Dependency Graph (QDG) . This method models the relationship among enterprises and then evaluates the candidate enterprises in the service selection process. QDG Method uses two algorithms to obtain the optimal workflow. The first algorithm is applied to create quality dependency graph and the second algorithm is used to get the needed services.

3. The suggested method

The suggested method for determining the optimal workflow is composed of five steps which have been shown in figure 1.

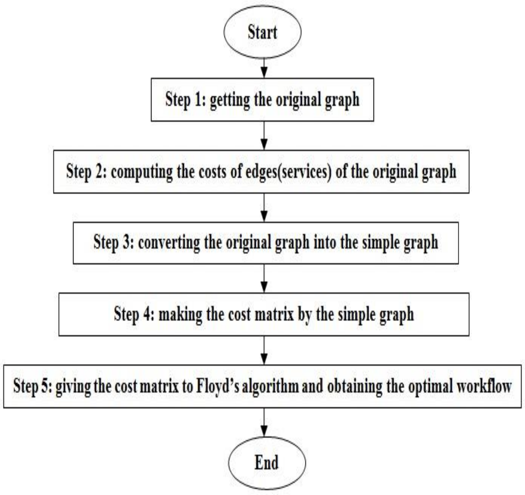

Figure 1. The steps of the design of the optimal workflow

In sequel, each of five phases will separately be discussed. Note that the very simple examples are used to easily understand the stages of the proposed method.

Step 1- The original graph is a directed graph which consists of initial node(initial state), intermediate nodes, goal node (goal state), and several edges which play the role of services.

Initial node only includes inputs which are related to some of services. The intermediate nodes include the inputs and outputs which belong to services. It means that existing parameters in the intermediate nodes, on one hand, are outputs of one service or more, and on the other hand, are taken into account as inputs of other services. The goal node only includes outputs of services. For example, figure 2 indicates a graph which has four nodes. Initial node and goal node are named initial state and goal state respectively. Also, nodes **A** and **B** are the intermediate nodes. Inputs of services S1,S2,S3, and S4 are gotten from initial node and their outputs are placed in nodes **A** and **B** and given to services S5 and S6 as inputs. At last, outputs of S5 and S6 are inserted into goal state.

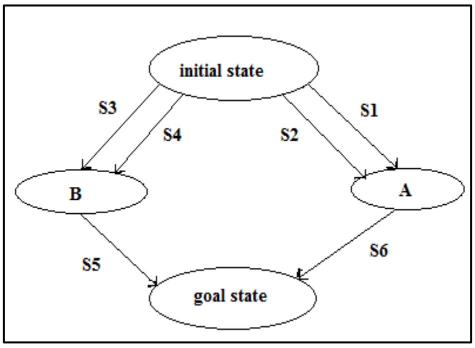

Figure 2. The original graph

Step 2- In this step, the cost of service is computed through the parameters of quality of service. Quality of any service is stated by the criterions such as the access time, the execution time, distance, and so on. Each of criterions has a weight which indicates the value of importance of criterion. The following formula that is based on the criterions of quality of service and their weight, computes the cost of service (C(S)).

$C(S)=\sum_{i=1}^{n} wc_i \times v(c_i)$ (1)

In equation (1) , $\mathbf{c_i}$ is criterion, $\mathbf{v(c_i)}$ is the value of c_i for service **S**, $\mathbf{wc_i}$ is the weight of criterion $\mathbf{c_i}$, and **C(S)** shows the cost of service **S**.

For example, supposing, there are two criterions of quality for an existing service in graph shown in figure 2 (e.g. p and q), the cost of any service is computed by:

$C(S_i)= wq\times v_i\,(q)+wp\times v_i\,(p) \quad 1\leq\ i\ \leq 6$ (2)

Where **q** and **p** are criterions, $\mathbf{v_i}$ **(q)** and $\mathbf{v_i}$ **(p)** are the values of **q** and **p** that belong to service $\mathbf{S_i}$, **wq** and **wp** are weight of **q** and weight of **p** respectively, and $\mathbf{C(S_i)}$ shows the cost of service $\mathbf{S_i}$.

Step 3- In this stage, the original graph must be converted into the simple graph. The simple graph is a graph having up to a directed edge between any two nodes, such as '**C**' and '**D**', where the edge is started from '**C**' and ended to '**D**' (i.e. an edge from **C** to **D**).

Considering what was said above, for converting the graph of figure 2 into the simple graph, any two nodes between which two edges or more exist must be specified. Afterwards, all edges between two nodes except the lowest-cost edge must be eliminated. Supposing, in the original graph of figure 2, the costs of S1and S3 are the lowest, the simple graph is shown as follows:

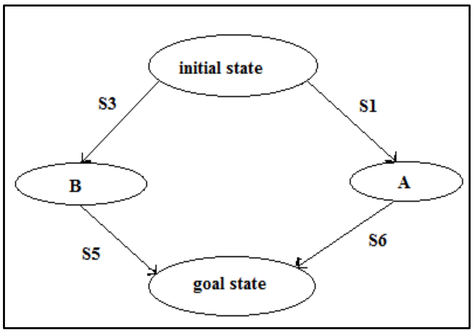

Figure 3. The simple graph

The cause of selecting the lowest-cost edge (service) is that an optimal workflow must contain the lowest-cost edges (services), otherwise, the workflow is never optimal.

Step 4- Now, the cost matrix is obtained by the simple graph. Each of elements of the cost matrix indicates the cost of an edge(service). If there is an edge(service) from a node to another in the simple graph then the cost of edge(service) is inserted into the cost matrix. If there are no edges(services) from a node to another then '∞' is inserted into the cost matrix. Considering what was said above, the cost matrix of the simple graph shown in figure 3 is as:

$$\text{Matrix}=\begin{bmatrix} \infty & C(S1) & C(S3) & \infty \\ \infty & \infty & \infty & C(S6) \\ \infty & \infty & \infty & C(S5) \\ \infty & \infty & \infty & \infty \end{bmatrix}$$

Step 5- In this step, role of floyd's algorithm is understood. In this paper, the proposed idea for the design of the optimal workflow is based on floyd's algorithm. Floyd's algorithm is an algorithm which can show the lowest-cost path between two nodes. Pseudo code of floyd's algorithm is as follows:

```
Void Floyd ( const number m[ ][ ] , number D[ ][ ] ,
{                 index path[][],int n)
 Index i , j, k ;
  For ( i=1; i <= n ; i++)
      For ( j=1; j <= n ; j++)
             Path[i][j]=0;
  D = m;
  For ( k=1 ; k <= n ;k++)
      For ( i=1; i <= n ;i++)
          For ( j=1; j <= n ;j++)
              If (D[i][k] + D[k][j] < D[i][j])
                {
                   D[i][j] = D[i][k] + D[k][j] ;
                   path[i][j]=k;
                }
 }
```

There are four inputs in the above pseudo code. The first input is matrix **m** which indicates a graph. It means that matrix **m** shows a graph. The second input is matrix **D** which is used for finding the lowest-cost path between two nodes. The mentioned matrix is initially equal to matrix **m** and its elements gradually change during the run of floyd's algorithm. The third input is a matrix named **path** which shows the lowest-cost path between any two nodes. The last input is number **n** which shows size of matrices. The cost matrix is delivered to floyd's algorithm as input **m**. After the algorithm was executed, the optimal workflow is obtained by matrix **path** as follows:

Suppose that matrix 'path' is as:

$$\text{Path}=\begin{bmatrix} 0 & 0 & 2 & 3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

In the above matrix, rows 1,2,3, and 4 (from up to down), show initial node, node 2, node 3, and goal node respectively. Also, columns 1,2,3,and 4 (from left to right), indicate initial node, node 2, node 3, and goal node respectively. The optimal workflow is equal to a set of the lowest-cost edges(services) between initial node and goal node.

Considering matrix 'path', it is understood that path[1,4]=3. It means when the transition is done from initial node to goal node via node 3, the cost of the transition becomes lower. Figure 4 shows the above explanation.

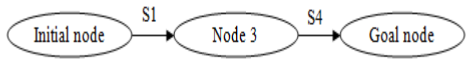

Figure 4. Node 3 between initial node and goal node

On the other hand, path[1,3]=2 shows that if the transition is done from initial node to node 3 through node 2 then the cost of the transition will become lower. In other words, S1 is replaced by S2 and S3. It has been drawn in figure 5.

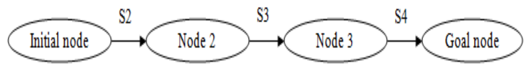

Figure 5. Node 2 between initial node and node 3

Now, edges(services) S2,S3, and S4 make the lowest-cost path between initial node and goal node. Therefore, the optimal workflow is {S2,S3,S4}.

4. Experiment

In this expriment, the proposed method is compared with an analogous method named multi-criteria search algorithm (available at http://www.eng.utoledo.edu/~gserpen/Publications/ENASE%202010%20Manuscript.pdf) to be proved that its mechanism is correct in recognition of the optimal workflow. Multi-criteria search algorithm is a method that works like the proposed method. This method considers the services and their input/output as edges and nodes of a graph. It uses many criterions of quality to compute cost of the transition from initial node to goal node as the proposed method computes the cost of the transition. An important difference between Multi-criteria search algorithm and the proposed method is that Multi-criteria search algorithm is relied on two algorithms whereas the proposed method is based on an algorithm. Table 1 shows the tools by which the experiment is done.

Table 1. The tools used in experiment

Tool	Type
Processor	Intel Core 2 Duo (Clock Speed : 2.40 GHZ)
Cash	4 MB
Memory	2 GB
Operating System	Windows 7 professional
Language of Implementation	Visual studio 2008(C#)

4.1 Description

Two samples are considered in experiment. Each of samples gets its special information.

The information consists of three parts. The first part is related to initial state and goal state. The second part contains the criterions of quality of service, and their weights . The last part includes the services, their inputs and outputs, and the values of criterions.

There are two criterions of quality for each of samples.Two criterions are named **p** and **q**. Criterion **p** states the execution time of service and criterion **q** states the access time of service. Both criterions are stated in second.The information of sample 1 is as follows:

initial state is caracter **a** and goal state is caracter **g**. Also, weight of criterion **p** is **0.25** and weight of criterion **q** is **0.5**. The services, their inputs/outputs, and the values of criterions have been shown in table 2.

Table 2. The services of sample 1

Service name	Input(s)	output	p(second)	q(second)
S1	a	c	0.5	0.5
S2	c	d	0.25	0.5
S3	c	e	0.5	0.5
S4	a	f	0.25	0.5
S5	f	h	0.25	0.25
S6	f	l	0.25	1
S7	l	g	0.25	0.5

The original graph of sample 1 has been drawn in figure 6.

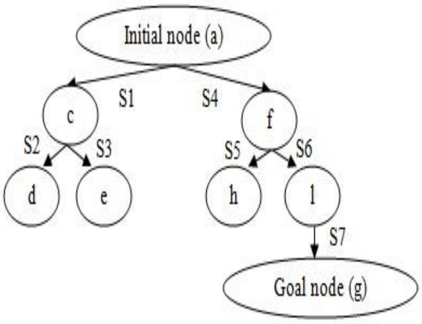

Figure 6. The original graph of sample 1

Also, The information of sample 2 is as follows:
initial state is composed of caracters **a** and **b** and goal state is caracter **g**. weights of **p** and **q** are equal to **1**. The services, their inputs/outputs, and the values of criterions have been shown in table 3.

Table 3. The services of sample 2

Service name	Input(s)	output	p(second)	q(second)
S1	a,b	c	1	1
S2	a,b	d	1	1
S3	d	h	1	1
S4	a,b	e	1	1
S5	e	g	1	1
S6	e	g	1	1
S7	a,b	g	1	1

The original graph of sample 2 has been drawn in figure 7.

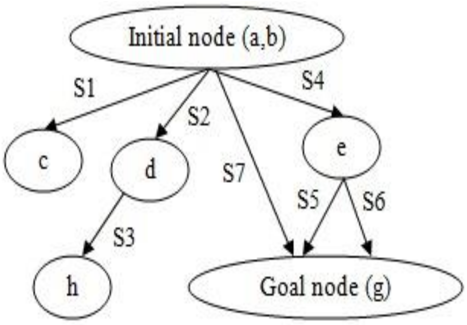

Figure 7. The original graph of sample 2

Now,initially, the optimal workflows of samples 1 and 2 are manually computed. Afterwards the optimal workflows of samples are automatically computed by both the proposed method and multi-criteria search algorithm . In this stage, the existing optimal workflows in both states (i.e. manually and automatically) should be compared with each other. If the results of both states are the same then it is concluded that the proposed method correctly works.

4.2 Results

Initially, considering the original graphs of samples (Figures 6 and 7), the optimal workflow of each of samples is manually computed.

As observed in the original graph of sample 1(Figure 6), the only workflow between initial node and goal node is **{S4,S6,S7}**. Therefore, the optimal workflow is equal to **{S4,S6,S7}**.

Note that cost of **{S4,S6,S7}** is sum of costs of services **S4**,**S6**, and **S7**. It means:

Cost of {S4,S6,S7}=C(S4)+C(S6)+C(S7) =(0.25×0.25+0.5×0.5)+ (0.25×0.25+1×0.5)+ (0.25×0.25+0.5×0.5)=1.1875 (3)

There are three workflows in the original graph of sample 2 (Figure 7). The first workflow is equal to **{S7}**. The second is **{S4,S5}** and The third is **{S4,S6}**. As said before, the optimal workflow is composed of the lowest-cost services. Therefore, for determining the optimal workflow, the cost of each of the mentioned workflows must initially be computed by equation (1) as follows:

Cost of {S7}=C(S7)=1×1+1×1=2 ,
Cost of {S4,S5}=C(S4)+C(S5)=(1×1+1×1)+ (1×1+1×1)=4 ,
Cost of {S4,S6}=C(S4)+C(S6)=(1×1+1×1)+ (1×1+1×1)=4
min{ Cost of {S7}, Cost of {S4,S5}, Cost of {S4,S6}}=min{2,4,4}=2 (5)

Considering (5), **{S7}** is the lowest-cost workflow, therefore the optimal workflow of sample 2 is equal to **{S7}.**

Now, the original graphs of samples are given to both the proposed method and multi-criteria search algorithm to automatically compute the optimal workflow. The results have been shown in Table 4.

Table 4. Automatic computation of the optimal workflow by both methods

	Optimal workflow by the proposed method (automatically)	Optimal workflow by the multi-criteria search algorithm (automatically)
Sample 1	S4,S6,S7	S4,S6,S7
Sample 2	S7	S7

Considering table 4, the results of the methods are the same as the results of manual manner.
It shows that both methods properly work. On the other hand, it is proved that the proposed method can correctly recognizes the optimal workflow because the optimal workflows of both methods are the same for both samples.

5. Conclusion

When the number of services is so much, the manual design of the optimal workflow is a hard and time-consuming work. Because of it, a lot of methods were created to automatically compute the optimal workflow. In this paper, a novel method has been proposed that its structure is based on floyd's algorithm. This method automatically obtains the optimal workflow by making an original graph from services. In the mentioned graph, services play the role of edges and their inputs/outputs are as nodes of graph. The results showed that this method works well.

References

[1]. Chien-Hsiang Lee, San-Yih Hwang," WS-Data Model: A Data Model for Web Services Composition and Optimization", Journal of the Association for Information Systems, Vol. 6, No. 2, 2014,pp.20-47

[2].Seog.chan oh,Dongwon Lee,Soundar R.T.Kumara," WSPR: An Effective and Scalable Web Service Composition Algorithm", International Journal of Web Services Research,Vol.4,No.1, 2007 ,pp.1-22

[3]. Gerardo Canfora, Massimiliano Di Penta, Raffaele Esposito, Maria Luisa Villani," A Framework for QoS-Aware Binding and Re-Binding of Composite Web Services ",Elsevier,Journal of systems and software,vol.81,issue 10,2008,pp.1754-1769.

[4]. M. Sathya, M. Swarnamugi, P. Dhavachelvan & G. Sureshkumar," Evaluation of QoS Based Web- Service Selection Techniques for Service Composition", International Journal of Software Engineering (IJSE), Vol.1, Issue 5,2011,pp.73-124