# Optimization of Multiple-Objective Web Service Selection Using Fractional Programming

Milan Stanojević · Bogdana Stanojević · Nina Turajlić

Received: 22 July 2014 / Revised: 10 August 2014 / Accepted: 5 September 2014 /

Published online: 11 November 2014

© Springer-Verlag Berlin Heidelberg 2014

Abstract The Service Oriented Approach (SOA) is based on the idea that distributed applications can be flexibly composed by integrating independently developed component services. Due to the fact that a large number of available services offer similar functionality, when choosing actual services to be included in the composition their non-functional (Quality of Service – QoS) properties must also be taken into account. On the other hand certain constraints regarding the required performances (i.e. the required QoS levels) may also be given. Therefore, web service selection presents a multiple-objective multiple constraint problem and in this paper we model it as a discrete multiple-objective linear fractional programming (MOLFP) problem. In order to solve this problem we use a previously published technique for generating strongly efficient solutions to a continuous MOLFP problem to the discrete case. We present a complete methodology for solving the multiple criteria web service selection problem and report the experimental results obtained by testing it on 50 random instances generated by varying the input parameters.

**Keywords** Web service selection  $\cdot$  SOA  $\cdot$  QoS  $\cdot$  Multiple objective programming  $\cdot$  Linear fractional programming

M. Stanojević (⋈) · N. Turajlić

Faculty of Organizational Sciences, University of Belgrade, Jove Ilića 154, Belgrade, Serbia e-mail: milans@fon.bg.ac.rs

N. Turajlić

e-mail: nina.turajlic@fon.bg.ac.rs

B. Stanojević

Mathematical Institute of the Serbian Academy of Sciences and Arts, Kneza Mihaila 36, Belgrade,

e-mail: bgdnpop@mi.sanu.ac.rs

Springer

### 1 Introduction

A large number of software components providing certain functionality i.e. web services are accessible on the Internet and they can be grouped into classes on the basis of the functionality they provide. When a desired functionality cannot be provided by a single service it might be achieved by executing a set of different services which is the basis of the Service Oriented Approach (SOA) [2]. In SOA distributed applications can be flexibly composed by integrating independently developed component services. To this end, the first step would be to establish which classes are necessary for achieving the desired complex functionality. The next step would be to choose actual services from each of the required classes to be included in the composition. However, with the proliferation of web services offering similar functionality, developed on various platforms and in various programming languages, the problem of how to choose the best one from a given class has become increasingly important. Thus the non-functional properties of services—Quality of Service (QoS) have become even more relevant. On the other hand, certain constraints regarding the required performances (i.e. the required QoS levels) are usually also given. Therefore, the goal is to select the composition with the best performance which at the same time satisfies the given constraints.

There are different approaches in literature for modeling the web service selection problem. In a large number of them it is modeled as a multiple-dimensional multiple-choice knapsack problem (e.g. [1,13]). Other approaches propose modeling the problem as a multiple-constrained optimal path problem [11,12] or a resource constrained project scheduling problem [5]. Most of these approaches use the Simple Additive Weighting technique for combining the multiple QoS related criteria into a single utility function.

At the same time various approaches, both exact and heuristic, have been proposed, as well, for solving the problem. Approaches for obtaining exact solutions of the problem include Integer Linear Programming [1,13], Branch and Bound Integer Programming [11,12], Dynamic Programming [3,4] (however, in these last two papers the authors assume the absence of QoS constraints which is too restrictive for most real-world problems), etc. In light of the large number of different approaches for solving the problem which are closely tied to the models used for formulating the problem it would be impossible to give a comprehensive overview within the space constraints of the paper. A detailed survey of the various approaches can be found in [6,9].

When the selection has to be made on the basis of several QoS attributes the problem becomes a multiple objective one, and finding the best possible compromise between objectives values is desired. The solution concept of such problems is related to the efficiency. In the present study we present a general approach for solving the web service selection problem based on the MOLFP model, and then select four QoS attributes to illustrate it. This idea was first presented in [8].

In our case study we look at the following QoS attributes: the time taken to send a request and to receive a response (the response time), the time the server takes to process a given request (the latency), the probability that a service is operating (the availability), and the ratio of the number of error messages over the total number



of messages as QoS attributes (the reliability). Further, we linearize the non-linear functions that were used for modeling some of the attributes, and obtain a multiple objective linear programming problem to solve. In order to reduce the number of the objectives we favorable combine the original objectives and formulate a bi-objective linear fractional programming (BOLFP) problem. Using the procedure introduced in [7] we generate an efficient solution to the BOLFP problem and adopt it as a solution to our initial problem.

The remainder of the paper is organized as follows: in Sect. 2 the web service selection problem is explained in detail. Sect. 3 describes the MOLFP model that we propose for solving the web service selection problem. This section also outlines the technique for deriving efficient solutions to the formulated model. Experimental results are presented in Sect. 4. Finally, in Sect. 5 conclusions and ideas for further work are offered.

## 2 The Web Service Selection Problem

Selecting the best web service for a given functional requirement from a vast collection of suitable services available on the internet is not an easy task, and non-functional requirements (QoS) usually play an important role in the decision making process. When the desired functionality cannot be provided by a single service but might be achieved by executing a set of services the task becomes even more complex. First it is necessary to establish which classes are necessary for achieving the desired complex functionality. Then, actual services are choosen from each of the required classes, and included in the composition. The web service selection problem therefore, consists of selecting component services in such a manner that the resulting composition not only fulfills the desired functional requirements but also ensures a good compromise of the relevant non-functional requirements.

Given a set of classes  $C = \{C(1), \ldots, C(m)\}$  which are necessary to provide the required functionality, each C(i),  $i = 1, \ldots, m$  consisting of a set of services  $s_{ij}$ ,  $j = 1, \ldots, p_i$ . The aim is to select a single service from each of the classes to be included in the composition CS. Since the selection should be made on the basis of non-functional requirements first the relevant QoS attributes must be chosen  $q^k$ ,  $k = 1, \ldots, n$ .

Though, there exist various QoS attributes which could be regarded (performance, reliability, scalability, capacity, robustness, exception handling, accuracy, integrity, accessibility, availability, interoperability, security, network-related QoS requirements—a detailed description can be found in [10]) only four most commonly used attributes have been chosen for illustrating the proposed approach (see Table 1).

When selecting services for a composition the basis for the selection should be the performances of the composite service as a whole. Several challenges arise when computing the overall QoS for a web service composition. First, the QoS values need to be aggregated in some way, and the nature of the aggregation depends on the chosen QoS attributes e.g. the response time of a web service composition is calculated as the sum of the response times of the participating services, the throughput of a composition is equal to the throughput of its fastest component, while the reliability is calculated as



Table 1	Selected QoS attributes	QoS attribute	Description	Unit
		Availability	The probability that a service is operating	Percent
	Response time Reliability	Response time	Time taken to send a request and receive a response	Millisecond
		Ratio of the number of error messages to total messages	Percent	
		Latency	Time the server takes to process a given request	Millisecond

T

Table 2 Composite QoS characteristics

QoS attribute	Aggregation formula	Туре
Availability	$q^{1}(x) = \prod_{i=1}^{m} \sum_{j=1}^{p_{i}} q_{ij}^{1} x_{ij}$	Max
Response time	$q^{2}(x) = \sum_{i=1}^{m} \sum_{j=1}^{p_{i}} q_{ij}^{2} x_{ij}$	Min
Reliability	$q^{3}(x) = \prod_{i=1}^{m} \sum_{j=1}^{p_{i}} q_{ij}^{3} x_{ij}$	Max
Latency	$q^{4}(x) = \sum_{i=1}^{m} \sum_{j=1}^{p_{i}} q_{ij}^{4} x_{ij}$	Min

the product of the reliabilities of the participating services etc. (Table 2). Next, if the selection should be made on the basis of more than one QoS attribute it is necessary to take into account the fact that for some of the QoS attributes the maximum value is preferable, while for others the minimum value is preferred.

Generally, certain constraints regarding the performances (i.e. the required QoS attribute levels) of the composite service are also given  $Q = \{Q^1, \dots, Q^n\}$ , so the goal then becomes to select the best composition which satisfies the given constraints. It should be noted that, in general, the set of attributes on the basis of which the selection should be made need not necessarily be the same as the set of attributes for which the constraints are given and the proposed approach does not require that these sets be the same.

When the selection has to be made on the basis of several QoS attributes the problem becomes a multiple-objective problem. Denoting by  $K_{max}$  and  $K_{min}$  the sets of indexes of the attributes that have to be maximized and minimized, respectively, the problem can be stated as follows:

$$\max q^k(x), k \in K_1$$
  
$$\min q^k(x), k \in K_2$$

while satisfying the following constraints:

$$q^{k}(x) \leq Q^{k}, \quad k \in A_{min}$$
  
 $q^{k}(x) \geq Q^{k}, \quad k \in A_{max}$   
 $\sum_{j=1}^{p_{i}} x_{ij} = 1, i = 1, \dots, m,$   
 $x_{ij} \in \{0, 1\}, \quad i = 1, \dots, m, j = 1, \dots, p_{i},$ 

where  $A_{max}$  and  $A_{min}$  identify the indexes of the relevant QoS attributes on which upper and lower bounds, respectively are imposed; i = 1, ..., m index the classes in



the composition and  $j = 1, ..., p_i$  the services in class C(i). The variable  $x_{ij}$  is set to 1 if the corresponding service should be included in the composition otherwise, it is set to 0.

# 3 Solving the Web Service Selection Problem

There are different approaches in the literature for solving the web service selection problem. The Simple Additive Weighting technique is generally used for combining the multiple QoS related criteria into a single utility function. Each of the chosen QoS attributes is assigned a preference (usually arbitrary) and a weighted sum is composed. Unfortunately, the change of the weights does not produce a proportional change in the rezults, thus the DM's preferences are not well expressed in this way. Bearing in mind that the QoS attributes are often heterogeneous and defined using different metrics it is also necessary to normalize their values so that they can be uniformly combined in the utility function. Since the objectives are many times of different nature, the normalization may be irrelevant in practice.

In this paper we propose a new method for solving the multiple-objective problem: we pair the initial objectives, based on their nature (unit measure and optimization type max/min) and optimize their ratios. In this way we reduce the number of objectives, we avoid the assignment of arbitrary weights to the chosen QoS attributes, and we replace the normalization of the values of these attributes by a favorable aggregation.

The proposed multiple-objective fractional programming problem is

$$\max_{x \in X} \left\{ \frac{q^{i_1}(x)}{q^{i_2}(x)}, \frac{q^{i_3}(x)}{q^{i_4}(x)}, \dots, \frac{q^{i_{2h-1}}(x)}{q^{i_{2h}}(x)} \right\}$$
(1)

subject to:

$$q^{k}(x) \leq Q^{k}, \quad k \in A_{min}$$

$$q^{k}(x) \geq Q^{k}, \quad k \in A_{max}$$

$$\sum_{j=1}^{p_{i}} x_{ij} = 1, \quad i = 1, \dots, m,$$

$$x_{ij} \in \{0, 1\}, \quad i = 1, \dots, m, \quad j = 1, \dots, p_{i},$$
(2)

where indexes  $i_l$ ,  $l=1,\ldots,2h$  are such that  $i_1,i_3,\ldots,i_{2h-1}\in K_{max}$ , and  $i_2,i_4,\ldots,i_{2h}\in K_{min}$ . Unfortunately in a real-world problem, neither precisely half of the attributes are maximized and the second half minimized nor for all of them can be found a convenient pair. Such situation can be solved either by replacing some objectives by themselves multiplied by -1 and in the same time changing the optimization direction, i.e. using the advantage of max  $(f(x)) = -\min(-f(x))$ ; or considering artificial objectives, constantly equal to 1 as denominators and/or numerators for some fractions.

One can consider a disadvantage the optimization of fractional objectives instead of the original objectives. But, in the case of linear original objectives, the new objectives are linear fractional and simple procedures involving linear optimization problems can be used for optimizing them (a solving procedure is described in next section); and in



the case of original nonlinear objectives, the fractions of nonlinear functions do not necessarily have worse properties than their numerators or denominators.

For our case study, the initial problem with four objectives is transformed into the discrete bi-objective fractional problem

$$\max_{x \in X} \left\{ \frac{q^1(x)}{q^2(x)}, \frac{q^3(x)}{q^4(x)} \right\} \tag{3}$$

subject to:

$$q^{k}(x) \leq Q^{k}, \quad k = 2, 4,$$

$$q^{k}(x) \geq Q^{k}, \quad k = 1, 3,$$

$$\sum_{j=1}^{p_{i}} x_{ij} = 1, i = 1, \dots, m,$$

$$x_{ij} \in \{0, 1\}, \quad i = 1, \dots, m, j = 1, \dots, p_{i}.$$

$$(4)$$

The numerators and denominators in the objective functions of Problem (3)–(4) represent the composite QoS values calculated according to the formulas given in Table 2. Taking into account the nature (unit measure and type) of the selected attributes, the composite availability and response time have been chosen for the first fraction, while the composite reliability and latency have been chosen for the second fraction. Since the response time and latency should be minimized while the availability and reliability should be maximized the former are designated as the denominators while the latter are designated as the numerators of the fractions. Thus, both fractional objectives will be maximized.

Since  $q^1$  and  $q^3$  are non-linear Problem (3)–(4) is not quite easy for direct solving, but making some adjustments we can transform it to a BOLFP problem. Applying a logarithmic function to  $q^1$  and  $q^3$  we transform the products into sums of logarithms. The relation

$$\log \left( \prod_{i=1}^{m} \sum_{j=1}^{p_i} q_{ij}^k x_{ij} \right) = \sum_{i=1}^{m} \sum_{j=1}^{p_i} (\log(q_{ij}^k)) x_{ij}, k = 1, 3$$

is true knowing that, for any feasible solution, exactly one  $x_{ij}$  is 1 and all others are 0, for each i = 1, ..., m.

Next, since it is convenient to have both numerators and denominators expressed by positive functions [7], and the logarithmic functions do not meet this requirement we scale them by adding a value  $\alpha_l$  to numerators to render them positive. Since the denominator functions are positive there is no need to adjust them. The values  $\alpha_l$ , l=1,2 are calculated as follows:

$$\alpha_{1} = w + \sum_{i=1}^{m} \left| \min_{j=1,\dots,p_{i}} \log \left( q_{ij}^{1} \right) \right|$$

$$\alpha_{2} = w + \sum_{i=1}^{m} \left| \min_{j=1,\dots,p_{i}} \log \left( q_{ij}^{3} \right) \right|$$
(5)



where w > 0 is a parameter which is set to an appropriate value to ensure that the smallest possible sum is always strictly greater than smallest possible value of the sum of logarithms of the aggregating function (e.g. w = 1).

Now the discrete BOLFP model of the web service selection problem can be formulated as follows:

$$\max_{x \in X} \left\{ z(x) = \left( \frac{\sum_{i=1}^{m} \sum_{j=1}^{p_i} \left( \log \left( q_{ij}^1 \right) \right) x_{ij} + \alpha_1}{\sum_{i=1}^{m} \sum_{j=1}^{p_i} q_{ij}^2 x_{ij}}, \frac{\sum_{i=1}^{m} \sum_{j=1}^{p_i} \left( \log \left( q_{ij}^3 \right) \right) x_{ij} + \alpha_2}{\sum_{i=1}^{m} \sum_{j=1}^{p_i} q_{ij}^4 x_{ij}} \right) \right\}$$
(6)

subject to:

$$\sum_{i=1}^{m} \sum_{j=1}^{p_i} \left( \log \left( q_{ij}^k \right) \right) x_{ij} \ge \log(Q^k), \ k = 1, 3,$$

$$\sum_{i=1}^{m} \sum_{j=1}^{p_i} q_{ij}^k x_{ij} \le Q^k, \qquad k = 2, 4,$$

$$\sum_{j=1}^{p_i} x_{ij} = 1, \qquad i = 1, \dots, m,$$

$$x_{ij} \in \{0, 1\}, \qquad j = 1, \dots, p_i, i = 1, \dots, m.$$
(7)

The use of a novel technique introduced in [7] is proposed for solving the posed problem, in the case of linear-fractional objectives. As shown in the cited paper this technique can be used to generate strongly efficient solutions to a MOLFP problem starting from any feasible solution. It should be noted that this technique has been formulated for the continuous MOLFP problems, but without any impediment we may use it for solving the discrete MOLFP problems. Thus, Problem (8)

$$\max \sum_{k=1}^{h} (d_{k}^{+} + d_{k}^{-})$$
s.t.  $q^{i_{2k-1}}(x) + \alpha_{1} - d_{k}^{+} = (q^{i_{2k-1}}(x^{*}) + \alpha_{1}) \theta_{k}, k = 1, ..., h$ 

$$q^{i_{2k}}(x) + \alpha_{2} + d_{k}^{-} = (q^{i_{2k}}(x^{*}) + \alpha_{2}) \theta_{k}, \quad k = 1, ..., h$$

$$q^{l}(x) \leq Q^{l}, \quad l \in A_{min},$$

$$q^{l}(x) \geq Q^{l}, \quad l \in A_{max},$$

$$\sum_{j=1}^{p_{i}} x_{ij} = 1, \quad i = 1, ..., m,$$

$$x_{ij} \in \{0, 1\}, \quad j = 1, ..., p_{i}, i = 1, ..., m,$$

$$d_{k}^{+}, d_{k}^{-}, \theta_{k} \geq 0, \quad k = 1, ..., h$$

$$(8)$$

is used successively in the solving procedure  $P_{sMOLF}$  [7] for finding an efficient solution to Problem (1)–(2) starting from a given feasible solution  $x^*$ .

Turning back to our case study, and denoting

$$F^{1}(x) = \sum_{i=1}^{m} \sum_{j=1}^{p_{i}} \left( \log \left( q_{ij}^{1} \right) \right) x_{ij},$$

$$F^{3}(x) = \sum_{i=1}^{m} \sum_{j=1}^{p_{i}} \left( \log \left( q_{ij}^{3} \right) \right) x_{ij}$$



we construct the following discrete linear programming problem on the basis of the posed BOLFP problem (6)–(7):

$$\max \left(d_{1}^{+} + d_{2}^{+} + d_{1}^{-} + d_{2}^{-}\right)$$
s.t.  $F^{1}(x) + \alpha_{1} - d_{1}^{+} = \left(F^{1}(x^{*}) + \alpha_{1}\right)\theta_{1},$ 

$$F^{3}(x) + \alpha_{2} - d_{2}^{+} = \left(F^{3}(x^{*}) + \alpha_{2}\right)\theta_{2},$$

$$q^{2}(x) + d_{1}^{-} = q^{2}(x^{*})\theta_{1},$$

$$q^{4}(x) + d_{2}^{-} = q^{4}(x^{*})\theta_{2},$$

$$F^{k}(x) \ge \log\left(Q^{k}\right), \qquad k = 1, 3,$$

$$q^{k}(x) \le Q^{k}, \qquad k = 2, 4,$$

$$\sum_{j=1}^{p_{i}} x_{ij} = 1, \qquad i = 1, \dots, m,$$

$$x_{ij} \in \{0, 1\}, \qquad j = 1, \dots, p_{i}, i = 1, \dots, m,$$

$$d_{1}^{+}, d_{2}^{+}, d_{1}^{-}, d_{2}^{-}, \theta_{1}, \theta_{2} \ge 0,$$

$$(9)$$

where  $x^* \in X$  is an arbitrary fixed feasible solution to Problem (6)–(7), and  $d_i^+, d_i^-$ , and  $\theta_i$ , j = 1, 2 are positive scalar variables used for describing the change of the nominators, denominators, and fractions respectively.

Problem (9) is the particular case of Problem (8), and is used in the following procedure, with  $x^0$  an arbitrary fixed feasible solution to (6)–(7).

```
Step 1: Set x^* = x^0 and h = 1.
```

Step 2: Solve problem (9). Let  $(x^h, \theta^h, d^{h+}, d^{h-})$  be its optimal solution. Step 3: If  $d_1^{h+} + d_2^{h+} + d_1^{h-} + d_2^{h-} = 0$ , STOP,  $x^h$  is a strongly efficient solution to Problem (6)–(7). Otherwise, set  $x^* = x^h$ , h = h + 1 and go to Step 2.

The philosophy of Problem (9), as it was proven in [7], is that by maximizing the sum of the deviational variables the values of all objectives increase or remain unchanged. Thus, in each step of the procedure a feasible solution that dominates the previous one is obtained. The procedure stops when no increase is possible. These facts function in the same way in the discrete case as in the continuous one.

# 4 Experimental Results

The proposed technique was tested by varying the number of classes (m) and services in the classes (p). The results were compared to the results obtained through an exhaustive search of combinations where possible. Every time an efficient solution to the initial problem was obtained. The proposed algorithm has been implemented in GNU MathProg. The optimization problems were solved using GNU Linear Programming Kit (GLPK). The experiments were performed on 50 random test instances with parameters m and p given in Table 3.

The results show that the proposed technique generates a strongly efficient solution from an arbitrary feasible solution in an extremely short time-frame in just a few iterations. For each instance with m < 10 only 1 iteration was needed, and for instances with m = 10 an efficient solution was found in at most 3 iterations. For each instance with  $p \le 100$  an efficient solution was obtained in less than 1 second. In the case of larger instances, i.e. p = 1,000, the time was between 40 and 400 seconds per iteration.



**Table 3** The average exhaustive search time and the average number of iterations per group of instances

m	p	Average time (s)	Average number of iteration
3	10	0.02	1
7	10	46.86	1
10	10	26218.03	2
10	100	N/A	2.2
10	1000	N/A	2.2

In Table 3 for each group of instances, the average time needed for an exhaustive search of an efficient solution is reported.

#### 5 Conclusion

The web service selection problem for a required functionality consists of creating an optimal composition of web services (in terms of its non-functional requirements i.e. the chosen QoS attributes) by selecting adequate component services, from sets of available services, each providing part of the desired functionality. When the selection is made on the basis of several QoS attributes the problem has multiple objectives.

In this paper we proposed a MOLFP model. The benefits of the proposed approach in comparison to the customary Simple Additive Weighting Technique used in most existing approaches based on the multiple-dimensional multiple-choice knapsack model are twofold. Firstly, the assignment of preferences, which are usually arbitrary, is avoided. Secondly, since the QoS attributes are usually heterogeneous and defined using different metrics it is usually necessary to normalize their values in order to uniformly combine them into a single utility function. Since the proposed model does not require that a single utility function be constructed there is no need to normalize these values.

Subsequently, for solving the posed problem the use of a novel technique for generating efficient solutions for such problems was proposed. The experimental results showed that the proposed technique yielded an efficient solution in a short time-frame and in just a few iterations. The exact solution for 100 services per class from 10 classes was found, through an exhaustive search of combinations, in 7.5 h while it was found in less than 2 s by the proposed technique. It should be noted that even for larger instances containing up to a 1,000 services per class the proposed technique took less than a minute while it was impossible to time the exhaustive search for these instances.

The proposed technique yields a single strongly efficient solution which is only convenient for a priori decision making. Further work in this area may be related to exploring the possibility of obtaining all of the efficient solutions of the posed problem. It is assumed that this could be achieved by varying the starting point and/or the value of parameter w in (5). Exploring this hypothesis will be the subject of our further work.



**Acknowledgments** This research was partially supported by the Ministry of Education and Science, Republic of Serbia, Project numbers TR36006 and TR32013.

#### References

- Aggarwal R, Verma K, Miller J, Milnor W (2004) Constraint driven web service composition in Meteor-S. In: Proceedings of SCC'04, Shanghai, pp 23–30
- 2. Erl T (2008) SOA: principles of service design. Prentice Hall/PearsonPTR, London
- Yan G, Na J, Zhang B, Yang L, Gong Q (2006) Optimal web services selection using dynamic programming. In: Proceedings of the 11th IEEE international symposium on computers and communications (ISCC'06), Pula-Cagliari, pp 365–370
- Huang Z, Jiang W, Hu S, Liu Z (2009) Effective pruning algorithm for QoS-aware service composition. In: Proceedings of the 11th IEEE conference on commerce and enterprise computing (CEC'09), Vienna, pp 519–522
- Jaeger M, Muhl G, Golze S (2005) QoS-aware composition of web services: an evaluation of selection algorithms. LNCS 3760:646–661
- Yulu S, Xi C (2011) A survey on QoS-aware web service composition. In: Proceedings of MINES '11, Nanjing, pp 283–287
- Stanojević B, Stanojević M (2013) On the effciency test in multi-objective linear fractional programming problems by Lotfi et al. 2010. Appl Math Model 37(10–11):7086–7093
- Stanojević M, Stanojević B, Turajlić N (2013) Solving the web service selection problem using multiobjective linear fractional programming. In: Proceedings of the 11th Balkan conference on operational reasearch, pp 617–622
- Strunk A (2010) QoS-aware service composition: a survey. In: Proceedings of the ECOWS'10, Nicosia, pp 67–74.
- W3C (2003) QoS for web services: requirements and possible approaches. http://www.w3c.or.kr/ kr-office/TR/2003/ws-qos/
- 11. Yu T, Lin KJ (2005) Service selection algorithms for composing complex services with multiple QoS constraints. In: Proceedings of 3rd international conference on service-oriented computing (ICSOC'05), Amsterdam, LNCS 3826, pp 130–143
- 12. Yu T, Zhang Y, Lin KJ (2007) Efficient algorithms for web services selection with end-to-end QoS constraints. ACM Trans on Web 1(1):1–26
- Zeng L, Benatallah B, Ngu AHH, Dumas M, Kalagnanam J, Chang H (2004) QoS-aware middleware for web services composition. IEEE Trans Softw Eng 30(5):311–327



Milan Stanojević was born in Belgrade, Serbia in 1965. He graduated at University of Belgrade, Faculty of Organizational Sciences in 1990. He obtained doctoral degree at the same faculty in 2005. Since 1993 he works at Faculty of Organizational Sciences, in the begining as a teaching assistant and now as associate professor of operational research. He has published more than 50 papers in national and international journals, and conference proceedings in the field of operational research. His research interest includes multiobjective optimization, combinatorial optimization and software for operational research.





Bogdana Stanojević was born in Oradea, Romania in 1972. She graduated Mathematics and Computer Science specialization at "Transilvania" University of Brasov in 1995, and she obtained her doctoral degree in Mathematics in 2003 from the Romanian Academy. Currently she is researcher at Mathematical Institute of the Serbian Academy of Sciences and Arts. Her research interests include different aspects of fuzzy optimization, multiple objective optimization, fractional programming and mathematical fundamentals of computers.



Nina Turajlić graduated at University of Belgrade, Faculty of Organizational Sciences in 2006. She obtained a doctoral degree at the same faculty in 2014 and is currently employed at the faculty as a teaching assistant. Her research interests include information system design, software development and optimization methods.

