Service selection method with multiple probabilistic QoS attributes using probabilistic AHP

Ireneusz Jozwiak, Michal Kedziora and Aleksander Marianski

Wroclaw University of Technology, Department of computer science and management, Poland

Summary
Nowadays web service selection is important task in companies. Facing uncertainty in engineering is also a common problem. Our goal was to develop method for solving decision problems including probabilistic uncertainties. The paper presents modified analytic hierarchy process (AHP), where quantitative criterion values are random variables. The method was used to solve service selection problem with probabilistic uncertainties. The article consists literature overview, problem formulation, new service selection method, descriptive example and conclusions. Six different ranking calculation methods were used and we performed an experiment focused on showing differences between them. The paper also discusses critical hurdle and saturation effect.

Keywords:
service selection problem, service oriented architecture, QoS constraints, probabilistic QoS, probabilistic AHP

1. Introduction

Service-Oriented Architecture (SOA) is a widely used concept to connect business services using unified formats and frameworks. From the beginning of SOA development, it was considered as the next generation of computer systems evolution. Right now, many small companies and big enterprises offer their web services to fulfil users' requirements. Question which arose early in SOA development was how to select services which fulfil the desired Quality of Service (QoS).

One of the first articles regarding the subject of QoS describes QoS issues in web services [8]. The paper describes example of Service Level Agreement (SLA) with conditions based on QoS constraints. QoS attributes can be physical quantities such as response time, throughput or availability and can be also defined by some document such as security rules. From the user's perspective, some numerical values, like availability, should be maximized and some, like response time, should be minimized. Every service has also the cost of its usage.

The problem arises when we have more than one service fulfilling the same functionality. To avoid unnecessary cost we need to choose only one of them. This problem is called Service Selection Problem (SSP). There are many publications regarding the subject and many different approaches. Basic approach includes assumption that every QoS parameter is a number. To select the best service one can use genetic algorithm [14], combinatorial algorithm [17], graph modelling [18], AHP method with fuzzy logic [2] and some other heuristic methods [6,9,19]. Taxonomy of web services selection approaches can be found in [1]. Another approach includes using probability function as QoS parameter. One can use probability mass function (PMF) [3,4] or probability density function (PDF) [20,21]. Several other approaches include an evidence-based scheme [15], trust and reputation awareness [16] and stochastic QoS attributes [13].

In our approach, we used modified Analytic Hierarchical Process (AHP) [12] combined with obtaining probability distribution function of QoS parameters. AHP is a well known algorithm. It is usually used when we have multiple criterias and have knowledge which of these criterias are more important to us. Combining AHP with service selection problem has been succesfully done recently [10]. However, there are no publications regarding the topic of service selection with probabilistic QoS attributes using AHP method.

In our work, we used modified AHP method suited to random variables instead of numbers. Idea is similar to [7], however we put random variables only on QoS attributes and do not assume that we have specific probability distribution. To obtain resulting probability distribution function we used convolution algorithm and tool described in [5].

The remainder of the article is organized as follows. In section 2 we formulate problem. In section 3 we show our method to solve the problem with possible variations. In section 4 we present example calculations using our method. Finally, we conclude this study in section 5.

2. Introduction

Service composition is a task in which one need to design workflow of web services. Each of these web services have different functionalities. Web services in a workflow are called abstract web services and their concrete

Manuscript received February 5, 2018
Manuscript revised February 20, 2018

implementation with specific endpoint are called concrete web services [14]. When the workflow is designed, we can ask web service broker or web service discovery tool to find services which match our criteria. Result of that operation should be concrete endpoints of desired concrete web services and their QoS.

Service selection is a task performed usually by end-user application. When we have n concrete implementations of abstract web service W, our task is to select one concrete implementation in order to use it in web service composition. In our case, we assume that obtained QoS are in the form of probability distribution function. Those PDFs can be obtained using method described in [20]. We also assume that every concrete web service has m QoS attributes. These attributes can be for example: response time, reputation, reliability, availability.

Input:

- Abstract web service W and its concrete implementations given by web service broker $\{w_1, w_2, \ldots, w_n\}$
- QoS attributes $\{a_1, a_2, \ldots, a_n\}$
- Pairwise comparison between QoS attributes – the relative importance of one criterion over another. One can use AHP scale from 1 - equal to 9 – extreme importance. It should be preference matrix with elements b_{ij} matching following rules: $1 \leq i, j \leq m, b_{i,j} = b_{i,j}, 0 < \leq 9$
- QoS attributes values as PDFs of random variables. PDFs are defined for each attribute a_j and for each concrete web service w_i: $v_{ij}(x)$, where $1 \leq i \leq n$ and $1 \leq j \leq m$. QoS random variables we denote as $V_{ij} \geq 0$.
- QoS optimization type $c^i \in \{maximize, minimize\}$, $1 \leq i \leq m$
- Output:
- Ranking of web services $rank(w_i)$

3. Service selection with AHP method

Our algorithm uses standard AHP method to determine priority vector. To do so we use pairwise comparison from the input variables and determine eigenvector from given matrix. This eigenvector will be used to calculate weighted arithmetic mean of QoS random variables.

Next step is to obtain the same optimization type. It is important because in AHP we generally want to maximize the rank.

Then we transform and normalize random variables. Every random variable V_{ij} should be transformed in a way that significance of changes is the same for every random variable. For example, when we have two quantities: first the width of square and second the surface of the square. Changes of the surface depend quadratically on changes of the width. That is why we need proper transformation. Normalization in our case is a process of adjusting random variables values to a notionally common scale.

After that we use normalized random variables to calculate weighted sum of random variables. This step is performed as in AHP method except that we use random variables instead of numbers. With that sum we can use three methods to obtain final ranking of web service.

The method contains several steps. The remainder of this section will be divided into these steps.

Step 1: Calculate prioritization vector - Prioritization vector can be calculated using any of eigenvectors calculation method. Eigenvector should have all positive numbers and should be normalized, so that sum of all vectors element should equal to 1. In this step, we obtain ranking of attributes. One can also use other method proposed by Saaty or other authors such as: normalized arithmetic mean over columns, normalized geometric mean over columns etc. In the example, we use standard eigenvector method. Input of this step is pairwise comparison matrix b_{ij}. Output of this step is prioritization vector p_j, where $1 \leq j \leq m$ and $\sum_{j=1}^{m} p_j = 1$.

Step 2: changing optimization type - For each V_{ij} where optimization type is 'minimize', we have to change it to 'maximize'. Random variable should also remain positive and should have the same rate of change as original random variable. If we know maximum value of V_{ij} for each i we can use it to obtain new random variable. We denote the maximum value as $max_{V_{ij}}$. The formula for new random variable is as follows:

$$V_{ij} := max_{V_{ij}} - V_{ij} \quad (1)$$

However, if random variable has no boundary we can use another formula:

$$V_{ij} := := \frac{1}{V_{ij}} \quad (2)$$

Here we also update PDFs of random variables according to previous changes. Input of this step is random variables V_{ij} and their PDFs v_{ij} . Output is updated random variables and their PDFs.

Step 3: transformation of QoS random variables - We do this operation for each QoS random variable V_{ij} . Significance of changes should be the same for every random variable. Our goal is to obtain linear correlation between QoS attributes and the level of support to the overall objective. Transforming random variables should be done according to rules described in [11]. Two main effects which may show up during studying of QoS attributes are saturation and critical hurdle effect or even a combination of them. In both cases rising the QoS attributes values leads to nonlinear growth of level of support to the overall objective.

In the first case the rise of QoS values first leads to fast grow of level of support and then the growth is insignificant. That situation may occur for example when we want to buy a computer and QoS attribute is RAM capacity. As we increase RAM at first the level of support increase very fast as every new megabyte of RAM gives us more speed to our computer. Then after we reach our desired RAM level the growth becomes less significant and support level grows very slow. In order to make linear growth of support we should transform QoS values with for example following function:

$$f(X) = \begin{cases} (X/4)^{2/3} & when\ X \in [0,4] \\ (X/4)^{1/4} & when\ X \in [4,+\infty] \end{cases} \quad (3)$$

In this case when X is less than 4(GB) the function grows significantly - the same as demand on RAM. If the RAM value exceeds 4(GB) the function grows but much slower than before. That means that our demand on RAM is fulfilled and more RAM has less value than before.

In the second case (hurdle effect) the rise of QoS values first leads to slow growth of level of support and then the growth is more than proportional. That situation may occur for example when we want to buy a sports car for racing. One of QoS attributes is maximum velocity. In formula one racing it is very important to have maximum possible velocity as high as possible. However, if maximum velocity is from 0 km/h to 250 km/h it is insignificant because all other racing cars will reach better performance. The growth of support level becomes significant from 250 km/h. In order to make linear growth of support one can use following function:

$$f(X) = \begin{cases} 0 & when\ X \in [0,250] \\ X - 250^{3/2} & when\ X \in [250,+\infty] \end{cases} \quad (4)$$

Both above cases should be considered as an example not consulted with domain experts. In real life cases on should consult transformation function to fit the level of support.

Input of this step are random variables V_{ij} and their PDFs v_{ij} . Output is updated random variables and their PDFs.

Step 4: normalization of QoS random variables - After transforming random variable we perform normalization. Normalization in standard AHP algorithm with tangible measure attributes relies on dividing attribute values for each alternative by sum of attribute values. In our case, we divide QoS random variable by the sum of expected values of QoS random variables for all alternatives for the given attribute. Normalization formula is given by following function:

$$V_{ij} = \frac{v_{ij}}{\sum_{i=1}^{n} Ev_{ij}} \quad (5)$$

Normalization is done for each web service and for each attribute. Input of this step is random variables V_{ij} and their PDFs v_{ij} . Output is updated random variables and theirs PDFs.

Step 5: ranking random variables - In this step we calculate random variables which will be used to determine final ranking of web services. This step is performed the same as in standard AHP method, except the fact that we are using random variables instead of numbers in alternatives attributes. For each web service w_i we define new random variable as follows:

$$R_i := \sum_{j=1}^{m} V_{ij} p_j \quad (6)$$

Random variables R_i are called ranking random variables. Their PDFs are needed to calculate the final ranking. Visualization of PDFs also gives us information about likelihood of having certain ranking.

Input of this step is random variables V_{ij}, their PDFs v_{ij} and prioritization vector p_j . Output is ranking random variable R_i for each web service.

Step 6: ranking calculation - Ranking calculation is a task performed on ranking random variables. We can use several ranking measures however we think that ranking function list can be extended. Ranking functions for ranking random variables and their usage are as follows:

- P-BEST - Probability of having the best rank in deterministic method:
  $rank(w_i) = P(R_i \geq R_k, 1 \leq k \leq n)$
  when our goal is the first rank only.
- P-AT-LEAST - Probability of having at least k-th place in deterministic method
  $rank(w_i) =$
  $P(\exists_{L \subset \{1,2,\dots,n\}} \forall_{l \in L} R_i \geq R_l \wedge |L| \geq n-k+1)$
  when our goal is to be somewhere on the top.
- P-THRESHOLD - Probability of having rank more than threshold
  $rank(w_i) = P(R_i > t)$
  when we want to avoid a situation of having very bad rank.
- EV - Expected value of ranking random variable
  $rank(w_i) = ER_i$
  standard approach in many methods, especially when we care about the best result only.
- EU - Expected utility of ranking random variable
  $rank(w_i) = E[u(R_i)]$
  approach from economics. Used when we want to avoid risk.
- DEE - Domain expert evaluation of ranking random variable - used when we want to have depth evaluation, however it can be subjective and easily manipulative.

The result of this function, like in standard AHP method, is final ranking. Ranking results can be obtained using tool in [5] or Monte Carlo method.

Input of this step is ranking random variables R_i and the output of this step is final ranking $rank\,(w_i)$.

Summary. The method of ranking web services was shown here. Our method does not suggest which transformation function or ranking calculation function should be used. It depends on many criteria as shown in [11]. In the next section, we will provide example with all suggested functions except domain expert evaluation and show the differences in final rankings.

4. Experiment

In this section, we provide descriptive example on how different ranking functions may affect final ranking. We assume that we have 3 web services (w_1, w_2, w_3) with three QoS attributes: total execution time (a_1), throughput (a_2) and availability(a_3). All QoS attributes are random variables. Total execution time should be minimized and never exceed 5s. This attribute is also correlated with level of support linearly. Throughput should be maximized and it is correlated with level of support quadratically. The availability attribute should be maximized and it is correlated with level of support with the following function: $f(X) = \log_{10}(\frac{1}{100-X})$. This parameter is also bounded from above by 100. We also assume that this parameter is focused in only one point.

In the following pictures, we show PDFs of QoS attributes. In each row, there is shown QoS PDF for subsequent web services. In first column, there are PDFs of total execution time attribute, in second column there are PDFs of throughput. We do not illustrate PDF of availability, because it would be diracs delta focused in one specific point. Values of these points for subsequent web services are: [99.99,99.9995,99.9999999]

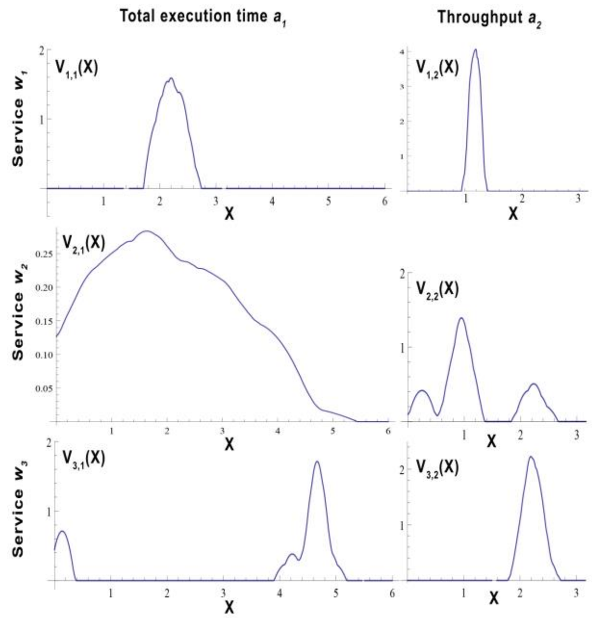

Fig. 1 Probability density functions of QoS attributes for each service

Pairwise comparison between attributes is given in the matrix below:

$$\begin{array}{c} \\ a_1 \\ a_2 \\ a_3 \end{array} \begin{array}{c} \begin{array}{ccc} a_1 & a_2 & a_3 \end{array} \\ \begin{pmatrix} 1 & 3 & 5 \\ \frac{1}{3} & 1 & 3 \\ \frac{1}{5} & \frac{1}{3} & 1 \end{pmatrix} \end{array} \quad (7)$$

From the given matrix one can obtain eigenvector which is our prioritization vector. The eigenvector is calculated in step 1 and is its output. This vector is as follows:

$$\begin{array}{c} a_1 \\ a_2 \\ a_3 \end{array} \begin{pmatrix} 0.637 \\ 0.258 \\ 0.105 \end{pmatrix} \quad (8)$$

In step 2 we have to change optimization type so all parameters should be maximized. In our case, total execution time should be changed. As we know from description of our task it is bounded from above by 5. That's why we can use function mentioned in algorithm: $V_{i,1} := 5 - V_{i,1}$ where $1 \le i \le 3$. Every other attribute should be maximized so there is nothing to be done with them.

In step 3 we have to transform QoS attributes so that all of them grow linearly with growth of level of support. In the description, we have explicit functions that link level of support to overall objective and QoS attributes. We use these functions for each $i \in 1, 2, 3$:

$$\begin{cases} V_{i,1} := V_{i,1} \\ V_{i,2} := V_{i,2}^2 \\ V_{i,3} := \log_{10} \dfrac{1}{100 - V_{i,3}} \end{cases} \quad (9)$$

In step 4 we perform normalization. To do so we use formula in equation (5). In step 5 we multiply random variables by values from prioritization vector using equation (6). All operations on random variables were done using application [5]. PDFs of ranking random variables are shown on the picture below.

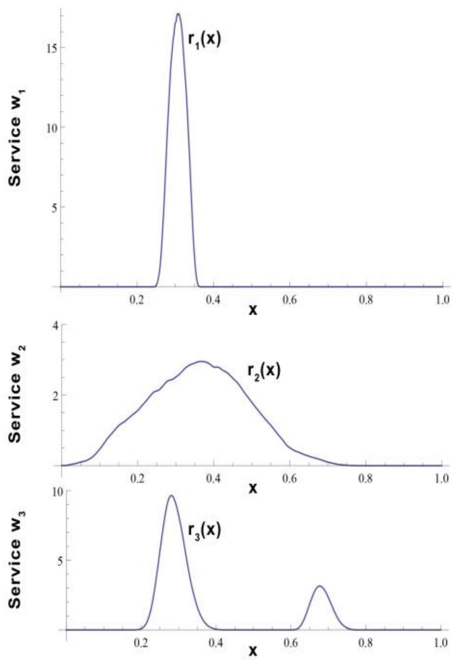

Fig. 2 Probability density functions of ranking random variables for each service

From PDFs of ranking random variables we performed step 6 and calculated final rankings. We used every method described in step 6, except domain expert evaluation. Comparison between rankings is shown in the table 1.

Table 1: Comparison between different ranking methods. P-BEST - probability of having the best rank, P-AT-LEAST - probability of having at least 2nd best rank, P-THRESHOLD - probability of having rank greater than threshold (0.25), EV - expected value, EU – expected utility (utility function: $1 - e^{-10x}$)

Service	Method				
	P-BEST	P-AT-LEAST	P-THRESHOLD	EV	EU
$rank(w_1)$	0.199	0.348	0.999	0.307	0.952
$rank(w_2)$	0.496	0.351	0.786	0.360	0.941
$rank(w_3)$	0.305	0.299	0.903	0.372	0.953

From table 1 one can choose the suitable ranking method and obtain final ranking. This example shows that results for various probabilistic versions of AHP algorithm can differ. One can observe that for P-BEST we have sure winner w_2, however it has the worst expected utility and P-THRESHOLD. The similar observation could be made for other services. The main goal of this example was to show the case where there is no unequivocal winner and it depends on ranking method which one will be the best. Choosing ranking method depends on many circumstances especially on our main goal of service selection.

5. Conclusion and future works

In the article, we showed a method to obtain rankings for QoS attributes in SOA environment and in generality for all alternatives attributes described as continuous random variables. Our method uses probabilistic calculations on random variables which can be performed using Monte Carlo method or statistical applications. We focused on providing descriptive example which will show differences between various ranking methods. These methods are reserved only for our version of multi criterion optimization as for deterministic case it has no meaning.

Our AHP method and problem formulation has many applications in other decision theory problems. We focused on service selection problem which was resolved using AHP method but only in deterministic case. In management, there can be many uncertain attributes such as overall cost or overall project time. These values can be expressed using random variables and PDFs can be obtained basing on previous projects. Other areas of applications are for example: finances, contractor selection, project management, bookmakers, work safety evaluation and many more.

Our main future task is to create an extended AHP method where one can use random variables in pairwise comparison and in alternatives attributes. We also want to focus our studies on different application of probabilistic AHP method in various fields especially in IT. Our goal is to create a method suitable for most multi-criteria decision analysis problems and methodology for easy applying it.

References

[1] Dastjerdi, A. V. and Buyya, R. (2011). A taxonomy of qos management and service selection methodologies for cloud computing, Cloud Computing: Methodology, Systems, and Applications pp. 109–131. URL: http://www.buyya.com/papers/QoSTaxonomyCloud2011.pdf

[2] Dragović, I., Turajlić, N., Radojević, D. and Petrović, B. (2014). Combining boolean consistent fuzzy logic and ahp illustrated on the web service selection problem, International Journal of Computational Intelligence Systems 7(sup1): 84–93.
[3] Hwang, S.-Y., Wang, H., Srivastava, J. and Paul, R. A. (2004). A probabilistic qos model and computation framework for web services-based workflows, Conceptual Modeling–ER 2004, Springer, pp. 596–609.
[4] Hwang, S.-Y., Wang, H., Tang, J. and Srivastava, J. (2007). A probabilistic approach to modeling and estimating the qos of web-services-based workflows, Information Sciences 177(23): 5484–5503.
[5] Jaroszewicz, S. and Korzen, M. (2012). Arithmetic operations on independent random variables: a numerical approach, SIAM Journal on Scientific Computing 34(3): A1241–A1265.
[6] Kang, G., Liu, J., Tang, M. and Xu, Y. (2012). An effective dynamic web service selection strategy with global optimal qos based on particle swarm optimization algorithm, Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International, IEEE, pp. 2280–2285.
[7] Lafleur, J. M. (2011). Probabilistic ahp and topsis for multi-attribute decision-making under uncertainty, Aerospace Conference, 2011 IEEE, IEEE, pp. 1–18.
[8] Menasce, D. (2002). Qos issues in web services, Internet Computing, IEEE 6(6): 72–75.
[9] Menascé, D. A., Casalicchio, E. and Dubey, V. (2010). On optimal service selection in service oriented architectures, Performance Evaluation 67(8): 659–675.
[10] Negi, N. and Chandra, S. (2014). Web service selection on the basis of qos parameter, Contemporary Computing (IC3), 2014 Seventh International Conference on, IEEE, pp. 495–500.
[11] Perego, A. and Rangone, A. (1996). On integrating tangible and intangible measures in ahp applications: A reference framework, Systems, Man, and Cybernetics, 1996., IEEE International Conference on, Vol. 3, IEEE, pp. 1836–1841.
[12] Saaty, T. L. (1980). The analytical hierarchical process, J Wiley, New York.
[13] Schuller, D., Siebenhaar, M., Hans, R., Wenge, O., Steinmetz, R. and Schulte, S. (2014). Towards heuristic optimization of complex service-based workflows for stochastic qos attributes, International Conference on Web Services (ICWS).
[14] Tang, M. and Ai, L. (2010). A hybrid genetic algorithm for the optimal constrained web service selection problem in web service composition, Evolutionary Computation (CEC), 2010 IEEE Congress on, IEEE, pp. 1–8.
[15] Wang, P., Chao, K.-M., Lo, C.-C. and Farmer, R. (2011). An evidence-based scheme for web service selection, Information Technology and Management 12(2): 161–172.
[16] Wang, Y. and Vassileva, J. (2007). A review on trust and reputation for web service selection, Distributed Computing Systems Workshops, 2007. ICDCSW'07. 27th International Conference on, IEEE, pp. 25–25.
[17] Yu, T. and Lin, K.-J. (2005a). Service selection algorithms for composing complex services with multiple qos constraints, Service-Oriented Computing-ICSOC 2005, Springer, pp. 130–143.
[18] Yu, T. and Lin, K.-J. (2005b). Service selection algorithms for web services with end-to-end qos constraints, Information Systems and E-Business Management 3(2): 103–126.
[19] Zhao, X., Huang, P., Liu, T. and Li, X. (2012). A hybrid clonal selection algorithm for quality of service-aware web service selection problem, Int J Innov Comput Inf Control 8(12): 8527–8544.
[20] Zheng, H., Yang, J. and Zhao, W. (2010). Qos probability distribution estimation for web services and service compositions, Service-Oriented Computing and Applications (SOCA), 2010 IEEE International Conference on, IEEE, pp. 1–8.
[21] Zheng, H., Yang, J., Zhao, W. and Bouguettaya, A. (2011). Qos analysis for web service compositions based on probabilistic qos, Service-Oriented Computing, Springer, pp. 47–61.

Ireneusz Jozwiak, Professor.
Education: Mc in Computer Science, Department of Electronics, Wroclaw University of Technology (WUT), 1975; Doctor's degree in Computer Science, Institute of Engineering Cybernetics, Wroclaw University of Technology, 1979; Habilitation degree in Computer Science, Automatic and Robotics, Institute of System Research of the Polish Academy of Sciences, 1994; Professor of Wroclaw University of Technology, 1998.

Michal Kedziora received his M.Sc. (Eng.) and Ph.D. degree in Computer Science from Wroclaw University of Science and Technology, Wroclaw, Poland. In 2016 Dr. Kedziora was recipient of Thelxione postdoctoral research grant in University of Wollongong, New South Wales, Australia. Dr. Kedziora is Certified Information Systems Security Professional (CISSP) and Digital Forensics Investigator (EnCE) with many years working in public and private sector.

Aleksander Marianski, 2006-2011 MSc degree in Computer Science, Wroclaw University of Technology, Faculty of Computer Science and Management. 2012-now: PhD studies in Computer Science, Wroclaw University of Technology, Department of Computer Science and Management. Specialization: decision theory Affiliations and functions: 2009-2011 assistant in SOA project at Wroclaw University of Technology Research interests: applied probability in computer science