

Prioritizing interdependent software requirements using tensor and fuzzy graphs

Negin MISAGHIAN¹, Homayun MOTAMENI^{1*}, Mohsen RABBANI²

¹Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran

²Department of Mathematics, Sari Branch, Islamic Azad University, Sari, Iran

Received: 24.06.2018

Accepted/Published Online: 10.04.2019

Final Version: 26.07.2019

Abstract: Owing to the special stance of prioritizing tasks in requirements engineering processes, and as the requirements are not independent in nature, considering their dependencies is essential during the prioritizing process. Although different classifications of dependency types among requirements exist, only a few approaches in the prioritization process consider such valuable data (dependency among requirements). To achieve a practical prioritization, this study proposes a method based on the effects of the requirement dependencies (increase/decrease cost of) on the value of prioritization provided by the tensor concept. Since the strengths of dependencies are also influential factors in the act of prioritization, the algebraic structure of fuzzy graphs is used to model the requirement dependencies and their strengths. Moreover, a weighted page rank algorithm based on the fuzzy concept is provided to determine the final dependency strength of the dependent requirements of the fuzzy graph. To evaluate the proposed approach, a controlled experiment is also conducted. The proposed approach is compared with an analytic hierarchy process-based approach, TOPSIS, and EVOLVE in the experiment. The results analysis demonstrates that our approach is less time-consuming, much easier to use, and highly accurate.

Key words: Requirements prioritization, requirements engineering, software engineering, interdependent requirements, tensor, fuzzy graphs

1. Introduction

Due to limited resources and frequent releases in mobile application systems, prioritization of requirements is a key process [1]. Prioritizing refers to making a balance between the benefits of a business and other parameters such as cost involved [2], but considering this process from such diverse commercial and technical aspects can be a challenging issue for stakeholders [3–6]. Consequently, several techniques have been developed to provide requirements prioritization, among which are the analytic hierarchy process (AHP) [3, 7], cost-value analysis [8, 9], the multidimensional approach [10], and the technique of ordered preference by similarity to ideal solution (TOPSIS) [11]. In the AHP [12] the process can be divided into four steps: S1) decomposing the structure of the problem into a hierarchy of elements, S2) comparing all possible pairs of the attributes of decision elements based on their importance value to form a reciprocal matrix (usually a nine-point scale with 1 representing equal importance and 9 representing absolutely more importance is used), S3) calculating the weights and priorities of the elements, and S4) calculating the consistency ratios.

Based on the tensor concept, the multidimensional approach [10] provides a prioritization technique and takes the simultaneous relations among the requirements and stakeholders into account using this tensor

*Correspondence: motameni@iausari.ac.ir

modeling. Tucker decomposition of the tensor is also used in this approach to provide the final ranking of the requirements. This approach is based on the concept of revealing latent relations among the entities simultaneously. Each entity forms a dimension. By applying multiway analysis, an appropriate ordered list of requirements will be obtained.

TOPSIS was first provided by Hwang and Yoon [11], and it evaluates the alternatives simultaneously based on two main criteria types: cost (in which less is better) and benefit (in which more is better). The criteria of the problem were divided into these two main types and received a related weight. Two positive and negative ideal solutions or alternatives were created to measure the distance of the alternatives. Therefore, an alternative closer to the positive ideal alternative and farther away from the negative ideal alternative received a higher rank.

It should be indicated that when there are no dependencies among the requirements, these techniques could provide the prioritized list according to the available resources and their related parameters. However, the fact that approximately 20% of all requirements are independent creates a situation that is nowhere near the ideal [8]. Thus, dependency is a critical factor when considering prioritizing the requirements, which becomes even more challenging when the requirement scale increases in number and when the stakeholders are forced to prioritize their requirements not only from personal preferences but from requirement dependencies [6]. With these points in mind, the task is challenging in providing an effective prioritizing technique that can satisfy the needs of the stakeholders and the requirements dependencies. In addition, as value-related requirement dependencies can have different strengths that can be classified accordingly [13] as weak, moderate, and strong [14], taking into account these value-related requirement dependencies and the existence of them are critical steps during the process of prioritizing the requirements. Accordingly, in this study, we have considered the “increase/decrease cost of” dependency type during requirements prioritization and we propose an approach by combining link analysis and the tensor concept to prioritize dependent requirements. First, by the algebraic structure of fuzzy graphs, we model requirement dependencies and their strengths [15], and we propose a weighted page rank algorithm on the basis of the fuzzy graphs to compute the final strength in each requirement dependency. We have investigate the results with the primary order provided by the tensor concept to produce the final prioritization order of the requirements. Accordingly, Section 2 provides a summary of the related literature, and in Section 3, necessary background knowledge on tensor algebra and fuzzy graphs is considered. Section 4 proposes the approach with a step-by-step presentation through an example, and in Section 5, a controlled experiment is depicted. Sections 6 and 7 cover the discussion of the results and the conclusion, respectively.

2. Related works

Several studies investigated requirements prioritization [6, 16, 17] and focused on several issues such as dependency, scalability, and rank update that scholars face during the process of techniques to tackle requirements prioritization. Furthermore, types of dependencies have been classified as refinement, contribution, and conflict relations in GORE and i^* [18, 19] by Pohl [4] and Dahlstedt and Persson [20]. Zhang et al. [21] divided dependencies into “increase/decrease” value and cost along with business-related relations such as “precedence and concurrence dependencies”. Although requirements dependencies are important factors in the prioritization process and the final ordered results [21, 22], dependencies between and among the requirements is a less investigated field, which is the rationale for using an indirect factor in current studies.

A linguistics-based approach, named SNIPR [23], considered the dependencies between the requirements

by grouping them as well as considering stakeholders' pairwise comparisons. In the first phase, it exploited natural language processing to assist users in identifying interdependencies and constraints between requirements as input into a satisfiability modulo theories (SMT) solver. This phase also involved calculating the ranking of the requirements while estimating the uncertainty associated with each requirement's rank based on different ranked sets generated by the SMT solver. The second phase involved reranking a subset of requirements for improved accuracy.

In addition, EVOLVE was used in software development to generate the final decision through opting for innate technical precedences of the requirements [24]. It combined the computational strength of genetic algorithms with the flexibility of an iterative solution method. At each iteration, a genetic algorithm was applied to determine an optimal or near optimal assignment of requirements. The process needed prioritization of requirements by stakeholders, effort estimation for all requirements, effort constraints, precedence, and coupling constraints, as well as changes in the weights assigned to stakeholders.

A TOPSIS-based tool was provided in [9] to consider the prerequisites for the requirement dependencies, which change the score of each requirement provided by TOPSIS to set the score of each as less than each of its prerequisites/dependencies. Moreover, [25] proposed Drank (a method of semiautomated requirements prioritization) to integrate the preferences and dependencies of the requirements using a weighted page rank algorithm. In Drank, a prioritization evaluation attributes tree (PEAT) is developed, and then a subjective requirements prioritization using RankBoost is created. Through extracting business dependencies and the contribution among the requirements using the i^* model, a weighted page rank algorithm is used to assess the contribution order of each requirement. Finally, the final prioritization list of requirements is provided by integrating the subjective requirements along with prioritizing the contribution order and the business relationships among the requirements.

Software requirement selection was optimized in [26] to investigate the dependency between requirements using a fuzzy graph. An optimization model was proposed in the study to test the dependencies of value-based requirements and their effects on the appointed requirements. Fuzzy-graph structures were used to model value-based requirement dependencies alongside their strengths. The authors provided a new value measure for an optimal set as overall value (OV), which observes the value-based requirement dependencies of the selected requirements.

In summary, none of the above works considered the preference or dependency type (increase/decrease cost of) among the requirements while using a fuzzy concept. Since considering the dependencies among the requirements is important for requirements prioritization, the focus of this paper has been on the "increase/decrease cost of" dependency type among the requirements in prioritizing them. We have considered both the existence and the strengths of the "increase/decrease cost of" requirement dependencies during the prioritization task. The main contributions of our work are as follows: first, we have modeled the "increase/decrease cost of" dependency type and its strengths using the algebraic structure of fuzzy graphs. Second, we have used a weighted page rank algorithm on the basis of the fuzzy graphs to compute the final strength in each requirement dependency that not only takes into account the dependencies among the requirements but also considers the strengths of "increase/decrease cost of" dependency using a provided modified weighted page rank algorithm regarding the fuzzy uncertainty concept. Finally, we have presented a requirement prioritization approach, integrating the dependency order of requirements with the requirements order provided by tensor decomposition.

3. Fuzzy graphs and tensor algebra

The application of graph theory covers a broad range of combinational problem areas in scientific research, and fuzzy systems have vast applications in diverse scientific areas, including decision-making systems and social sciences. Fuzzy graphs has been used extensively in modern sciences and technologies since their introduction in 1975 [15].

Considering S as a finite set, a fuzzy subset of S is identified as mapping $\xi:S\rightarrow[0,1]$, which determines a degree of membership $0\leq\xi(X)\leq 1$ for each element of $X\in S$. A fuzzy relation $f:S\times S\rightarrow[0,1]$ is also defined on two fuzzy subsets of S_1 and S_2 with membership function of $\xi_1:S_1\rightarrow[0,1]$, $\xi_2:S_2\rightarrow[0,1]$, respectively, iff $\forall(x,y)\in S_1\times S_2, f(x,y)\leq\xi_1(x)\wedge\xi_2(y)$. In addition, a fuzzy graph can be depicted as $G=(S,\xi,f)$ made of functions of $\xi:S\rightarrow[0,1]$ and $f:S\times S\rightarrow[0,1]$ for each $x,y\in S$, where $f(x,y)\leq\xi(x)\wedge\xi(y)$ [15].

The dependencies among requirements are fuzzy relations in general [27] because the strength of requirements dependencies varies from high to low in real-world projects [14, 27]. Fuzzy graphs are used in engineering and computer sciences, via the factor of uncertainty, to provide effective models [28–32]. Hence, fuzzy graphs can model the directed fuzzy relations among requirements more accurately. Furthermore, a directed fuzzy graph $G=(R,C,\xi,f)$ is defined such that its nodes are a set of nonempty identified requirements $R=\{r_1,\dots,r_n\}$ and the edges show the explicit cost relation among the requirements as $C=R\times R$ [26]. In addition, the dependency $(r_i,r_j)\in C$ represents that the cost implementation of r_j explicitly depends on implementation of r_i . The strengths of explicit dependencies are also defined by membership function $f:R\times R\rightarrow[0,1]$, where $f(x,y)=0$ indicates that there is no explicit dependency between x and y . Moreover, the membership function of requirement r_i in the requirement set R is defined as $\xi:(r_i)\rightarrow[0,1]$. Since all the software requirements are listed in its requirement set R , we have $\forall r_i\in R:\xi(r_i)=1$, and we can abbreviate the fuzzy graph $G=(R,C,\xi,f)$ as $G=(R,C,f)$.

For $G=(R,C,f)$ to be a fuzzy graph we consider the following condition that must be hold at all times: $\forall(x,y)\in C:f(x,y)\leq\xi(x)\wedge\xi(y)$,

where \wedge denotes a fuzzy AND operator (taking infimum). It means the membership degree of a relation $f(x,y)$ that shows the strength of the relation (dependency) must not exceed the membership degree of the two elements (x,y) .

An example of a fuzzy requirement interdependency graph is sketched in Figure 1. As shown, the requirements set $R=\{r_1,r_2,r_3,r_4\}$ and cost dependency relations $C=\{(r_2,r_1),(r_4,r_2),(r_4,r_1),(r_4,r_3)\}$ have been considered.

The strengths of explicit dependencies in C as membership function f are set as below:

$$f(r_2,r_1)=0.4, f(r_4,r_2)=0.75, f(r_4,r_1)=0.5, f(r_4,r_3)=0.36.$$

Explicit dependency (r_2,r_1) shows that the implementation cost of r_1 explicitly depends on r_2 and the strength of the dependency is determined by $f(r_2,r_1)=0.4$.

There are both implicit and explicit dependencies in a fuzzy requirement interdependency graph, in which they can be identified via the graph edges and explicit dependencies, respectively. For example, from explicit dependencies (r_4,r_2) and (r_2,r_1) we can infer the implicit dependency (r_4,r_2,r_1) .

Furthermore, the implementation cost of a requirement r_i can positively or negatively impact the implementation cost of requirement r_j . This has been considered in the “increase/decrease cost of” dependency type in this paper.

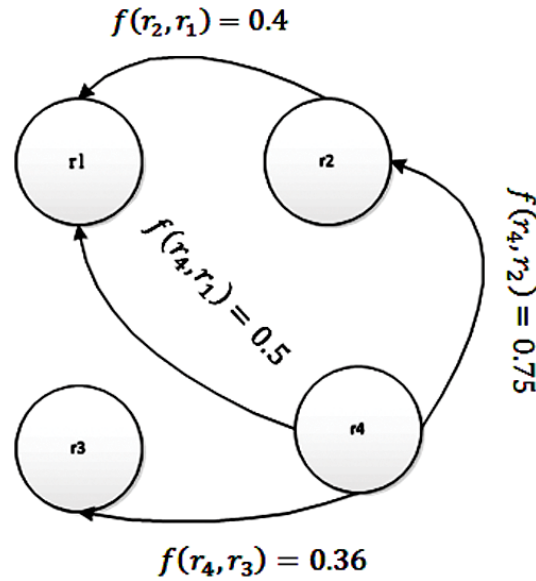


Figure 1. Sample fuzzy graph of requirements.

A tensor can be defined as a multidimensional array that generalizes the representation of the matrix [33], and each dimension of the tensor is called a mode. $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ displays an N -order tensor \mathcal{A} , where its elements are a_{i_1, \dots, i_N} , and the size of the n th mode \mathcal{A} is I_n . To achieve a high-order tensor, a mode- n flattening is used to cut up a tensor \mathcal{A} on the mode- n axis and connect the cut-up matrices into one matrix $\mathbf{A}(n)$ [33]. By expanding the singular value decomposition (SVD) to that of high-order tensors (here a three-order tensor), we can derive the Tucker decomposition [34].

4. Proposed approach

In describing our approach, we first mention the assumptions. The requirements conflicts have been solved by negotiation so that they are conflict-free during the prioritization process. The “increase/decrease cost of” dependencies among requirements are also identified and modeled; that is, the dependencies are confirmed by requirements engineers and experts. Given these assumptions, our approach includes the following steps as in Figure 2.

- **Eliciting importance values of functional requirements (FRs) regarding nonfunctional requirements (NFRs):** In this step, stakeholders extract the importance value of each NFR regarding FRs through a five-point scale [10].
- **Generate primary prioritization list using tensor concept:** In this step the primary list of prioritized requirements is provided using the tensor concept as proposed in [10]. The steps are listed as following:
 1. The three-order tensor is constructed from stakeholders’ preference information.
 2. The Tucker decomposition is used to reveal relations among entities of the tensor.
 3. The reconstructed tensor is calculated.

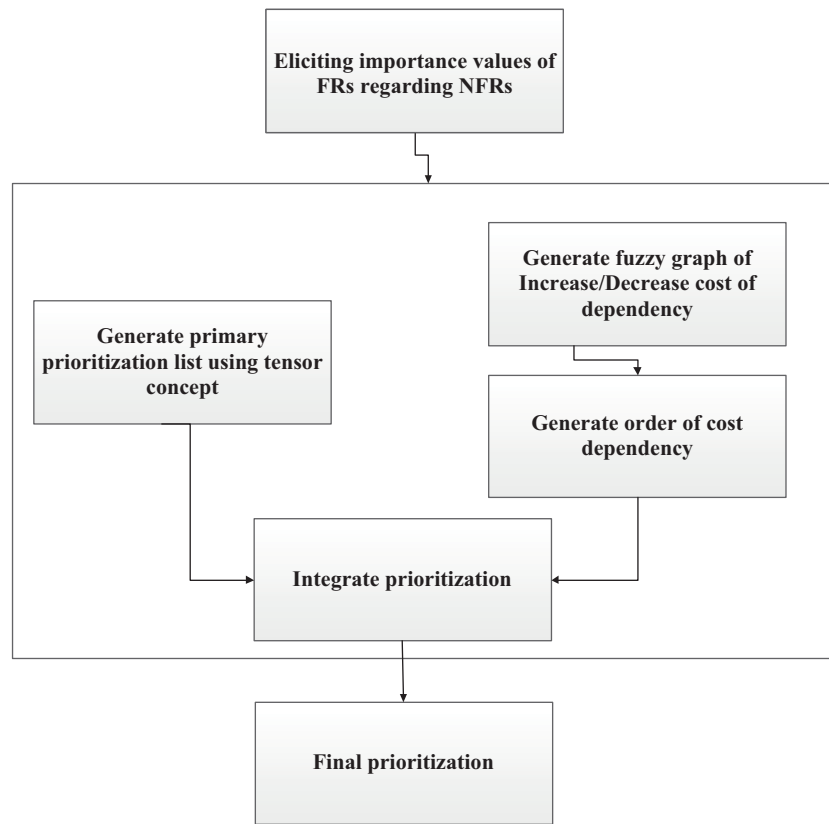


Figure 2. Proposed approach.

4. Finally, the ordered list of FRs is provided.

- **Generate fuzzy graph of “increase/decrease cost of” dependency:** Cost-related dependencies are used in order to generate a fuzzy requirement dependency graph. A fuzzy cost graph (FCG) is derived based on the “increase/decrease cost of” dependency type (details are provided in Section 4.1).
- **Generate order of cost dependency:** According to the generated fuzzy graph, the cost dependency strength and transitivity are used to compute the dependency influence, and the computed dependencies are used to generate the cost-order of requirements.
- **Integrate prioritization:** The cost dependency can affect the implementing requirements. The final ordered list of the requirements is obtained by adjusting the primary list based on the cost dependency.
- **Final prioritization:** The integrated list of the requirements is provided as the final ordered list.

4.1. Example 1

An example is used in order to display the steps in our proposed model in detail. Five decision makers assisted us to study 3 FRs and 4 NFRs. Based on the steps provided in Figure 2, the importance values of FRs regarding NFRs are gathered as shown in Table 1. This shows how much importance each nonfunctional requirement has according to the functional requirement.

Table 1. DMs' preference values of NFRs with respect to FRs.

DMs	NFR1			NFR2			NFR3			NFR4		
	(FRs)			(FRs)			(FRs)			(FRs)		
	FR1	FR2	FR3	FR1	FR2	FR3	FR1	FR2	FR3	FR1	FR2	FR3
DM1	0.5	0.25	1	0.25	0.5	1	0.5	0.5	0.5	0.5	0.75	0.75
DM2	0.5	1	0.75	0.5	1	1	0.75	0.25	0.5	0.25	0.5	0.75
DM3	0.5	1	0.25	0.25	0.5	0.5	0.5	1	1	0.5	0.25	1
DM4	0.75	0.75	0.5	0.75	0.5	0.25	0.5	0.25	1	0.5	1	0.75
DM5	0.25	0.75	1	0.5	0.5	0.75	0.75	0.5	0.75	0.5	0.25	1

The three-order tensor $\mathcal{A} \in \mathbb{R}^{5 \times 3 \times 4}$ is constructed (5 is the number of stakeholders, 3 is the number of FRs, and 4 is the number of NFRs), and tensor decomposition is applied as in [10] to our constructed tensor \mathcal{A} to produce the reconstructed tensor $\hat{\mathcal{A}}$. First three new matrices are extracted from the constructed tensor as A_1 , A_2 , and A_3 . Then SVD is applied to each matrix A_i . The U matrices are reduced with parameters $c1=$, $c2=$, $c3= 0.6$. The core tensor \mathcal{S} and then reconstructed tensor $\hat{\mathcal{A}}$ should be calculated. Reconstructed tensor $\hat{\mathcal{A}}$ is calculated by the product of the core tensor \mathcal{S} and the mode products of the three matrices $U^{(i)}$ [10].

Furthermore, to generate the primary prioritization list using the tensor concept, the normalized final rank of FRs is calculated and sorted in a descending manner using Eqs. (1) and (2).

$$w_{FR_i} = \sum_{j=1}^{|ST|} \sum_{k=1}^{|NFR|} \hat{\mathcal{A}}(st_j, fr_i, nfr_k) \quad 1 \leq i \leq |FRs| \tag{1}$$

$$NW_{FR_i} = \frac{w_{FR_i}}{|NFR| \times |ST|} \quad 1 \leq i \leq |FRs| \tag{2}$$

Here $|ST|$ means the number of stakeholders, and $|NFR|$ equals the number of nonfunctional requirements. The final rank of our example will be provided as $r_3 > r_2 > r_1$.

The next step of our proposed approach, “generate fuzzy graph of increase/decrease cost of dependency”, is explained here. The cost relation between requirements of our example is provided in Figure 3a by an expert person, and since both the increase and decrease cost dependencies are processed in our proposed approach, two matrices ($\omega^{inc}, \omega^{dec}$) as in Figures 3b and 3c are filled separately from the graph provided in Figure 3a.

We should now apply the fuzzy membership function on the values of the graph edges. The fuzzy membership function for this example has been shown in Figure 4, in which the cost values are mapped to a fuzzy function to produce the fuzzy relation between the requirements in terms of the cost. The strengths of the explicit dependencies provided by the expert person are also defined by membership function $f : R \times R \rightarrow [0, 1]$ in Figure 4. It should be indicated that the fuzzy membership function f can be different for each project.

According to the fuzzy membership function in Figure 4, the fuzzy dependency matrices and fuzzy cost graph of our example are produced as in Figure 5.

Now we analyze the fuzzy graph dependencies. There are two traditional link analyses: centrality and prestige. The former is based on the outlinks and the latter calculates the inlinks while analyzing, and they do not take into account the propagation of links. These expansion rates come from a high cost requirement

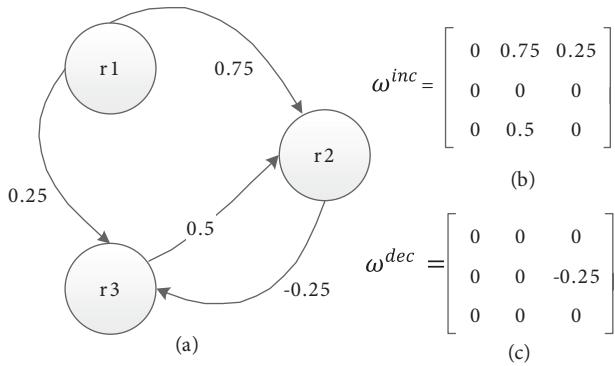


Figure 3. Cost dependency graph and “increase/decrease cost of” dependency matrices of example 1.

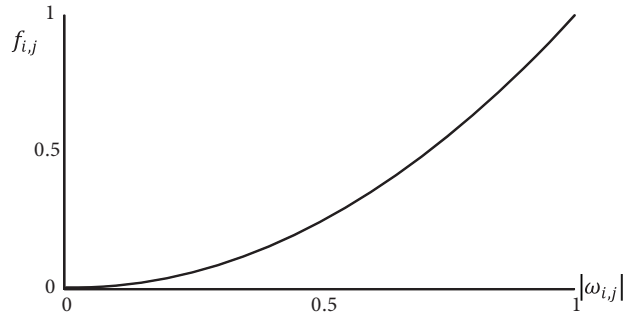


Figure 4. Membership function $f_{i,j}$.

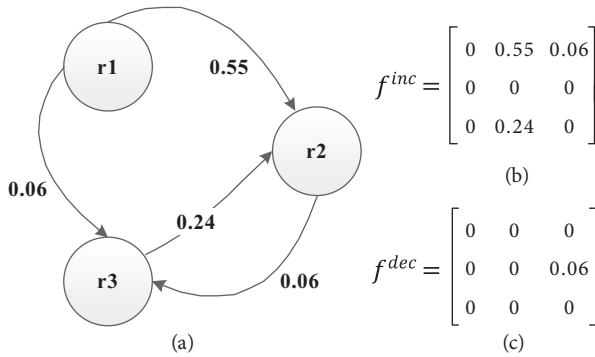


Figure 5. Fuzzy cost graph and fuzzy dependency matrices of example 1.

A dependent on requirement B, which in turn should be assumed as another high cost requirement. In this way, we are able to calculate the real cost influence of each requirement in a fuzzy graph using an appropriate algorithm.

The PageRank algorithm [35] also considers the outlinks and inlinks in its analysis, and we can map our problem (finding the suitable order of cost dependency from the constructed fuzzy cost graph) in the PageRank algorithm with some changes as there are obvious resemblances between the influence of cost dependency analysis and that of web pages. In web page analysis, popular web pages are compared based on their connections [35], leading to the proposition of a weighted PageRank algorithm [36], in which popular pages get higher ranks instead of their ranks being divided among their outlink pages; that is, a rank value is dedicated to every outlink page according to its popularity, measured by the frequency of the afferent and efferent dependencies. Furthermore, an algorithm similar to PageRank is proposed to differentiate afferent and efferent dependencies by integrating the fuzzy distance among requirements in terms of cost dependency. It can analyze the increase and decrease dependencies separately. It can be observed that greater requirement possibility with higher inlink dependencies can be affected by the requirements, which in turn are lower concerning their cost rank. On the other hand, any requirements with higher dependencies on outlinks can mean that influences on more requirements would lead their cost ranks to be higher.

Since the dependencies are analyzed in negative and positive manners, the two dependency orders calculated from the fuzzy cost graph are combined to achieve a conflict-free ordered list. Furthermore, in order to achieve the influence rank of each requirement in the fuzzy cost graph, the following modified formula

is provided based on [25]:

$$Rnk(r_j) = (1 - d) + d \cdot \sum_{r_i \in RS} \left(\frac{Rnk(r_i) \cdot (\alpha - FCW_{(r_i,r_j)}) (\beta + FCW_{(r_j,r_i)})}{Max(\alpha + \beta + 1)} \right) \quad St : 1 < \alpha \leq 2, 0 < \beta \leq 1 \quad (3)$$

Here, $\alpha = 2, \beta = 1$, $Rnk(r_j)$ is the cost rank of requirement r_j , d is a moderating factor to handle cycles in the dependency graph (usually set to 0.85) [35], RS is the group of optional connected requirements, and $FCW_{(r_i,r_j)}$ (fuzzy cost weight) is the total strength of the dependencies from r_i to r_j , as calculated in Eq. (4). In the process, the total strength of m dependencies between r_0 and r_n can be equated with the strength of the strongest dependency among all dependencies between r_0 and r_n , which can be computed in a fuzzy requirement dependency graph of $G = (R, C, f)$ with m dependencies using the following formula:

$$FCW(r_0, r_n) = \bigvee_{i=1}^m \left(\bigwedge_{j=1}^n f(r_{j-1}, r_j) \right) \quad (4)$$

With no implicit cost-related dependency between r_0 and r_n , we have $FCW(r_0, r_n) = f(r_0, r_n)$. The algorithm is similar to the PageRank algorithm as an iterative approach, and the initial values of $Rnk(r_i)$ are set to 1, $r_i \in RS$. The results can be obtained after several iterations [35]. The final rank of requirements with no dependencies is set to NULL.

According to our example in Section 4.1, we compute the cost rank of each requirement. First, we calculate the FCW (r_i, r_j) , $r_i, r_j \in R$, in which, according to Eq. (4), the FCW matrices are calculated as in Figure 6.

$$FCW^{inc}(r_i, r_j) = \begin{bmatrix} 1 & 0.55 & 0.06 \\ 0 & 1 & 0 \\ 0 & 0.24 & 1 \end{bmatrix} \quad FCW^{dec}(r_i, r_j) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.06 \\ 0 & 0 & 1 \end{bmatrix}$$

(a) (b)

Figure 6. Fuzzy cost weight of example 1 requirements (increase mode and decrease mode).

We calculate the value of an element of the matrix as follows:

$$FCW^{inc}(r_1, r_2) = \bigvee ([f(r_1, r_2)], [f(r_1, r_3) \wedge f(r_3, r_2)]) = \bigvee (0.55, [0.06 \wedge 0.24]) = \bigvee (0.55, 0.06) = 0.55$$

By setting the initial value of $Rnk(r_i)$ to 1, the results converge with threshold equal to 1×10^{-6} based on Eq. (3). Finally, the stable rank values of $Rnk(r_i)$ can be sorted as $ord^{inc} : r_1 > r_3 > r_2$. For decrease analysis we do the same process as in positive analysis separately without considering the positive dependencies. Finally, we achieve the decrease cost dependency order of requirements as $ord^{dec} : r_2 > r_3 > r_1$.

According to the steps of our proposed algorithm we should integrate the output of the tensor concept step with the cost ranks provided by the modified PageRank and fuzzy concept. We have two ordered lists including $r_1 > r_3 > r_2$ and $r_2 > r_3 > r_1$, respectively. We merge the two “increase” and “decrease” ordered lists into one conflict-free order using a merge algorithm as in Figure 7 (Algorithm 1).

Input: ord^{inc}, ord^{dec}, R .
Output: ord^{cost} .

1. For each requirement a in R
2. For each requirement b in R
3. If $(ord_a^{dec} > ord_b^{dec})$ and $(ord_a^{inc} \leq ord_b^{inc})$
4. then $ord_a^{cost} < ord_b^{cost}$,
5. Endif
6. If $(ord_a^{dec} < ord_b^{dec})$ and $(ord_a^{inc} \geq ord_b^{inc})$
7. then $ord_a^{cost} > ord_b^{cost}$,
8. Endif
9. If $(ord_a^{dec} == ord_b^{dec})$
10. If $(ord_a^{inc} > ord_b^{inc})$
11. then $ord_a^{cost} > ord_b^{cost}$,
12. Elseif $(ord_a^{inc} < ord_b^{inc})$
13. then $ord_a^{cost} < ord_b^{cost}$,
14. Elseif $(ord_a^{inc} == ord_b^{inc})$
15. then $ord_a^{cost} = ord_b^{cost}$,
16. Endif
17. Endif
18. Endif
19. Endif

Figure 7. Merging algorithm of two increasing and decreasing cost orders.

According to the merging algorithm, the final order of cost dependency from our example is $\{r_1 > r_2, r_1 > r_3, r_2 < r_3\}$; that is, the cost influence of r_1 is higher than r_2 totally and the cost influence of r_1 is also higher than r_3 , but the cost influence of r_2 is lower than r_3 .

The final prioritized list is computed through the integration of the final cost order ($\{r_1 > r_2, r_1 > r_3, r_2 < r_3\}$) and the primary order list provided by tensor decomposition phase ($r_3 > r_2 > r_1$) as shown in Figure 8. The process is continued to check all the requirements and obtain the final prioritized list. Accordingly, the final integrated list of example 1 is $r_2 > r_3 > r_1$, in which both the stakeholders' preferences and increased/decreased cost of dependency among requirements are considered. The produced result displays the accurate results according to the definitions in [37]. The ideal target ranking in the prioritization domain is defined as follows: 1) Ranking based on the opinions of decision makers in their minds. 2) Ranking extracted from a negotiation process among decision makers [37].

5. Controlled experiment

Based on the guidelines of [38], an experiment is designed, and the dependent variables of time consumption, accuracy, and ease of use of the proposed approach, AHP, TOPSIS, and EVOLVE are measured. AHP is

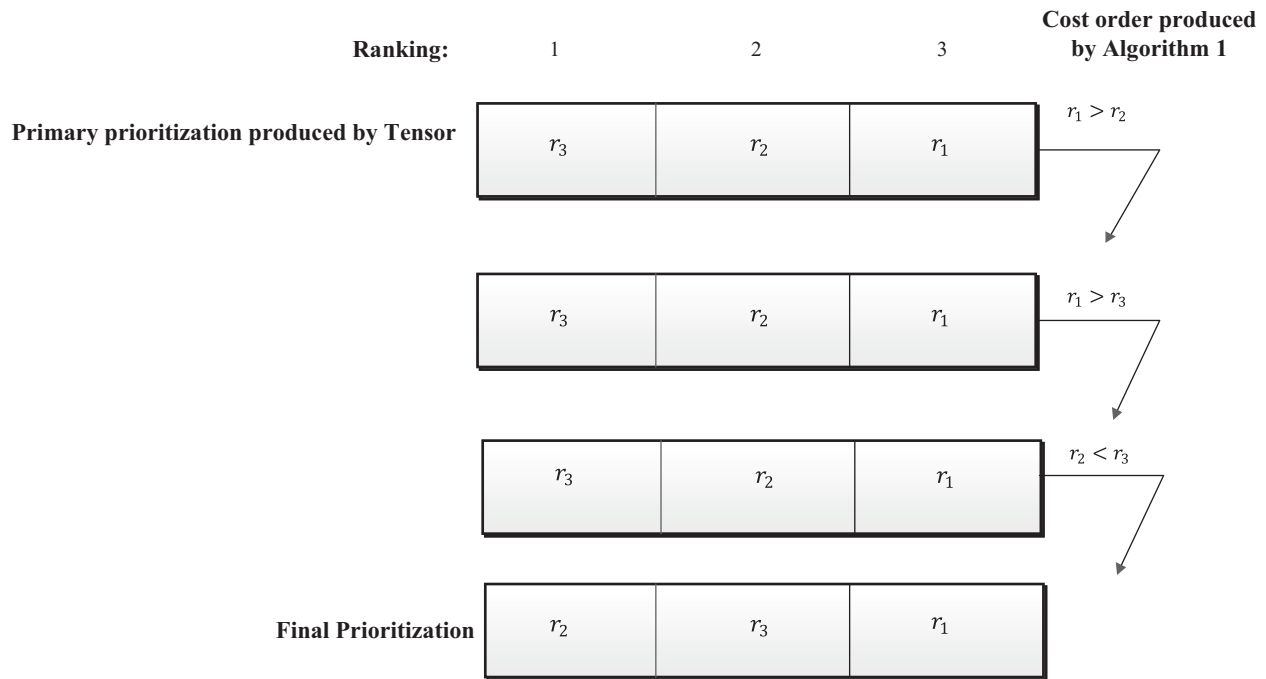


Figure 8. Integration of the final cost order and the primary order list of example 1.

considered as the benchmark approach in several domains and studies [37], as a major industrial requirement prioritization method and prioritization requirement that only observes preferences by stakeholders. TOPSIS has been selected due to its few rank reversals in comparison to multicriteria decision making methods and its wide industrial and academic acceptance and applications [39]. It can also involve interactions among different criteria [40]. EVOLVE has been chosen for its method of typical requirements prioritization, which observes business dependencies among requirements.

As explained in our approach, the “increase/decrease cost of” dependencies among requirements are also considered, so, for our experiment, the fuzzy dependency graph extracted by the expert is used to complete the process of prioritization. The business dependencies among the requirements are also used for the EVOLVE approach.

5.1. Experiment definition

The goals of the experiment are to evaluate:

1. The proposed approach’s effectiveness in terms of time consumption and accurate results compared with AHP, TOPSIS, and EVOLVE.
2. The proposed approach’s ease of use compared to AHP, TOPSIS, and EVOLVE.

Questions: According to goal 1, RQ1 and RQ2 are devised as follows:

RQ1: Is the proposed approach less time-consuming than AHP, TOPSIS, and EVOLVE?

RQ2: Is the result of the proposed approach more accurate than AHP, TOPSIS, and EVOLVE?

For goal 2, RQ3 is constructed as below:

RQ3: Is the proposed approach easier to use than AHP, TOPSIS, and EVOLVE?

Hypothesis formulation: The hypotheses based on the research questions are formulated as follows:

Null hypothesis ($H_{0time-AHP}$) There is no difference between the proposed approach and AHP’s time consumption.

Null hypothesis ($H_{0time-TOPSIS}$) There is no difference between the proposed approach and TOPSIS’s time consumption.

Null hypothesis ($H_{0time-EVOLVE}$) There is no difference between the proposed approach and EVOLVE’s time consumption.

Null hypothesis ($H_{0accuracy-AHP}$) There is no difference between the proposed approach and AHP’s accuracy of the prioritized requirements.

Null hypothesis ($H_{0accuracy-TOPSIS}$) There is no difference between the proposed approach and TOPSIS’s accuracy of the prioritized requirements.

Null hypothesis ($H_{0accuracy-EVOLVE}$) There is no difference between the proposed approach and EVOLVE’s accuracy of the prioritized requirements.

Null hypothesis ($H_{0easeofuse-AHP}$) There is no difference between the proposed approach and AHP’s ease of use.

Null hypothesis ($H_{0easeofuse-TOPSIS}$) There is no difference between the proposed approach and TOPSIS’s ease of use.

Null hypothesis ($H_{0easeofuse-EVOLVE}$) There is no difference between the proposed approach and EVOLVE’s ease of use.

Variables and measures: In the domain of software engineering, the controlled experiments need independent and dependent variables, and our experiment also considers these variables. In accordance with the hypotheses, dependent variables such as actual time consumption, ease of use, and accuracy of the results have been selected. These variables are mostly known as dependent variables in requirements prioritization areas [41]. It is also required to produce accurate results of the prioritization process fast and easily [3], so these properties are considered in this study. Table 2 shows the variables of our controlled experiment.

Table 2. Overview of the controlled experiment.

Independent variables		Our approach, AHP, TOPSIS, and EVOLVE
Dependent variables	Type	Explanation
Actual time consumption	Float	Actual time consumption (s) is recorded from the start of the prioritization process to the end of the manual part of the process by each participant
Accuracy of results	Ordinal	It determines which approach produces a better ordered list and reflects participants’ opinions
Ease of use	Ordinal	Shows how easily the participants work when performing the prioritization process through the provided prioritization approach

The time consumption variable is calculated according to the expended time by every participant in the manual part of the prioritization approaches.

To measure the accuracy variable, we determine which approach (our approach, AHP, TOPSIS, and EVOLVE) produces better sequences and shows the opinions of the participants better in the produced results. The more accurate approach here means that it produces a closer ordered list to the participants’ opinions.

Therefore, two post questionnaires including post questionnaire 1 and post questionnaire 2 are used as well. In post questionnaire 1, the participants are asked to choose a Likert scale value (1–5) [42] to show how accurately the approaches, our approach and the compared methods (AHP, TOPSIS, and EVOLVE), produce the ordered lists. The second post questionnaire, post questionnaire2, has a question with three answers. The participants are required to select one option, our approach or compared methods (AHP, TOPSIS, and EVOLVE), assuming that they are equal in accuracy, to express which one produces more accurate results.

The ease of use variable determines the amount of ease the subjects feel during the prioritization process. Two post questionnaires are also used to collect the subjects' opinions about this variable. Post questionnaire 1 asks subjects (to rank the prioritization approaches (our approach and compared methods (AHP, TOPSIS, and EVOLVE)) according to a Likert scale “How easy is it to use the prioritization approach?”) [42]. The second post questionnaire (“Which approach is easier to use?”) is used to allow the subjects to opt for only one option among “our approach”, “compared methods (AHP, TOPSIS, and EVOLVE)”, and “they are equal”.

5.2. Threats to validity

Four internal, external, construct, and conclusion validity threats were selected to be investigated in this section according to [38].

External validity refers to the outcome generalization of the experiment, and it must be taken into account when students and researchers are participating in the experiment. Some studies do not differentiate between students and professionals [43], whereas some others differentiate between the results provided by students and professionals [44]. Also, the study in [45] showed that students provide highly probable results similar to those provided by practitioners. To handle this threat, research students were selected to participate in the study since, compared to classroom students, resorting to these students raises the probability of acquiring more reliable data [46]. In addition, the participants had enough experience in requirements prioritization and software engineering. However, to improve external validity, our approach needs to be tested in industrial projects.

The fatigue of participants is an example of an internal threat that can influence the findings of a study, so considering the low number of the requirements, the participants' obligatory break for the experiment can reduce the impact of this factor.

Threats to construct validity refer to the interaction between theory and observation as the variable of time is measured in an automatic manner as in [37]. The ease of use and accuracy variables known as dependent variables are based on the participants' perceptions. Therefore, the experience and information of participants can affect the variables [37]. This makes the measurement of the accuracy of prioritization challenging; consequently, we considered the approach that provides prioritization close to the participants' viewpoints as more accurate [37]. This was done via the post questionnaires.

The statistical analyses are related to conclusion validity threats, and to remove this threat, an appropriate statistical analysis should be used to test the hypotheses. It should be mentioned that the limited number of participants could affect the statistical power (20 participants), so it is necessary to perform the experiment with more participants to increase the statistical analysis power.

5.3. Experimental implementation

The experiment was done on a distance learning management system (DLMS) project, and in this experiment, 10 functional and 5 nonfunctional requirements were settled as the objects. Also, 20 PhD candidates and students

experienced in industrial projects and research with good knowledge of requirements prioritization participated in the study. The experiment was divided into three experiments (experiment 1, experiment 2, experiment 3) to compare the three selected methods (AHP, TOPSIS, and EVOLVE) with the proposed approach.

The participants were selected using the convenience sampling technique [38], and the participants used each approach (our approach, AHP, TOPSIS, and EVOLVE) on the same requirements in a random order to avoid any potential bias as a paired comparison design [38].

An Intel Core i5 processor with processing power of 2.30 GHZ, hard disk 500 GB, RAM 4.0 GB, and Windows 7 operating system were the specifications of the system used to perform the experiments. In addition, the approaches were written in MATLAB programming language.

5.4. Experimental results and analyses

The analysis of the experimental results is provided here. The data have been collected through form completion. To accept or reject the hypotheses formulated in Section 5.1, the 5% significance level was used in the statistical analysis using SPSS 16.0.

The Shapiro–Wilk distribution test was applied on the collected values of the experiments to accept or reject the defined hypotheses for the variables of time consumption, ease of use, and accuracy of the results. The output of the test shows that the distribution of the data gathered to test $H_{0time-AHP}$, $H_{0time-TOPSIS}$, and $H_{0time-EVOLVE}$ was normal as the P-values were greater than 0.05. This led to the application of the t-test to accept or reject these hypotheses. The distribution of the data gathered to test $H_{0accuracy-AHP}$, $H_{0accuracy-TOPSIS}$, $H_{0accuracy-EVOLVE}$, $H_{0easeofuse-AHP}$, $H_{0easeofuse-TOPSIS}$, and $H_{0easeofuse-EVOLVE}$ was not normal since each P-value of the data used to test these hypotheses was found to be less than 0.05, so the chi-square test [38] and a nonparametric test, the Mann–Whitney test [47], were used to accept or reject the hypotheses.

Time consumption analysis: The box-plots and details of time consumption by participants are displayed in Figure 9 and Table 3.

Table 3. Average time consumption of prioritizing the requirements.

	The proposed approach	AHP	TOPSIS	EVOLVE
Average time taken (s)	269	1311.25	736.35	613.25

As shown in Table 3, the average time taken by our approach is less than the average time taken by AHP, TOPSIS, and EVOLVE. The median of our approach is also lower than the ones for the compared approaches, as shown in the box-plots sketched in Figure 9.

Table 4 illustrates the t-test results for experiment 1, experiment 2, and experiment 3 and shows that $H_{0time-AHP}$, $H_{0time-TOPSIS}$, and $H_{0time-EVOLVE}$ are to be rejected at a significance level of $\alpha = 0.05$. The results also indicate that, on average, our approach requires less time than the compared approaches. Observing the following considerations, the results are reasonable: AHP works based on pairwise comparisons and passes the consistency check. The participants should rank all of the pairwise requirements using a nine-point Likert scale [12], showing that the process is time-consuming. TOPSIS starts with the initial ranking of alternatives for each criterion, and an absolute scale (e.g., dollars and effort-hours) or a relative scale (e.g., 1–9 Likert scale, 1–3–9, or Fibonacci scale) can be used for scoring. Determining the weight of each criterion in the TOPSIS algorithm also takes time. EVOLVE requires two types of scoring for each requirement, in which the

participants should determine the impact on the business value of the final product and the degree of urgency (time criticality). It uses a five-point Likert scale (from 1 to 5) to score the requirements.

Table 4. Results of t-test based on time variable.

Experiment	Method	Mean	SD	P-value	Hypothesis status
Experiment 1	The proposed approach AHP	2.6900E2 1.3112E3	138.05796 665.51119	0.000	$H_{0time-AHP}$ rejected
Experiment 2	The proposed approach TOPSIS	2.6900E2 7.3635E2	138.05796 170.22563	0.000	$H_{0time-TOPSIS}$ rejected
Experiment 3	The proposed approach EVOLVE	2.6900E2 6.1325E2	138.05796 222.42181	0.000	$H_{0time-EVOLVE}$ rejected

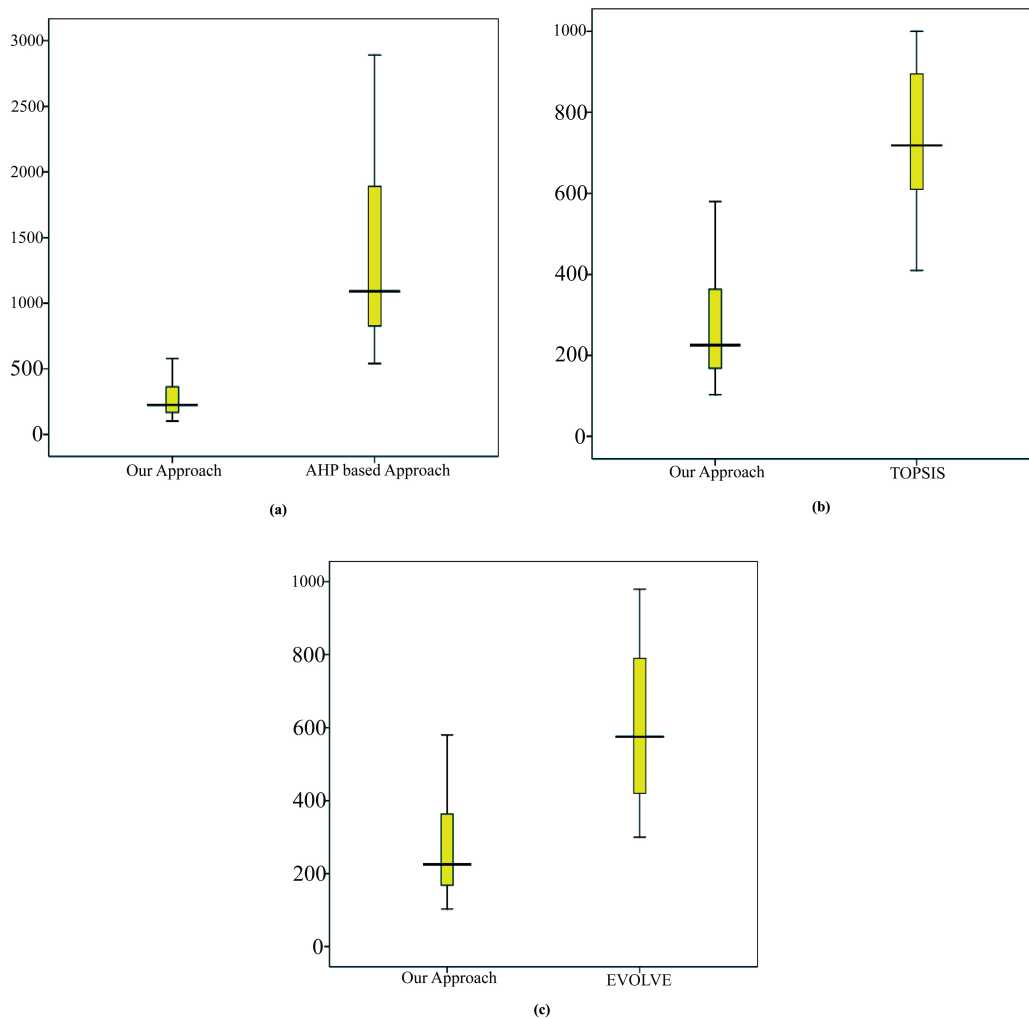


Figure 9. Boxplots of the time consumption.

Ease of use analysis: To measure the ease of use variable, two post questionnaires were used. For the first post questionnaire, a Likert scale was used to make the decision on the value of ease of use in each approach. The results of this post questionnaire study are illustrated in Table 5. The results show that our approach provides the subjects' opinions with a higher degree of ease of use.

The results of the Mann–Whitney test show that the ease of use in our approach is higher than that in AHP, TOPSIS, and EVOLVE, and the ease of use null hypotheses can be rejected as depicted in Table 5, since P-values are less than 0.05. Accordingly, we can conclude that the approach we use works more effectively than AHP, TOPSIS, and EVOLVE.

Table 5. The data gathered from post questionnaire 1 based on ease of use.

Experiment	Approaches	Likert scale					Mean rank	P-value	Hypothesis status
		1	2	3	4	5			
Experiment 1	Proposed approach (number of participants)	-	1	2	3	14	26.62	0.000	$H_{0easeofuse-AHP} \text{ rejected}$
	AHP (number of participants)	-	2	12	3	3	14.38		
Experiment 2	Proposed approach	-	1	2	3	14	25.50	0.004	$H_{0easeofuse-TOPSIS} \text{ rejected}$
	TOPSIS	-	1	8	7	4	15.50		
Experiment 3	Proposed approach	-	1	2	3	14	24.28	0.027	$H_{0easeofuse-EVOLV} \text{ rejected}$
	EVOLVE	-	1	5	8	6	16.72		

In testing the second null hypotheses of ease of use (Table 6), because of the nonnormal distribution of data, the chi-square test [38] was used. The analysis of the chi-square test (Table 6) shows that the difference is statistically significant, as the P-value is less than 0.05 for each experiment; thus, we are able to reject the second null hypotheses of ease of use.

Table 6. The data gathered from post questionnaire 2 based on ease of use.

Experiment	Method	Number of participants	P-value	Hypothesis status
Experiment 1	Proposed approach	15	0.025	$H_{0easeofuse-AHP} \text{ rejected}$
	AHP	5		
	Equally easy	-		
Experiment 2	Proposed approach	13	0.011	$H_{0easeofuse-TOPSIS} \text{ rejected}$
	TOPSIS	3		
	Equally easy	4		
Experiment 3	Proposed approach	12	0.022	$H_{0easeofuse-EVOLV} \text{ rejected}$
	EVOLVE	6		
	Equally easy	2		

These outcomes are sensible under the following considerations. Participants consider the AHP to be more difficult to use than our approach because of the pairwise comparison and passing of the consistency check. In TOPSIS scoring, the criterion-based requirements and making a weight for each criterion make it more difficult to use than our approach. Since EVOLVE requires two types of scoring (impact on business value and urgency) for each requirement, it is more difficult to use it than our approach.

Accuracy of results analysis: An accuracy check using two post questionnaires was performed on the results provided by our proposed approach and the compared approaches (AHP, TOPSIS, and EVOLVE), presented in Section 5.1. The summarized data collected from post questionnaire 1 are displayed in Table 7. The table shows the opinions of the participants, each expressed in terms of the expected accuracy of both our approach and the compared ones.

To determine which approach provides more accuracy compared to the other ones, we tested the participants’ opinions in terms of the null hypothesis ($H_{0accuracy-AHP}$, $H_{0accuracy-TOPSIS}$, and $H_{0accuracy-EVOLV}$)

Table 7. The data collected from post questionnaire 1 based on accuracy.

Experiment	Approaches	Likert scale					Mean rank	P-value	Hypothesis status
		1	2	3	4	5			
Experiment 1	Proposed approach (number of participants)	-	-	4	12	4	24.90	0.010	$H_{0accuracy-AHP}$ rejected
	AHP (number of participants)	-	4	7	8	1	16.10		
Experiment 2	Proposed approach	-	-	4	12	4	24.00	0.040	$H_{0accuracy-TOPSIS}$ rejected
	TOPSIS		3	7	8	2	17.00		
Experiment 3	Proposed approach	-	-	4	12	4	23.90	0.045	$H_{0accuracy-EVOLVE}$ rejected
	EVOLVE		2	8	8	2	17.10		

in Section 5.1). We employed the Mann–Whitney test [47] due to the nonnormal distribution of the data.

From the statistical point of view (Table 7), the difference between the proposed approach and the compared approaches is of great significance. The reason is the smallness of the P-value, which is smaller than 0.05, leading to the rejection of the $H_{0accuracy}$ hypotheses. This indicates higher accuracy, which is produced in the results of our approach in contrast to that of the AHP-based approach, TOPSIS, and EVOLVE.

In addition, the analysis of the second post questionnaire was done to make the decision on the accuracy of the results provided by either one of the approaches (Table 8). The analysis of $H_{0accuracy}$ hypotheses was done employing the chi-square test on the data collected from post questionnaire 2 (see Section 5.1). As the result of the P-value is smaller than 0.05 in the chi-square test of each experiment (see Table 8), the statistical difference is significant and thus we can reject the null hypotheses ($H_{0accuracy-AHP}$, $H_{0accuracy-TOPSIS}$, and $H_{0accuracy-EVOLVE}$). Therefore, it can be concluded that the accuracy of results in our approach is higher than that of the AHP-based approach, TOPSIS, and EVOLVE.

Table 8. The data collected from post questionnaire 2 based on accuracy of results.

Experiment	Method	Number of participants	P-value	Hypotheses status
Experiment 1	Proposed approach	14	0.002	$H_{0accuracy-AHP}$ rejected
	AHP	4		
	Equally accurate	2		
Experiment 2	Proposed approach	12	0.011	$H_{0accuracy-TOPSIS}$ rejected
	TOPSIS	7		
	Equally accurate	1		
Experiment 3	Proposed approach	11	0.019	$H_{0accuracy-EVOLVE}$ rejected
	EVOLVE	8		
	Equally accurate	1		

These outcomes are sensible under the following considerations: The accuracy of the final prioritization list can be improved considering the “increase/decrease cost of” dependencies among the requirements using the fuzzy concept and link analysis in our approach. AHP and TOPSIS only focus on preferences of the stakeholders and do not take into account the requirement dependencies. Therefore, the sequences of their prioritization merely reflect the stakeholders’ preferences.

EVOLVE takes into account both the stakeholders’ preferences and business dependencies (precedence and concurrence dependencies) among the requirements during the prioritization process using an iterative

solution method, which leads to its effective performance. The statistical analyses provided in Table 8 are just based on the participants' opinions according to their usage of approaches and what they expected from them. We know that each approach can be improved to cover more expectations of stakeholders.

6. Discussion

This section discusses the findings and analysis of the experiment. The analysis of time consumption, the first dependent variable, shows that producing the prioritized requirements in our approach is done faster than in the AHP-based approach, TOPSIS, and EVOLVE. Furthermore, the number of decisions, each of which is expected to be made by any participant in our approach, differs from that in the AHP-based approach. For example, consider the prioritization of 10 functional and 5 nonfunctional requirements. In our approach, the participants are required to make $(10 * 5)$ decisions and $((10 * 9)/2) + ((5 * 4)/2)$ decisions in the AHP-based approach. This difference is increased by the increase in the number of requirements [10]. The AHP approach uses the pairwise comparison by participants and checks the consistency of the participants' rank values. This process leads to more time consumption. TOPSIS needs the weight of each criterion and the importance value for each requirement according to that criterion, and EVOLVE requires two types of scoring for each requirement. These can make their applications time-consuming. There are varieties of ranges for the values that are used in decision-making process. In our approach, we opted for a five-point Likert scale, whereas in the AHP-based approach, a nine-point Likert scale was utilized [12]. This can affect the time consumption in the approaches in providing the prioritized requirements. The accuracy of the results in our approach is higher than that in the AHP-based approach and TOPSIS, which provide prioritized requirements by focusing on stakeholders' preferences without considering the requirement dependencies. The prioritization only reflects the stakeholders' preferences, but our approach requires participants' preferences according to the FR and NFR relation. The "increase/decrease cost of" dependency extracted by an expert person using the fuzzy graph and modified page rank algorithm is also used to improve the primary prioritization. In addition, our approach provides the ordered list using three-order analysis, whereas other approaches analyze the preferences apart from each other (FRs and NFRs) or in a two-way analysis. This results in increasing the accuracy of the produced prioritization using our approach [10]. By considering the business dependencies among the requirements, EVOLVE acts well in terms of accuracy of the results. Needless to say, the number of experiment requirements (10 FRs and 5 NFRs) was rather small. The time being spent on doing the experiment should be reasonable. We took this into account as mentioned in [3]. This factor forced us to choose a smaller number of requirements for conducting the experiment, whereas industrial projects have more requirements. Therefore, the findings of this experiment are useful when prioritization is needed for subsets of requirements in large-scale systems. In this regard, as a suggestion for future studies, our approach can be evaluated for an industrial case study. Our approach requires the identification and consideration of the dependencies among the requirements during its process. This task is difficult, but it is a necessary one in the process of requirement analysis. Therefore, we can consider that the dependency identification task has successfully been fulfilled. The approach provided in this paper only supports the "increase/decrease cost of" dependency type. It is important to improve it by considering more dependency types.

Some explanations on the reduction parameters of the tensor and the scalability of the tensor are discussed in [10].

7. Conclusion

Many factors are involved in the prioritization of the requirements. We proposed an approach for requirements prioritization in this paper that considers not only the importance of requirements in a project, but also the dependencies among them in the process. Multidimensional analysis is provided using a three-order tensor and Tucker decomposition since the tensor algebra can reveal the relations among its entities in an effective manner. Also, the fuzzy graph algebra and link analysis have been used to analyze the requirements dependencies. In order to compare the approaches with each other, we have employed a controlled experiment, and the results show the superiority of our approach regarding time consumption, ease of use, and accuracy. Future works can involve considerations such as the weight of stakeholders and handling other types of dependencies. Additionally, future research could be conducted on such controlled experiments in which industrial case studies are considered. By doing so, one can investigate how similar the results are to our findings. In a similar way, one can investigate how great numbers of requirements and stakeholders can be handled in our approach.

References

- [1] da Silva FQ, Santos AL, Soares S, França ACC, Monteiro CV et al. Six years of systematic literature reviews in software engineering: an updated tertiary study. *Information and Software Technology* 2011; 53 (9): 899-913. doi: 10.1016/j.infsof.2011.04.004
- [2] Wiegers K, Beatty J. *Software Requirements*. Seattle, WA, USA: Microsoft Press, 2013.
- [3] Karlsson J, Ryan K. A cost-value approach for prioritizing requirements. *IEEE Software* 1997; 14 (5): 67-74. doi: 10.1109/52.605933
- [4] Pohl K. *Requirements Engineering: Fundamentals, Principles, and Techniques*. Berlin, Germany: Springer-Verlag, 2010.
- [5] Perini A, Susi A, Avesani P. A machine learning approach to software requirements prioritization. *IEEE Transactions on Software Engineering* 2013; 39 (4): 445-461. doi: 10.1109/TSE.2012.52
- [6] Achimugu P, Selamat A, Ibrahim R, Mahrin MN. A systematic literature review of software requirements prioritization research. *Information and Software Technology* 2014; 56 (6): 568-585. doi: 10.1016/j.infsof.2014.02.001
- [7] Fellir F, Nafil K, Touahni R. System requirements prioritization based on AHP. In: *IEEE 2014 Third International Colloquium in Information Science and Technology*; Tetouan, Morocco; 2014. pp. 163-167.
- [8] Sureka A. Requirements prioritization and next-release problem under non-additive value conditions. In: *IEEE 2014 23rd Australian Software Engineering Conference*; Milsons Point, Australia; 2014. pp. 120-123.
- [9] Kukreja N, Payyavula SS, Boehm B, Padmanabhuni S. Value-based requirements prioritization: usage experiences. *Procedia Computer Science* 2013; 16: 806-813. doi: 10.1016/j.procs.2013.01.084
- [10] Misaghian N, Motameni H. An approach for requirements prioritization based on tensor decomposition. *Requirements Engineering* 2018; 23 (2): 169-188. doi: 10.1007/s00766-016-0262-6
- [11] Hwang CL, Yoon K. *Multiple Attribute Decision Making*. Berlin, Germany: Springer-Verlag, 1981.
- [12] Saaty TL. *Fundamentals of Decision Making and Priority Theory with the Analytic Hierarchy Process*. Pittsburgh, PA, USA: RWS Publications, 2000.
- [13] Dahlstedt AG, Persson A. Requirements interdependencies-moulding the state of research into a research agenda. In: *2003 Ninth International Workshop on Requirements Engineering: Foundation for Software Quality*; Klagenfurt, Austria; 2003. pp. 55-64.
- [14] Wang J, Li J, Wang Q, Zhang H, Wang H. A simulation approach for impact analysis of requirement volatility considering dependency change. In: *2012 International Working Conference on Requirements Engineering: Foundation for Software Quality*; Essen, Germany; 2012. pp. 59-76.

- [15] Rosenfeld A. Fuzzy graphs. In: Zadeh LA, Fu KS, Tanaka K, Shimura M (editors). *Fuzzy Sets and Their Applications to Cognitive and Decision Processes*. Cambridge, MA, USA: Academic Press, 1975. pp. 77-95.
- [16] Riegel N, Doerr J. A systematic literature review of requirements prioritization criteria. In: 2015 International Working Conference on Requirements Engineering: Foundation for Software Quality; Essen, Germany; 2015. pp. 300-317.
- [17] Pitangueira AM, Maciel RSP, Barros M. Software requirements selection and prioritization using SBSE approaches: a systematic review and mapping of the literature. *Journal of Systems and Software* 2015; 103: 267-280. doi: 10.1016/j.jss.2014.09.038
- [18] Yu ES. Towards modelling and reasoning support for early-phase requirements engineering. In: IEEE 1997 3rd International Symposium on Requirements Engineering; Annapolis, MD, USA; 1997. pp. 226-235.
- [19] Sadiq M, Jain S. Stakeholder identification method in goal oriented requirements elicitation process. In: IEEE 2014 5th International Workshop on Requirements Prioritization and Communication; Karlskrona, Sweden; 2014. pp. 25-33.
- [20] Dahlstedt ÅG, Persson A. Requirements interdependencies: state of the art and future challenges. In: Aurum A, Wohlin C (editors). *Engineering and Managing Software Requirements*. Berlin, Germany: Springer-Verlag, 2005. pp. 95-116.
- [21] Zhang H, Li J, Zhu L, Jeffery R, Liu Y et al. Investigating dependencies in software requirements for change propagation analysis. *Information and Software Technology* 2014; 56 (1): 40-53. doi: 10.1016/j.infsof.2013.07.001
- [22] Fernández DM, Mund J, Femmer H, Vetrò A. In quest for requirements engineering oracles: dependent variables and measurements for (good) RE. In: ACM 2014 18th International Conference on Evaluation and Assessment in Software Engineering; London, UK; 2014. pp. 21-31.
- [23] McZara J, Sarkani S, Holzer T, Eveleigh T. Software requirements prioritization and selection using linguistic tools and constraint solvers—a controlled experiment. *Empirical Software Engineering* 2015; 20 (6): 1721-1761. doi: 10.1007/s10664-014-9334-8
- [24] Greer D, Ruhe G. Software release planning: an evolutionary and iterative approach. *Information and Software Technology* 2004; 46 (4): 243-253. doi: 10.1016/j.infsof.2003.07.002
- [25] Shao F, Peng R, Lai H, Wang B. DRank: A semi-automated requirements prioritization method based on preferences and dependencies. *Journal of Systems and Software* 2017; 126: 141-156. doi: 10.1016/j.jss.2016.09.043
- [26] Mougouei D, Powers DM. Modeling and selection of interdependent software requirements using fuzzy graphs. *International Journal of Fuzzy Systems* 2017; 19 (6): 1812-1828. doi: 10.1007/s40815-017-0364-4
- [27] Carlshamre P, Sandahl K, Lindvall M, Regnell B, och Dag JN. An industrial survey of requirements interdependencies in software product release planning. In: IEEE 2001 Fifth International Symposium on Requirements Engineering; Toronto, Canada; 2001. pp. 84-91.
- [28] Zhang WR. Bipolar fuzzy sets and relations: a computational framework for cognitive modeling and multiagent decision analysis. In: IEEE 1994 First International Joint Conference of the North American Fuzzy Information Processing Society Biannual Conference; San Antonio, TX, USA; 1994. pp. 305-309.
- [29] Mathew S, Sunitha M. Strongest strong cycles and θ -fuzzy graphs. *IEEE Transactions on Fuzzy Systems* 2013; 21 (6): 1096-1104. doi: 10.1109/TFUZZ.2013.2243154
- [30] Mordeson JN, Nair PS. Applications of fuzzy graphs. In: Mordeson JN, Nair PS (editors). *Fuzzy Graphs and Fuzzy Hypergraphs*. Heidelberg, Germany: Physica, 2000, pp. 83-133.
- [31] Zadeh LA. Fuzzy sets. In: Klir GJ, Yuan B (editors). *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems*. River Edge, NJ, USA: World Scientific Publishing, 1996, pp. 394-432.
- [32] Klir GJ, Folger TA. *Fuzzy Sets, Uncertainty, and Information*. Hoboken, NJ, USA: Prentice Hall, 1988.

- [33] De Lathauwer L, De Moor B, Vandewalle J. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications* 2000; 21 (4): 1253-1278. doi: 10.1137/S0895479896305696
- [34] Tucker LR. Some mathematical notes on three-mode factor analysis. *Psychometrika* 1966; 31 (3): 279-311. doi: 10.1007/BF02289464
- [35] Page L, Brin S, Motwani R, Winograd T. *The PageRank citation ranking: bringing order to the web*. Stanford, CA, USA: Stanford InfoLab, 1999.
- [36] Xing W, Ghorbani A. Weighted pagerank algorithm. In: 2004 Second Annual Conference on Communication Networks and Services Research; Washington, DC, USA; 2004. pp. 305-314.
- [37] Perini A, Ricca F, Susi A. Tool-supported requirements prioritization: comparing the AHP and CBRank methods. *Information and Software Technology* 2009; 51 (6): 1021-1032. doi: 10.1016/j.infsof.2008.12.001
- [38] Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B. *Experimentation in Software Engineering*. Berlin, Germany: Springer-Verlag, 2012.
- [39] Shih HS, Shyr HJ, Lee ES. An extension of TOPSIS for group decision making. *Mathematical and Computer Modeling* 2007; 45 (7-8): 801-813. doi: 10.1016/j.mcm.2006.03.023
- [40] Govindan K, Khodaverdi R, Jafarian A. A fuzzy multi criteria approach for measuring sustainability performance of a supplier based on triple bottom line approach. *Journal of Cleaner Production* 2013; 47: 345-354. doi: 10.1016/j.jclepro.2012.04.014
- [41] Pergher M, Rossi B. Requirements prioritization in software engineering: a systematic mapping study. In: *IEEE 2013 Third International Workshop on Empirical Requirements Engineering*; Rio de Janeiro, Brazil; 2013. pp. 40-44.
- [42] Likert R. A technique for the measurement of attitudes. *Archives of Psychology* 1932; 22: 5-55.
- [43] Svahnberg M, Aurum A, Wohlin C. Using students as subjects—an empirical evaluation. In: *ACM-IEEE 2008 Second international Symposium on Empirical Software Engineering and Measurement*; Kaiserslautern, Germany; 2008. pp. 288-290.
- [44] Berander P. Using students as subjects in requirements prioritization. In: *2004 International Symposium on Empirical Software Engineering*; Redondo Beach, CA, USA; 2004. pp. 167-176.
- [45] Tichy WF. Hints for reviewing empirical work in software engineering. *Empirical Software Engineering* 2000; 5 (4): 309-312. doi: 10.1023/A:1009844119158
- [46] Danesh AS, Ahmad R. Study of prioritization techniques using students as subjects. In: *2009 International Conference on Information Management and Engineering*; Kuala Lumpur, Malaysia; 2009. pp. 390-394.
- [47] Siegal S. *Nonparametric Statistics for the Behavioral Sciences*. New York, NY, US: McGraw-Hill, 1956.