Jestr

Journal of Engineering Science and Technology Review 9 (4) (2016) 217 - 223

Research Article

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

A Novel Algorithm for Software Development Cost Estimation Based on Fuzzy Rough Set

WANG Rui [1,*], PENG Pin [1], XU Ling [1], HUANG Xiao-xin [1] and QIAO Xiu-ling [2]

[1] *School of Economics & Management, Jiangxi University of Science and Technology, Ganzhou 341000, China*
[2] *INRCI, Center. de Lyon Dep., Lyon, France*

Received 11 March 2016; Accepted 2 October 2016

Abstract

Software project cost estimation is a key point for enterprises to make reasonable project quotations. However, most software cost estimation methods have limited features, such as requiring higher data volume or only having lower estimation accuracy. Aiming to resolve these problems, a novel algorithm for software development cost estimation based on fuzzy rough set was presented. First, the influencing factors of software development cost were analyzed. The objective weight of each influencing factor was obtained from the data analysis of completed projects by using rough set theory. Second, the comprehensive weight of each influencing factor was recalculated by combining the results of the first step with the subjective weight of the factors given by the experts. Combining the comprehensive weight with fuzzy theory, fuzzy similarities were calculated. Third, according to fuzzy similarity, several items that were most similar to the current project were selected as the samples from the completed projects. Then, the software development cost was estimated based on the cost data of samples from the completed project. Finally, this new algorithm was verified to be effective. The result showed that the maximum and average deviations of the fuzzy rough set algorithm were less than 10%, and the estimated maximum and average deviations of the fuzzy rough set algorithm were less than that of the fuzzy analogy algorithm. Thus, the algorithm could estimate the software cost accurately.

Keywords: Software development cost, Rough set, Fuzzy similarity

1. Introduction

Software development cost estimation is an important task in software project planning. Accurate cost estimation of software development is an important measure to control the progress of the software development, reduce risk and ensure software quality. This step is essential for project managers to obtain accurate cost estimation by accurately estimating the development cost of the new projects. Project managers could provide customers with an accurate deadline for their projects and debate negotiation issues. Furthermore, accurate estimation can be used to determine the resources needed to complete to the project. Then, the budget of each phase of the project can be determined accurately based on the estimation. However, the existing software development cost control method is not optimistic.

According to the investigation of Al-Qudah and other scholars, only 32% of software projects were successful, which indicated that these projects were completed within budget and deadline. Moreover, 24% of projects were failed, which indicated that these projects were uncompleted or cancelled. The remaining 44% of the projects were questionable. Although these projects were completed, costs have exceeded the budget [1]. Choetkiertikul and his colleagues determined that underestimating the cost of software project and theinstability of demand were two major factors that lead software projects to go beyond control [2]. Thus, determining the methods of estimating the software development cost accurately is a controversial research topic in software project management.

2. State of the art

2. 1 Software development cost estimation

Currently, the major methods for software development cost estimation are algorithm model, expert judgment, regression analysis and analogous estimating [3]. The main steps of the algorithm model steps are as follows: First, factors that affect the software development cost are analyzed. Then, the weights of these factors are calculated. Finally, the cost is estimated by a function that takes these factors as parameters. The COCOMO model is a typical representation of this type of estimation method [4-5]. This kind of method is intuitive and reusable. However, this method is ineffective when a non-linear relationship or interaction exists between factors. The main idea of the expert judgment method is to estimate

Koch, et al [6] used voting rules to estimate software development cost based on expert judgment. Although this method has the advantage of simplicity, several disadvantages have been noted. For instance, the method relies too much on the experience of experts and lacks of objectivity. For the regression analysis method, the main characteristic is taking cost-driving factors as independent variables, taking software development cost as dependent variable, and seeking the linear or nonlinear relationship between independent and dependent variables based on the

* E-mail address: 397673667@qq.com

historical data of previous projects. Liu, et al. applied nonlinear PLS regression to estimate software development cost [7]. Mittas, et al. used least squares regression to predict software development cost [8]. Although this kind of method has the advantage of high accuracy, shortcomings make this method imperfect. For example, this method requires a high data quality and is unable to deal with descriptive or uncertainties data. In fact, descriptive and uncertainties data often emerge in the process of software development cost estimation, such as the development experience on related projects and the accessibility of project requirements. The main idea of the analogy estimation method is to select one or more projects from the completed projects by analyzing those that have the most similarities with the estimated project. The development cost of the estimated project is calculated based on the cost of similar projects. Bhatia, et al. used the analogy model that was based on improved particle swarm optimization to estimate software project development cost [9]. Malashi, et al. applied fuzzy theory to calculate the similarity between proposed and completed projects. The cost of the proposed project is derived based on the similarity [10]. The accuracy of this method depends on the calculation of similarity.

In summary, the existing four common methods for software development cost estimation have their advantages and disadvantages. Therefore, a novel method for software development cost estimation based on fuzzy rough set is proposed on the bases of the advantages of the aforementioned methods. The core steps of this method are as follows: First, factors that affect software development cost are identified. Second, the comprehensive weight of each factor is calculated by combining subjective weights given by experts and objective weights calculated by data analysis of completed projects. The calculation method is based on rough set theory. Third, fuzzy similarities can be calculated by comparing the factors between the proposed software project and the completed projects. After which, several projects that have the most similarities are selected. Finally, software development cost is derived according to the analysis of the cost data of previous projects.

2.2 Factors affecting software development cost

Cost-driving factor identification is an important step of software development cost estimation. These factors are derived from literature and existing research. Wallshein et al. improved cost-driving factors in the CMMI model by adding project requirement accessibility and other cost-driving factors [11].

(1) Accessibility of the project requirements

According to the COCOMO81 model, time spent on software development is decided by the accessibility of the project requirements. The more difficulty encountered in obtaining project requirements, the more time was needed for project development. Consequently, software development would incur higher cost [12]. Thus, the acquisition of software project requirements information is an important factor that affects the development cost of software projects.

(2) Database scale

According to the LI et al., the greater the size of the database, the higher the requirement needed for the data processing algorithm. Thus, the software development cost would be higher [13]. Therefore, the scale of the software database is another influencing factor of software development cost.

(3) Software reliability

According to Wallshein's research, the workload of software development is large when the requirement of software reliability is high. Meanwhile, when the requirement of software reliability is above average, the software testing requirement is high [11]. Therefore, software reliability is a cost-driving factor.

(4) Software functions

According to Rajper and Shaikh, the more functions the software needs, the larger the workload of software development would be. This feature could result in higher software development cost [14]. Therefore, software function is an influencing factor of software development cost.

(5) Development platform

According to the literature [12, 15], better development platform means higher efficiency of software development. Thus, the software development cycle can be shortened. Moreover, when the platform is better, the platform provides more development modules. The workload of software development would be decreased sharply. Thus, the software development cost would be lower significantly. Development platform should also be considered as a cost-driving factor.

(6) Development experience on related projects

According to the literature [11, 13], if the software development team had more relevant project experience, they could be more accurate in obtaining the requirements and more efficient in controlling the development process. Thus, the software development cost could be lower. Development experience on related projects is also a cost-driving factor.

(7) Developer capability

According to the literature [11, 16], if software developers had greater research and development capabilities, the software development would be more effective, and the software development cycle would be shortened. Correspondingly, the software development cost would be reduced. Thus, developer capability is an influencing factor of software development cost.

The remainder of this paper is organized as follows. The calculation method for weight of cost-driving factors and the estimation algorithm based on fuzzy similarity are established in section 3. Section 4 shows the calculation process of the algorithm through case studies and analyzes the accuracy of the algorithm. Section 5 summarizes the conclusions.

3. Methodology

3.1 Weight of factors

In above sections, all the factors that influence software development cost were presented. In this section, the weight of each factor will be calculated. In the process of software development cost estimation, the weight of each cost-driving factor is important. AHP[17], entropy weight [18], and other methods have been used to calculate the weight. However, theses methods depend on the factor scores given by experts.

Thus, these methods are obviously subjective. Meanwhile, descriptive data and discrete data need to be processed in software development cost estimation, which form the basis for a novel method. From the historical data of the existing projects, the objective weight of each factor is calculated by rough set. Afterwards, the comprehensive weight of each factor can be determined by combining the subjective weights given by experts.

(1)Determining the objective weight

In this section, the objective weight of each factor will be calculated by the attribute importance of rough set. The process of determining objective weight is as follows. First, the decision table is established based on the costs of previous projects and data of each factor. Second, the dependence degree which refers to the degree that software development cost depends on each factor is calculated. Third, the attribute significance of each factor is calculated based on the dependence degree. Finally, the objective weight of each factor can be determined.

Definition 1 (see [19]) In knowledge system K, $K=(U,R)$. For each subset X ($X\subseteq U$), and any one of the equivalence relationship R, $R\in ind(K)$. The subsets $\overline{R}(X)$ and $\underline{R}(X)$ can be defined as follows:

$$\underline{R}X=\cup\{Y\in U/R\,|\,Y\subseteq X\} \tag{1}$$

$$\overline{R}X=\cup\{Y\in U/R\,|\,Y\cap X\neq\phi\} \tag{2}$$

Where the expression $\overline{R}(X)$ is the R-upper approximation of X. $\underline{R}(X)$ is the R-lower approximation of X.

Definition 2 (see [20]) In knowledge system K, $K=(U,R)$, and $\forall P,Q\in IND(K)$. Then, $\gamma_P(Q)$ represents the dependency degree that Q depends on P. It can be expressed by the following formula:

$$\gamma_P(Q)=k=\frac{|Pos_p(Q)|}{|U|}=\frac{|\bigcup_{X\in U/Q}\underline{P}(X)|}{|U|} \tag{3}$$

Where the expression $Pos_P(Q)$ is called positive region of Q with respect to P.It can be expressed by the following formula:

$$Pos_P(Q)=\bigcup_{X\in U/Q}\underline{P}(X) \tag{4}$$

Definition 3 When the knowledge system is a decision table T, $T=(U,C\cup D,V,f)$. Where, U is the objects set, $C(C_1,C_2,...,C_n)$ is the condition attributes set, which is constituted by the factors, and D is decision attributes set, which is the software development cost. According to the definition 2, the dependency degree that D depends on C can be expressed by the following formula:

$$\gamma_C(D)=k=\frac{|Pos_C(D)|}{|U|}=\frac{|\bigcup_{X\in U/D}\underline{P}(X)|}{|U|} \tag{5}$$

Definition 4 For the decision table $T=(U,C\cup D,V,f)$, $sig(C_i)$ represents the significance of condition attribute C_i with respect to decision attribute D. $sig(C_i)$ can be expressed by the following formula:

$$sig(C_i)=\gamma_C(D)-\gamma_{\{C-C_i\}}(D) \tag{6}$$

All $sig(C_i)(C_i\in C)$ can be calculated by using the previously presented formulas. Then, $sig(C_i)$ can reflect influence degree that the cost-driving factor C_i affects the software development cost. After normalization, the objective weight of factor C_i, which is represented by Wo_i, can be determined by the following formula:

$$Wo_i=(\gamma_C(D)-\gamma_{C-C_i}(D))/\sum_{i=1}^{n}(\gamma_C(D)-\gamma_{C-C_i}(D)) \tag{7}$$

(2) Determining the comprehensive weight

In the previous section, the objective weight of each factor is calculated. Then, the subjective weight of each factor, which is represented by Ws_i, is given by experts. Based on Wo_i and Ws_i, the comprehensive weight of each factor, which is represented by W_i, is synthesized by the following formula:

$$W_i=L\times Wo_i+(1-L)Ws_i \tag{8}$$

where $0\le L\le 1$. L is an empirical coefficient given by experts according to the features of software project for estimated.

3.2 Method for Software development cost estimation

Finding existing projects, which are similar to the current software project, is the key to estimating the software development cost. In this process, calculating the similarity between the current project and historical projects is crucial. We used fuzzy nearness degree method to calculate the similarity.

(1) Calculation of the similarity

Definition 5. There are two projects. They are represented by A and B. $\mu_A(C_i)$ is the fuzzy membership of A based on factor C_i. Similar to $\mu_A(C_i)$, $\mu_B(C_i)$ is the fuzzy membership of B based on factor C_i. Then, we used $\alpha(A,B)$ to express the fuzzy nearness degree between A and B. According to fuzzy theory, $\alpha(A,B)$ can be calculated by the following formula:

$$\alpha(A,B)=\frac{\sum_{i=1}^{n}(\mu_A(C_i)\wedge\mu_B(C_i))}{\sum_{i=1}^{n}(\mu_A(C_i)\vee\mu_B(C_i))} \tag{9}$$

where, " $\wedge$ " and " $\vee$ " denotes the minimum and the maximum values respectively. According the section 3, each factor is entitled weight. Therefore, after the introduction of the weight of the factor, the formula will be expressed by the following formula.

$$\alpha(A,B)=\frac{\sum_{i=1}^{n}W_i(\mu_A(C_i)\wedge\mu_B(C_i))}{\sum_{i=1}^{n}W_i(\mu_A(C_i)\vee\mu_B(C_i))} \quad (10)$$

(2) Estimating the software development cost

In the process of software development cost estimation, the projects that are the same as the current project can not be selected from the finished projects. However, similar projects can be found. Therefore, the most similar projects are selected from the finished projects. From the costs of these projects, the development cost of the current software project is calculated by the exponential smoothing method.

Given n sample projects, $\alpha_i(i=1,2,...,n)$ represents the similarity between project i and the current project. The similarities are arranged in descending order. The series $\alpha_1,\alpha_2,...,\alpha_n$ can be obtained. With respect to $\alpha_1,\alpha_2,...,\alpha_n$, $M_1,M_2,...M_n$ represent the cost data series. We selected the cost data of the three projects, which are most similar to that of the current project, as samples to facilitate the calculation. From these costs data, the development cost of the current software project can be calculated by the following formula: software develoment cost based on the experience of experts and the historical data of previous projects. For example,

$$M^*=\alpha_1M_1+\alpha_2M_2(1-\alpha_1)+\alpha_3M_3(1-\alpha_1)(1-\alpha_2)+(1-\alpha_1)(1-\alpha_2)(1-\alpha_3)(M_1+M_2+M_3)/3 \quad (11)$$

where the expression M^* is the estimated development cost of the current software project.

4. Result analysis and discussion

4.1 Algorithm calculation process

Human resource cost is one of the main parts of software development cost. However, human resource cost varies widely in different times. For example, the average salary is quite different in different years. Therefore, we use "developers × months" to measure the software development cost. For example, if a software project requires 20 developers and 2 months for completion, the development cost of the project will be 40 "developers × months". In table 1, "developers × months" is abbreviated as "d*m". Table 1 shows the case database of a software development company in 2014.

Table 1. Case database of software projects

No	C_1	C_2	C_3	C_4	C_5	C_6	C_7	D (d*m)
1	low	small	medium	medium	better	more	general	42(d*m)
2	medium	small	medium	medium	better	more	general	45(d*m)
3	high	small	medium	medium	better	more	general	75(d*m)
4	medium	medium	medium	more	general	less	strong	90(d*m)
5	high	medium	medium	more	general	less	strong	120(d*m)
6	high	large	medium	medium	better	more	general	105(d*m)
7	medium	medium	high	more	better	less	strong	105(d*m)
8	low	medium	high	more	better	less	strong	85(d*m)
9	low	large	medium	medium	general	medium	strong	75(d*m)
10	low	medium	medium	medium	general	medium	strong	50(d*m)
11	medium	small	high	more	better	less	strong	80(d*m)
12	medium	large	high	more	better	less	strong	108(d*m)
13	low	medium	medium	medium	general	medium	general	55(d*m)
14	low	medium	medium	medium	general	more	general	48(d*m)
15	medium	medium	medium	more	general	medium	strong	80(d*m)
16	high	large	high	more	general	medium	general	110(d*m)
17	high	large	high	more	general	more	general	95(d*m)
18	medium	small	high	medium	better	more	general	68(d*m)
19	medium	medium	high	medium	general	medium	general	80(d*m)
20	low	medium	medium	more	general	medium	strong	58(d*m)
21	medium	small	medium	medium	general	more	general	60(d*m)
22	high	large	high	more	general	medium	strong	96(d*m)

In the first row of Tab.1, C_1, C_2, C_3, C_4, C_5, C_6 and C_7 represent the accessibility of the project requirements, database scale, software reliability, software functions, development platform, development experience on related projects and ability of developers respectively. D represents the software development cost. Given that most data in tab.1 are descriptive, thus these data need to be discretely processed. The discrete rule is shown in Tab.2. After discretely processing the data in Tab.1, Tab.3 can be obtained.

Table 2. Discrete rule

Factors	1	2	3
C_1	low	medium	high
C_2	small	medium	large
C_3	general	high	
C_4	medium	more	
C_5	general	better	
C_6	less	medium	more
C_7	general	strong	
D	≤ 50	> 50 and ≤ 100	>100

Table 3. Discrete case database

No	C_1	C_1	C_1	C_1	C_1	C_1	C_1	D
1	1	1	1	1	2	3	1	1
2	2	1	1	1	2	3	1	1
3	3	1	1	1	2	3	1	2
4	2	2	1	2	1	1	2	2
5	3	2	1	2	1	1	2	3
6	3	3	1	1	2	3	1	3
7	2	2	2	2	2	1	2	3
8	1	2	2	2	2	1	2	2
9	1	3	1	1	1	2	2	2
10	1	2	1	1	1	2	2	1
11	2	1	2	2	2	1	2	2
12	2	3	2	2	2	1	2	3
13	1	2	1	1	1	2	1	2
14	1	2	1	1	1	3	1	1
15	2	2	1	2	1	2	2	2
16	3	3	2	2	1	2	1	3
17	3	3	2	2	1	3	1	2
18	2	1	2	1	2	3	1	2
19	2	2	2	1	1	2	1	2
20	1	2	1	2	1	2	2	2
21	2	1	1	1	1	3	1	2
22	3	3	2	2	1	2	2	2

(1) **Comprehensive weight calculation**

First, according the method of above section, the objective weight of each factor can be calculated. The steps are as following:

U/C={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22}

U/D={(1,2,10,14),(3,4,8,9,11,13,15,17,18,19,20,21,22),(5,6,7,12,16)}

Then, according to the formulas (1), (2) and (4), we can derive the following equations:

$$Pos_C(D) = U$$

$$Pos_{C\text{-}C_1}(D) = \{U - (1,2,3) - (4,5) - (7,8)\}$$

$$Pos_{C\text{-}C_2}(D) = \{U - (3,6) - (9,10) - (7,11,12)\}$$

$$Pos_{C\text{-}C_3}(D) = \{U - (2,18)\}$$

$$Pos_{C-C_4}(D) = \{U - (10,20)\}$$

$$Pos_{C\text{-}C_5}(D) = \{U - (2,21)\}$$

$$Pos_{C\text{-}C_6}(D) = \{U - (13,14) - (16,17)\}$$

$$Pos_{C\text{-}C_7}(D) = \{U - (16,22)\}$$

According to the formula (5), the following expressions can be obtained.

$$\gamma_C(D) = \frac{|Pos_C(D)|}{|U|} = \frac{22}{22} = 1$$

$$\gamma_{C_1}(D) = \frac{|Pos_{C_1}(D)|}{|U|} = 15/22$$

The same calculation process is then applied to derive the following expressions:

$$\gamma_{C\text{-}C_2}(D) = 15/22,\ \gamma_{C\text{-}C_3}(D) = 20/22,\ \gamma_{C\text{-}C_4}(D) = 20/22,$$

$$\gamma_{C\text{-}C_5}(D) = 20/22,\ \gamma_{C\text{-}C_6}(D) = 18/22,\ \gamma_{C\text{-}C_7}(D) = 20/22.$$

According to the formula (6) in definition 4, each $sig(C_i)$ can be calculated as follows:

$$\mathrm{sig}(C_1) = \gamma_C(D) - \gamma_{C\text{-}C_1}(D) = 7/22,\ sig(C_2)=7/22,$$

$$sig(C_3)=2/22,\ sig(C_4)=2/22,\ sig(C_5)=2/22,\ sig(C_6)=4/22,$$

$$sig(C_7)=2/22.$$

According to the formula (7), the objective weight of each factor can be calculated as follows:

$$Wo_1 = (\gamma_C(D) - \gamma_{C-C_1}(D)) / \sum_{i=1}^{7} (\gamma_C(D) - \gamma_{C-C_i}(D)) = 7/26 = 0.27$$

The same calculation process is employed to calculate Wo_1, Wo_2, Wo_3, Wo_4, Wo_5, Wo_6 and Wo_7. The results are $Wo_2 = 0.27$, $Wo_3 = 0.077$, $Wo_4 = 0.077$, $Wo_5 = 0.077$, $Wo_6 = 0.152$ and $Wo_7 = 0.077$.

Second, combining the subjective weight given by experts, the compressive weight of every cost-driving factor can be determined by formula (8). The process is as follows:

According to the experience of software project management and development, the subjective weight of every cost-driving factor can be obtained. The results are $Ws_1 = 0.25$, $Ws_2 = 0.2$, $Ws_3 = 0.1$, $Ws_4 = 0.1$, $Ws_5 = 0.1$, $Ws_6 = 0.15$ and $Ws_7 = 0.1$.

Meanwhile, let the empirical coefficient $L = 0.6$. According to the formula (8), the comprehensive weight of every cost-driving factor can be calculated on the basis of Wo_i and Ws_i, as follows:

$$W_1 = L \times Wo_1 + (1 - L)Ws_1 = 0.6 \times 0.27 + 0.4 \times 0.25 = 0.262$$

The same calculation process is employed to calculate W_2, W_3, W_4, W_5, W_6 and W_7. The results are $W_2 = 0.242$,

$W_3 = 0.0862$, $W_4 = 0.0862$, $W_5 = 0.0862$, $W_6 = 0.1512$ and $W_7 = 0.0862$.

(2) Projects similarity calculation

Six similar projects in Tab.1 are selected as samples by the experts to show the calculation process of the projects similarity. Meanwhile, the fuzzy memberships (μ_i) of these samples are also given by experts. The data are presented in Tab.4.

Table 4. The fuzzy memberships of the samples

No	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7	*D*(d*m)
1	0.4	0.3	0.2	0.3	0.5	0.8	0.2	42
2	0.5	0.3	0.3	0.2	0.6	0.8	0.3	45
14	0.35	0.4	0.4	0.25	0.4	0.7	0.3	48
10	0.3	0.5	0.3	0.2	0.4	0.6	0.6	50
13	0.3	0.6	0.2	0.4	0.3	0.5	0.4	55
20	0.3	0.6	0.3	0.6	0.2	0.5	0.7	58

Project no.1 is used as an example to illustrate the process of software development cost estimation. Accordingto formula (10), the similarity between project nos.1 and 2 can be calculated as follows:

$$\alpha(1,2) = \frac{0.262 \times 0.4 + 0.242 \times 0.3 + \ldots + 0.0862 \times 0.2}{0.262 \times 0.5 + 0.242 \times 0.3 + \ldots + 0.0862 \times 0.3} = 0.866$$

Taking the three largest similarities as $\alpha_1, \alpha_2, \alpha_3$. Then, $\alpha_1 = \alpha(1,2) = 0.866$, $\alpha_2 = \alpha(1,14) = 0.8$ and $\alpha_3 = \alpha(1,10) = 0.665$

Finally, according to the formula (11) , the development cost estimation value of project no.1 can be calculated as follows:

M^*=0.866×45+0.8×48×(1-0.866)+0.665×50×(1-0.866) ×(1-0.8)+((1-0.866) ×(1-0.8) ×(1-0.665) ×(45+48+50))/3
=45.3(d*m)

4.2. Algorithm validity analysis

(1) Algorithm accuracy analysis

Similar to the estimation process of the project no.1, the estimated costs of other items are shown in Tab.5.

From the analysis of the actual and estimation costs shown in Tab.5, the deviation of the fuzzy rough set algorithm is determined and presented in Fig.1.

Table 5. Estimation result of fuzzy rough set algorithm

Fuzzy rough set algorithm(FRS)								
No	Similarity			Cost of sample (d*m)			Actual cost	Estimation cost
	α_1	α_2	α_3	M_1	M_2	M_3		
1	0.866	0.8	0.665	45	48	50	42	45.3
2	0.866	0.774	0.673	42	48	50	45	42.9
14	0.804	0.8	0.78	50	42	45	48	48.6
10	0.804	0.803	0.796	48	55	58	50	49.6
13	0.87	0.803	0.708	58	50	48	55	56.9
20	0.87	0.796	0.657	55	50	48	58	54.5

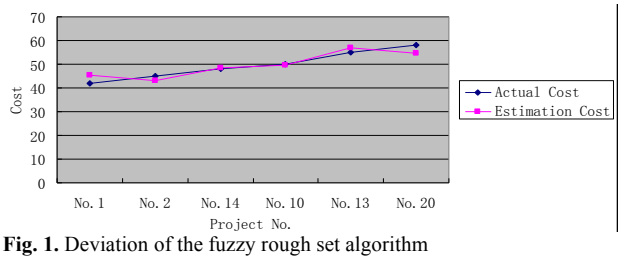

Fig. 1. Deviation of the fuzzy rough set algorithm

Fig.1 indicates that the maximum and average deviations of the fuzzy rough set algorithm are less than 10%. Thus the algorithm can effectively estimate software development cost.

(2) Algorithm comparisons

Fuzzy analogy algorithm is a type of important software cost estimation method. This method overcomes the shortcoming that depends on expertise to determine the similarity between projects. This algorithm uses fuzzy set theory to calculate similarity, which is more objectivite. According to formula (9), (10) and (11), the development cost can be estimated by using the fuzzy algorithm based on the data shown in Tab.4. The results are shown in Tab.6.

Table 6. Estimation result of fuzzy analogy algorithm

Fuzzy analogy algorithm(FA)								
No	Similarity			The cost of sample (d*m)			Actual cost	Estimation cost
	α_1	α_2	α_3	M_1	M_2	M_3		
1	0.838	0.774	0.65	45	48	50	42	45.5
2	0.838	0.784	0.686	42	48	50	45	42.9
14	0.784	0.781	0.774	45	50	42	48	45.7
10	0.781	0.75	0.743	48	55	58	50	49.5
13	0.79	0.75	0.692	58	50	48	55	56.3
20	0.79	0.743	0.6	55	50	48	58	53.7

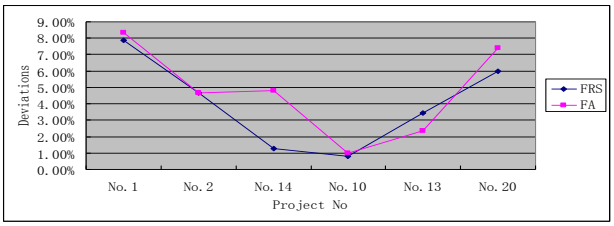

Fig. 2. Algorithm deviation comparisons

From the comparison of the deviations of fuzzy rough set algorithm and fuzzy analogy algorithm, Fig.2 can be obtained.

Fig.2 illustrates that the maximum deviation of fuzzy rough set algorithm (7.86%) is less than the maximum deviation of fuzzy analogy algorithm (8.33%). The average deviation of fuzzy rough set algorithm (4%) is less than the average deviation of fuzzy analogy algorithm (4.76%).

Therefore, estimation accuracy of fuzzy rough set algorithm is better than the estimation of fuzzy analogy algorithm.

5. Conclusion

The estimation of software development cost is always the critical and difficult point in software project management. A novel algorithm for software cost estimation based on fuzzy rough set is proposed in this study to improve the accuracy of the cost estimation. Based on the actual software development cost data, the validity of the algorithm is verified. The following conclusions are obtained:

(1)The proposed algorithm can deal with descriptive and uncertainty data in the process of software development cost estimation. The maximum and the average estimation deviations of the algorithm are less than 10%. In other words, the algorithm is effective in software development cost estimation.

(2)The maximum estimation deviation of the proposed algorithm is less than the maximum estimation deviation of the fuzzy analogy algorithm. In addition, the average estimation deviation of the proposed algorithm is also less than the average estimation deviation of the fuzzy analogy algorithm. Thus, the estimation accuracy of the proposed algorithm is better than the estimation accuracy of the fuzzy analogy algorithm. With the increase in historical projects data, the estimation accuracy of fuzzy rough set algorithm will be improved.

The proposed algorithm can not only be used to estimate software development cost, but can also be used in other fields. For example, the algorithm can be used to estimate the development cost of new products. However, the algorithm has some limitations. For example, this algorithm requires a certain amount of historical data as samples when making estimations. An insufficient amount of sample data will affect the estimation accuracy of the algorithm. The further study is needed to accurately estimate the software development cost in the case of insufficient historical data.

Acknowledgement

This work was supported by Young Teachers' Ability Improving Project of School of Management & Economics of JXUST under the project No. jgxy201502.

References

1. Al-Qudah, S., Meridji, K., Al-Sarayreh, K.T., "A comprehensive survey of software development cost estimation studies". In: *Proceedings of the international conference on intelligent information processing, security and advanced communication*, New York, USA: ACM, 2015, pp.53-58.
2. Choetkiertikul, M., Dam, H.K., Tran, T., Ghose, A., "Characterization and prediction of issue-related risks in software projects". *In: Proceedings of the 12th working conference on mining software repositories*, Florence, Italy: IEEE, 2015, pp.280-291.
3. Pandey, P., "Analysis of the techniques for software cost estimation". In: *Proceedings of the 3rd international conference on advanced computing and communication technologies*, Rohtak, India: IEEE, 2013, pp. 16-19.
4.Boehm, B.W., Valerdi, R., "Achievements and Challenges in COCOMO-based on Software Resource Estimation". *IEEE Software*, 25(5), 2008, pp.74-83.
5. Kazemifard, M., Zaeri, A., Ghasem-Aghaee, N., Nematbakhsh, M.A., Mardukhi, F., "Fuzzy Emotional COCOMO II Software Cost Estimation (FECSCE) using Multi-Agent System". *Applied Soft Computing Journal*, 11(2), 2011, pp.2260-2270
6. Koch, S., Mitlöhner, J., "Software Project Effort Estimation with Voting Rules". *Decision Support Systems*, 46(4), 2009, pp.895-901
7. Liu, H.T., Wei, R.X., Tian, Z.Q., "Military software cost estimation based on nonlinear PLS regression". *Systems Engineering and Electronics*, 36(7), 2014, pp.1352-1357.
8. Mittas, N., Angelis, L., "LSEbA: Least squares regression and estimation by analogy in a semi-parametric model for Software Cost Estimation". *Empirical Software Engineering*, 15(5), 2010, pp.523-555.
9. Bhatia, P., Mishra, K.K., Misra, A.K., "An approach to software cost estimation by improved - Time variant acceleration coefficient based PSO". *Journal of Multiple-Valued Logic and Soft Computing*, 27(1), 2016, pp.63-74.
10. Malathi, S., Sridhar, S., "Effort estimation in software cost using team characteristics based on fuzzy analogy method – A diverse approach". *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, Dubai, UAE, 2012, pp.1-8.
11. Wallshein, C.C., Loerch, A.G., "Software cost estimating for CMMI Level 5 developers". *Journal of Systems and Software*, 105, 2015, pp.72-78.
12. Hari, C.H., Reddy, P.V.G.D., "A fine parameter tuning for COCOMO 81 software effort estimation using particle swarm optimization". *Journal of Software Engineering*, 5(1), 2011, pp.38-48.
13. Li, B., Zhang, L., "Comparison and Choice of Typical Software Cost Estimation Methods". *Computer Integrated Manufacturing Systems*, 14(7), 2008, pp.1441-1448.
14. Rajper, S., Zubair, A. S., "Software Development Cost Estimation: A Survey". *Indian Journal of Science and Technology*, 9(31), 2016, pp.1-5.
15. Huang, J.L, Sun, H.Y., Li, Y.F., "An empirical study of the impact of project factors on software economics". *2015 International Conference on Industrial Engineering and Engineering Management,* Singapore: IEEE, 2015, pp.43-47.
16. Chowdary, V., and Reddy, V.K., "Software Effort Estimation: A Comparative Analysis". *International Journal of Progressive Sciences and Technologies*, 2(2), 2016, pp. 48-60.
17. Zhang, S.Z., Zeng, Q.D., "A new approach for prioritization of failure mode in FMECA using encouragement variable weight AHP". *Applied Mechanics and Materials*, 289, 2013, pp.93-98.
18. Li, R.J., Yao, K.W., "Resettlement implementation effect evaluation based on entropy weight - Principal component analysis". *Advanced Materials Research*, 864, 2014, pp.2257-2262
19. Pawlak, Z., Zymala-Busse, J., Slowinski, R., "Rough sets". *Communications of the ACM*, 38(11), 1995, pp.88-95.
20. Pawlak, Z., "Rough set theory and its applications to data analysis". *Cybernetics & Systems*, 29(7), 1998, pp.661-688.