set E by taking values in [0, 1].

Definition 2 *The upper and lower bounds are found by:*

$$h^{-}(x) = \min h(x) \tag{6}$$

$$h^{+}(x) = \max h(x) \tag{7}$$

Definition 3 *Basic operations for h, h_1, and h_2, which are HFSs, are found as follows:*

$$h^{\lambda} = \cup_{\gamma \in h}\{\gamma^{\lambda}\} \tag{8}$$

$$\lambda h = \cup_{\gamma \in h}\{1 - (1-\gamma)^{\lambda}\} \tag{9}$$

$$h_1 \cup h_2 = \cup_{\gamma 1 \in h1, \gamma 2 \in h2} \max\{\gamma^1, \gamma^2\} \tag{10}$$

$$h_1 \cap h_2 = \cup_{\gamma 1 \in h1, \gamma 2 \in h2} \min\{\gamma^1, \gamma^2\} \tag{11}$$

$$h_1 \mp h_2 = \cup_{\gamma 1 \in h1, \gamma 2 \in h2}\{\gamma^1 + \gamma^2 - \gamma^1\gamma^2\} \tag{12}$$

$$h_1 \otimes h_2 = \cup_{\gamma 1 \in h1, \gamma 2 \in h2}\{\gamma^1\gamma^2\} \tag{13}$$

Definition 4 *An OWA operator is found by:*

$$OWA(a_1, a_2, \ldots, a_n) = \sum\nolimits_{j=1}^{n} w_j b_j \tag{14}$$

Here, $b_j s$ hows the j th largest of the values of $a_1, \ldots, a_n; w_i \in$ [0, 1] $\forall i$; and $\sum_{j=1}^{n} w_j$ = 1.

Definition 5 *Depending on hesitant fuzzy linguistic term sets, comparative linguistic expressions are represented by a triangular fuzzy membership function $\tilde{A} = (a,\ b,\ c)$, where each operator is calculated by:*

$$a = \min\{a_L^i, a_M^i, a_M^{i+1}, \ldots, a_M^j, a_R^j\} = a_L^i \tag{15}$$

$$b = OWA_W\{a_M^i, a_M^{i+1}, \ldots, a_M^j\} \tag{16}$$

$$c = \max\{a_L^i, a_M^i, a_M^{i+1}, \ldots, a_M^j, a_R^j\} = a_R^j \tag{17}$$

3.4. Proposed method

The basic aim of this study is to solve the software cost estimation problem in uncertain environments by using expert judgement and analytical approaches. For this purpose, we present HFPC used in hesitant fuzzy AHP [25]. To the best of our knowledge, there is no study directly presenting HFPC and its applications. The steps of the HFPC method are as follows:

Step 1: Expert evaluations using linguistic terms are collected and pairwise comparison matrices for the criteria and subcriteria are constructed.

Step 2: The linguistic terms are transformed into triangular fuzzy numbers by the help of the scale given in Table 3 and a consistent pairwise comparison matrix $\tilde{A}^k$ including $\tilde{a}_{ij}^k$, which shows expert judgements about factor i over factor j.

Table 3. Scale used in HFPC method.

Linguistic term	Symbol	Triangular fuzzy number
Absolutely high importance	AHI	7, 9, 9
Very high importance	VHI	5, 7, 9
Essentially high importance	ESHI	3, 5, 7
Weakly high importance	WHI	1, 3, 5
Equally high importance	EHI	1, 1, 3
Exactly equal	EE	1, 1, 1
Equally low importance	ELI	0.33, 1, 1
Weakly low importance	WLI	0.2, 0.33, 1
Essentially low importance	ESLI	0.14, 0.2, 0.33
Very low importance	VLI	0.11, 0.14, 0.2
Absolutely low importance	ALI	0.11, 0.11, 0.14

Step 3: Expert evaluations are combined by fuzzy envelope approach [38]. For this aim, the scale given in Table 3 is sorted from the lowest s_0 to the highest s_g. Thus, if expert evaluations vary between s_i and s_j, then $s_0 \leq s_i < s_j \leq s_g$. Moreover, the parameters a and c of the triangular fuzzy membership function $\tilde{\mathrm{A}} = (a, b, c)$ are computed as given in Eqs. (15) and (17). Parameter b is calculated by using the OWA operator as follows:

$$b = \begin{cases} a_m^i & \text{Otherwise} \\ OWA_w\left(a_m^i, \ldots, a_m^j\right) & \text{if } i+1=j \end{cases} \tag{18}$$

The weight vector required in OWA operation is defined by [39]. This vector uses the α parameter in the unit interval [0,1]. $W = (w_1, w_2, \ldots, w_n)$ is defined as follows:

$$w_1 = a^{n-1}, w_2 = (1-a)\,a^{n-2}, \ldots, w_n = (1-a) \tag{19}$$

$$a = \frac{(g-j+i)}{(g-1)} \tag{20}$$

Here, g is the number of terms in the evaluation scale (Table 3); j is the rank of the highest and i is the rank of the lowest evaluation value.

Step 4: Collaborative pairwise comparison matrix $\hat{C}$ is constructed by $\hat{c}_i = (c_{ij}^l, c_{ij}^m, c_{ij}^u)$.

Due to the triangular fuzzy numbers, reciprocal values in $\hat{C}$ are calculated as follows:

$$\hat{c}_{ji} = (1/c_{ij}^l, 1/c_{ij}^m, 1/c_{ij}^u) \tag{21}$$

Step 5: The fuzzy geometric mean of each row $(\hat{r}_i)$ in the collaborative pairwise matrix is calculated as follows:

$$\hat{r}_i = (\hat{c}_{i1} \otimes \hat{c}_{i2} \ldots \otimes \hat{c}_{in})^{1/n} \tag{22}$$

Step 6: The fuzzy weight is calculated for each criterion $(\hat{w}_i)$.

$$\hat{w}_i = \hat{r}_i \otimes (\hat{r}_1 \mp \hat{r}_2 \ldots \mp \hat{r}_n)^{-1} \tag{23}$$

Step 7: In order to determine the ranking of factor i, triangular fuzzy numbers are defuzzified and transformed into crisp values by using Eq. (24).

$$D_i = w_i^l + 4w_i^m + w_i^u/6 \tag{24}$$

Step 8: Defuzzified importance weights of criteria are normalized and the criteria are ranked according to their normalized crisp weights.

3.5. Application of proposed method for a Turkish company

In order to identify the criteria for the software cost estimation, the factors used in the literature are analyzed and expert opinion is considered [5–8]. Six experts who work for a Turkish software company as top managers were asked to select the most important objects covering almost all entities developed for a software project. Among many alternatives affecting software development cost (e.g., testing, reporting, business process flexibility, being web-based, integration, warning), the number of client (i.e. screen, user interface), application (i.e. facade, entity, operation), batch (i.e. operating system), database (i.e. stored procedure, user-defined function), and data (table, field, sequence) objects were selected as the main cost factors by the experts.

In order to specify the importance of the factors, necessary information was gathered through an unstructured questionnaire to specify the fuzzy weights of factors affecting cost estimation by the help of face-to-face meetings. Interviews were conducted with six experts who selected the cost factors. Finally, six experts evaluated five different cost factors, namely the number of client (f_1), application (f_2), batch (f_3), database (f_4), and data (f_5) objects, according to the fuzzy linguistic scale given in Table 3. Table 4 shows the pairwise comparison matrices belonging to six experts. It is clear that the expert judgements are consistent with each other. Therefore, these judgements can be used to construct a collaborative pairwise comparison matrix using the fuzzy envelopes of the opinions of the six experts. Table 5 represents the fuzzy envelopes of all experts.

Table 4. HFPC for cost factors obtained by expert judgement.

Expert 1	f_1	f_2	f_3	f_4	f_5
f_1	EE	EHI	ESHI	VHI	AHI
f_2		EE	EE	VHI	VHI
f_3			EE	ELI	WHI
f_4				EE	VHI
f_5					EE

Expert 2	f_1	f_2	f_3	f_4	f_5
f_1	EE	WHI	WHI	WLI	EHI
f_2		EE	VHI	ELI	ESHI
f_3			EE	EHI	EHI
f_4				EE	AHI
f_5					EE

Expert 3	f_1	f_2	f_3	f_4	f_5
f_1	EE	WHI	EHI	EE	AHI
f_2		EE	VHI	VHI	VHI
f_3			EE	ESHI	WHI
f_4				EE	VHI
f_5					EE

Expert 4	f_1	f_2	f_3	f_4	f_5
f_1	EE	ESHI	VHI	ESLI	VHI
f_2		EE	ESHI	EHI	WHI
f_3			EE	WLI	EE
f_4				EE	ESHI
f_5					EE

Expert 5	f_1	f_2	f_3	f_4	f_5
f_1	EE	EHI	VHI	WHI	VHI
f_2		EE	ESHI	AHI	WHI
f_3			EE	WLI	EE
f_4				EE	ESHI
f_5					EE

Expert 6	f_1	f_2	f_3	f_4	f_5
f_1	EE	ESHI	VHI	WHI	ESHI
f_2		EE	ELI	WLI	EHI
f_3			EE	VLI	WLI
f_4				EE	VHI
f_5					EE

Triangular fuzzy sets associated with the fuzzy envelopes are computed by Eqs. (15)–(17). Table 6 presents the triangular fuzzy sets obtained by the OWA operator applied to Table 5. The detailed calculations of the associated fuzzy envelope for factor 1 over factor 2 in Table 6 are as follows: the fuzzy envelope for factor 1 over factor 2 is determined as between "essentially high importance" and "equally high importance". That is to say, $i = 6$ and $j = 8$. According to Eqs. (15) and (17), parameters a and c are minimum and maximum values of $\{a_L^i,\ a_M^i,\ a_M^{i+1},\ldots,\ a_M^j,\ a_R^j\} = \{1, 3, 5, 7\}$, respectively. Therefore, $a = 1$ and $c = 7$. In order to calculate parameter b:

Table 5. Hesitant fuzzy envelopes of six experts.

Six experts	f_1	f_2	f_3	f_4	f_5
f_1	EE	Between ESHI and EHI	Between VHI and EHI	Between VHI and ESLI	Between AHI and EHI
f_2		EE	Between ELI and VHI	Between AHI and WLI	Between VHI and EHI
f_3			EE	Between ESHI and VLI	Between WHI and WLI
f_4				EE	Between AHI and ESHI
f_5					EE

Table 6. Triangular fuzzy sets obtained by OWA operator.

	f_1	f_2	f_3	f_4	f_5
f_1	(1, 1, 1)	(1, 4.778, 7)	(1, 5.864, 9)	(0.14, 5.811, 9)	(1, 5.224, 9)
f_2	(0.143, 0.209, 1)	(1, 1, 1)	(0.33, 4.9, 9)	(0:2, 4.928, 9)	(1, 5.864, 9)
f_3	(0.111, 0.171, 1)	(0.111, 0.204, 3.03)	(1, 1, 1)	(0.11, 4.021, 7)	(0.2, 1.741, 5)
f_4	(0.111, 0.172, 7.143)	(0.111, 0.203, 5)	(0.143, 0.249, 9.091)	(1, 1, 1)	(3, 5.222, 9)
f_5	(0.111, 0.191, 1)	(0.111, 0.171, 1)	(0.2, 0.574, 5)	(0.111, 0.191, 0.333)	(1, 1, 1)

$\alpha = (g - j + i)/(g - 1) = (10 - 8 + 6)/(10 - 1) = 0.889$

$w_1 = \alpha^{2-1} = 0.889$ and $w_2 = (1 - \alpha)\alpha^{2-2} = (1 - 0.889) = 0.111(n = 2)$

Since $i + 1 \neq j$, $OWA_w\left(a_m^i, \ldots\ldots, a_m^j\right)$ is used to calculate parameter b according to Eq. (14) as follows: $b = 0.889 \times 5 + 0.111 \times 3 = 4.778$.

Thus, the fuzzy envelope for factor 1 over factor 2 is found as (1, 4.778, 7).

After determining all the triangular fuzzy sets by OWA operator in Table 6, the geometric mean of values is calculated for each row according to Eq. (22). For instance, the detailed calculations of the geometric mean of the first row are as follows:

$a_g = (1 \times 1 \times 1 \times 0.14 \times 1)^{1/5} = 0.675$

$b_g = (1 \times 4.778 \times 5.864 \times 5.811 \times 5.224)^{1/5} = 3.854$

$c_g = (1 \times 7 \times 9 \times 9 \times 9)^{1/5} = 5.515$

Thus, the geometric mean is calculated as (0.675, 3.854, 5.515) for the first row. Subsequently, the geometric means of all values in the triangular fuzzy set are normalized according to Eq. (23). In order to decrease the deviation between the weights, the highest value in the scale in Table 3, which is the maximum value of absolutely high importance (9), is selected as $(\hat{r}_1 \mp \hat{r}_2 \ldots \mp \hat{r}_n)^{-1}$ to normalize the geometric means. Normalized values of the triangular fuzzy set of the first factor are as follows:

$a_w = 0.675/9 = 0.075$ $\quad b_w = 3.854/9 = 0.428$ $\quad c_w = 5.515/9 = 0.613$

Similarly, the triangular fuzzy weights of other factors are calculated and given in Table 7. Finally, in order to obtain the final importance weight of each cost factor, the values in Table 6 are defuzzified according to Eq. (24) and normalized. Calculation of defuzzification of the first factor is as follows:

$D_1 = (0.075 + 4 \times 0.428 + 0.613)/6 = 0.400$

As seen in Table 8, the number of client objects (f_1) is the most important factor with its normalized weight of 43.7%. Number of application objects (f_2) also has a high importance weight, 24.3%. Numbers of batch (f_3) and database (f_4) objects have similar effects, with weights of 11.7% and 15.1%, respectively. Finally, number of data objects (f_5) is the least important factor for cost estimation.

Table 7. Triangular fuzzy weights of cost factors.

Factor	Fuzzy weights
f_1	(0.075, 0.428, 0.613)
f_2	(0.044, 0.219, 0.415)
f_3	(0.021, 0.084, 0,282)
f_4	(0.039, 0.060, 0.548)
f_5	(0.022, 0.036, 0.123)

Table 8. Defuzzified and normalized weights of cost factors.

Factor	Defuzzified weights	Normalized (crisp) weights
f_1	0.400	0.437
f_2	0.223	0.243
f_3	0.107	0.117
f_4	0.138	0.151
f_5	0.048	0.052

The proposed HFPC is applied to software cost estimation problem in a software company of a Turkish national bank (for confidentiality reasons, the name of the bank and software company cannot be provided) based on real data. To this end, first values of f_1, f_2, f_3, f_4, and f_5 for 1180 real software development projects completed in 2015 were obtained. Costs were then estimated using the weights in Table 8 and real values of f_1, f_2, f_3, f_4, and f_5 for each project. Subsequently, deviation between the actual development and estimated effort were calculated to analyze the effectiveness of the proposed method. Table 9 presents comparative results of HFPC and real costs.

Table 9. Benchmark of estimated and real costs for 1180 completed projects.

Interval of real costs (x) (man-days)	Number of projects	Average real cost (man-days)	Average estimated cost (man-days)	Deviation (%)
0 < x < 10	164	6.74	6.06	10.09
10 ≤ x < 50	593	35.26	29.82	15.43
50 ≤ x < 100	358	64.08	70.95	–10.72
x ≥ 100	65	192.32	218.04	–13.37

The deviation in Table 9 is calculated as 1 – (estimated / real cost). Therefore, the positive deviation values mean that the average estimated cost in the related interval is lower than average real effort. As seen in Table 9, the estimated cost obtained by HFPC was 10.09% and 15.43% lower than the project costs for projects that were lower than 10 and 10–50 man-days on average, respectively. On the other hand, estimated costs were higher than real effort by 10.72% and 13.37% for the projects whose costs were 50–100 and higher than 100 man-days on average, respectively. For all the projects, overall estimated cost was 58,250 man-days, while real effort was 57,456 man-days. Thus, the proposed method finds 1.38% higher cost estimation for 1180 projects, which shows that the proposed method provides efficient estimations due to low deviation.

Experimental study shows that estimated cost found by the proposed HFPC is acceptably lower than real effort for the projects with less than 50 man-days of effort. On the contrary, HFPC forecasts costs higher than the real effort for the bigger projects. The estimated costs of the smaller projects are found to be more acceptable and received higher approval from experts working for the company responsible for the projects and senior managers who contributed in identifying the factors. Thus, it is seen that HFPC is particularly efficient for small software projects, while it also provides reasonable estimations for big projects. Therefore, it may be more useful to estimate costs using HFPC on subtasks instead of considering the whole enterprise of large projects.

4. Conclusion and future research

In this study, we present a method to solve software cost estimation problems. Analytical methods and expert judgement are commonly used techniques to estimate costs of software projects. However, both approaches have some shortcomings. It is clear that analytical methods (i.e. COCOMO) have limited capability in considering expert opinion, and the accuracy of these methods depends on both function and cost factors used. On the other hand, estimation obtained by expert judgement alone may not be accurate if experts are not experienced and objective enough.

It is apparent that there are no studies in the literature integrating analytical and subjective methods. In our study, we benefit from the strengths and discard the shortcomings of both methods by integrating two techniques. Moreover, studies specific to fuzzy approaches are scarce and there is no study that uses HFPC to estimate software costs in an uncertain environment. In this study, after determining the most relevant cost factors via expert opinion and literature review, we obtain their importance weights by expert judgement and HFPC. We show that the number of client objects is the most important factor; numbers of application, batch, and database objects also have high effects, while the number of data objects is the least important factor in cost estimation. We apply our proposed methodology to the software cost estimation problem for a Turkish company. We also experimentally demonstrate that the proposed HFPC method provides good solutions for 1180 projects completed by this company and that it is especially reliable for estimations of projects with less than 50 man-days of effort. Thus, estimating costs by HFPC can be worthwhile for companies working on small projects or wishing to apply an agile methodology for software development.

Future studies may include development of a mathematical model using importance weights obtained by HFPC and considering different constraints.

References

[1] Aguilar-Ruiz JS, Ramos I, Riquelme JC, Toro M. An evolutionary approach to estimating software development projects. Inform Software Tech 2001; 43: 875-882.

[2] Heemstra FJ. Software cost estimation. Inform Software Tech 1992; 34: 627-639.

[3] Boehm BW. The COCOMO 2.0 software cost estimation model. Am Program 1996: 2-17.

[4] Banker R, Kauffman R, Kumar R. An empirical test of object-based output measurement metrics in a computer-aided software engineering (CASE) environment. J Manage Inform Syst 1992; 8: 127-150.

[5] Jacobson I. Object-Oriented Software Engineering: A Use Case Driven Approach. Wokingham, UK: Addison-Wesley, 1993.

[6] Ribu K. Estimating object-oriented software projects with use cases. MSc, University of Oslo, Oslo, Norway, 2001.

[7] Nesi P, Querci T. Effort estimation and prediction of object-oriented systems. J Syst Software 1998; 42: 89-102.

[8] Antoniol G, Fiutem R, Lokan C. Object-oriented function points: an empirical validation. Empir Softw Eng 2003; 8: 225-254

[9] Malczewski J. GIS and Multicriteria Decision Making. New York, NY, USA: John Wiley and Sons, 1999.

[10] Badri MA. A combined AHP–GP model for quality control systems. Int J Prod Econ 2001; 72: 27-40.

[11] Lin SW, Bier VM. A study of expert overconfidence. Reliab Eng Syst Safe 2008; 93: 711-721.

[12] Arkes H. Overconfidence in judgmental forecasting. In: Armstrong JS, editor. Principles of Forecasting. Norwell, MA, USA: Kluwer Academic Publishers, 2012. pp. 495-515.

[13] Stellman A, Greene J. Applied Software Project Management. Sebastopol, CA, USA: O'Reilly Media, 2005.

[14] Jørgensen M. A review of studies on expert estimation of software development effort. J Syst Software 2004; 70: 37-60.

[15] Miranda E. Improving subjective estimates using paired comparisons. IEEE Software 2001: 87-91.

[16] Parkinson GN. Parkinson's Law and Other Studies in Administration. Boston, MA, USA: Houghton Mifflin, 1957.

[17] Shepperd M, Schofield C. Estimating software project effort using analogy. IEEE T Software Eng 1997; 23: 736-743.

[18] Nelson R. Management Handbook for the Estimation of Computer Programming Costs. Santa Monica, CA, USA: Systems Development Corporation, 1967.

[19] Walston CE, Felix CP. A method of programming measurement and estimation. IBM Syst J 1977; 16: 54-73.

[20] Boehm BW. Software Engineering Economics. Englewood Cliffs, NJ, USA: Prentice-Hall, 1981.

[21] Boehm BW. Software Cost Estimation with COCOMO II. Englewood Cliffs, NJ, USA: Prentice Hall, 2000.

[22] Hwang CL, Yoon K. Multiple Attributes Decision Making Methods and Applications. Berlin, Germany: Springer, 1981.

[23] Saaty TL. The Analytic Hierarchy Process. New York, NY, USA: McGraw-Hill Inc., 1980.

[24] Buckley JJ. Fuzzy hierarchical analysis. Fuzzy Set Syst 1985;17:233-247.

[25] Zhu B, Xu Z. Analytic hierarchy process-hesitant group decision making. Eur J Oper Res 2014; 239: 794-801.

[26] Oztaysi B, Onar SC, Boltürk E, Kahraman C. Hesitant fuzzy analytic hierarchy process. IEEE Int Conf Fuzzy 2015: 1-7.

[27] Oztaysi, B. A decision model for information technology selection using AHP integrated TOPSIS-Grey: the case of content management systems. Knowl Based Syst 2014; 70: 44-54.

[28] Öztayşi B, Kaya T, Kahraman C. Performance comparison based on customer relationship management using analytic network process. Expert Syst Appl 2011; 38: 9788-9798.

[29] Sumeet KS, Yadwinder SB, Navdeep K. Multi-criteria decision making approach for selecting effort estimation model. Int J Comput Appl T 2012; 39: 10-17.

[30] Basar A, Kabak O, Topcu YI, Bozkaya, B. Location analysis in banking: a new methodology and application for a Turkish bank. In: Eiselt HA, Vladimir M, editors. Applications of Location Analysis. New York, NY, USA, Springer, 2014. pp: 25-54.

[31] Sivakumar R, Kannan D, Murugesan P. Discussion of green vendor evaluation and selection using AHP and Taguchi loss functions in production outsourcing in mining industry. Resour Policy 2015; 46: 64-75.

[32] Saaty TL. How to make a decision: the analytic hierarchy process. Eur J Oper Res 1990; 48: 9-26.

[33] Zadeh L. Fuzzy sets. Inform Control 1965; 8: 338-353.

[34] Ho W, Emrouznejad A, He T, Man Lee CK. Strategic logistics outsourcing: an integrated QFD and fuzzy AHP approach. Expert Syst Appl 2012; 39: 10841-10850.

[35] Rostamzadeh R, Sabaghi M, Sofian S, Ismail Z. Hybrid GA for material routing optimization in supply chain. Appl Soft Comput 2015: 26; 107-122.

[36] Tavana M, Zareinejad M, Di Caprio D, Amin Kaviani M. An integrated intuitionistic fuzzy AHP and SWOT method for outsourcing reverse logistics. Appl Soft Comput 2016: 40; 544-557.

[37] Torra V. Hesitant fuzzy sets. Int J Intell Syst 2010; 25: 529-539.

[38] Liu H, Rodriguez RM. A fuzzy envelope for hesitant fuzzy linguistic term set and its application to multi-criteria decision making. Inform Sciences 2014; 258: 220-238.

[39] Filev D, Yager RR. On the issue of obtaining OWA operator weights. Fuzzy Set Syst 1998; 94: 157-169.