On the Complexity of QoS-Aware Service Selection Problem

Faisal N. Abu-Khzam[1], Cristina Bazgan[2,3], Joyce El Haddad[2],
and Florian Sikora[2](✉)

[1] Departement of Computer Science and Mathematics,
Lebanese American University, Beirut, Lebanon
faisal.abukhzam@lau.edu.lb

[2] PSL, Université Paris-Dauphine, LAMSADE UMR CNRS 7243, Paris, France
{bazgan,elhaddad,florian.sikora}@lamsade.dauphine.fr

[3] Institut Universitaire de France, Paris, France

Abstract. This paper addresses the QoS-aware service selection problem considering complex workflow patterns. More specifically, it focuses on the complexity issues of the problem. The NP-hardness of the problem, under various settings, has been open for many years and has never been addressed thoroughly. We study the problem complexity depending on the workflow structure, the number of workflow tasks, the number of alternative services per task and the categories of quality of service criterion associated to services. We provide for the first time the NP-hardness proof of the problem. Additionally, we show that the problem is polynomial in case of only one criterion per task and pseudo-polynomial if there is a fixed number of criteria.

Keywords: Quality of Service · Service selection · Optimization · Complex workflows

1 Introduction

One of the key features of Service-Oriented Architectures is to realize workflows by composing loosely coupled web services. Each of these services provides a functionality and their composition creates the functionality of a new value-added service, called a composite service. As many providers might offer functionally equivalent services, several candidates might be available for the realization of a task in a workflow. To distinguish among these candidates, their non-functional properties are considered such as Quality of Service (QoS) (e.g. execution duration, execution cost, reputation, and availability). Since only one component service is needed for a task, the service selection problem based on quality of service, denoted QoS-aware Service Selection problem, is an important step in the composition process. It helps to choose services that best meet QoS constraints.

In the literature, the QoS-aware Service Selection problem has been claimed to be NP-hard and then essentially exponential time exact algorithms

A. Barros et al. (Eds.): ICSOC 2015, LNCS 9435, pp. 345–352, 2015.
DOI: 10.1007/978-3-662-48616-0_23

and heuristics have been investigated. Our first contribution in this paper is to establish that QoS-aware Service Selection is solvable in polynomial time for one criterion for complex workflow including those with inclusive patterns (OR pattern) which have never been studied as far as we know. Another contribution is to present a first NP-hardness proof of the problem for workflows with simple structure even with two criteria. Moreover, our approach goes beyond classical QoS categories and adds new ones. However, we show that this problem is only *weakly* NP-hard, which allows the existence of an exact pseudo-polynomial time algorithm. This type of algorithm is of particular interest since typical values for real instances are small and in this case a pseudo-polynomial time algorithm becomes a polynomial time algorithm. Therefore, the problem is not as hard as claimed by previous studies. Our results are thus tight in the sense that our pseudo-polynomial time results are the best possible considering the complexity lower bounds.

Related Work. QoS-aware Service Selection problem has been formulated as an optimization problem and discussed by several authors [1–3,9,10,12,14,15]. Surveys of existing approaches can be found in [5,8,11]. Most of these papers note that the problem is NP-hard. They state that QoS-aware Service Selection is equivalent to Multi-Dimension Multiple-Choice Knapsack problem [2]. To solve QoS-aware Service Selection problem, some approaches propose exact solutions. Bonatti and Festa [3] formulated the problem as a matching problem between requests and offers. The goal is to find a binding between requests and offers, compatible with the given matching and optimal regarding the preferences associated to services and invocations. Yu and Lin [14] proposed a combinatorial approach modeling the problem as a Multiple-Choice Knapsack problem applied only to sequential structure workflows, and a graph approach modeling the problem as a Constrained Shortest Path problem applied to more general structure workflows. Schuller et al. [9] formulated the problem as a linear optimization problem for simple structure workflows, which can be solved optimally using integer linear programming techniques. This approach was extended in Schuller et al. [10] to consider structured as well as unstructured workflows. More recently, Gabrel et al. [6] studied complexity of the problem in case of one criteria and proposed a mixed integer program to solve it. Likewise, a number of heuristic algorithms have been proposed to solve QoS-aware Service Selection. Zeng et al. [15] proposed a local optimization approach which selects Web services one at the time by associating a task of the workflow to the best candidate service. Canfora et al. [4] proposed to tackle the problem with a genetic algorithm. Zhang et al. [16] proposed a strategy to decompose a composite service with a general flow structure into parallel execution paths. Then, the authors modeled service selection problem for each execution path as a multi-objective optimization problem, and presented an ant colony optimization algorithm to solve it. More recently, Trummer et al. [12] proposed three algorithms to solve the problem: a first exact algorithm with exponential complexity, a second polynomial time heuristic algorithm with no guaranteed error bound, and a third polynomial time approximation algorithm with guarantee error bound.

Organization. The rest of the paper is organized as follows. Section 2 presents some background elements related to our problem, as well as definitions. Section 3 provides some positive complexity results. These positive results are tight as Sect. 4 shows complexity lower bounds. Section 5 concludes and highlights some future work. Due to space constraints, proofs are devoted to the full version of the paper.

2 System Model and Problem Statement

2.1 Workflows

A workflow is an abstract business process that combines several tasks. A task represents a step that has to be accomplished by a single web service invocation and is associated with a set of service candidates that are all able to provide the required functionality but might differ in their non-functional properties, namely Quality of Service (QoS) values. The connections (or transitions) between tasks represent a transfer of control from a preceding task to the one that follows. These connections include control structures such as sequence, parallel (AND-split, AND-join), exclusive (XOR-split and XOR-join), and inclusive (OR-split and OR-join) patterns (for details see in [13]). These patterns, written in Business Process Modeling Notation 2.0 could be concatenated or interlaced recursively to create complex workflows.

In this work, we expect all workflows to be *structured* [7] that is meeting the following requirements : (i) having a single *entry* point (i.e. with no incoming connection), (ii) a single *exit* point (i.e. with no outgoing connections), and (iii) a split of some kind is *closed* by a corresponding join of the same kind and the same number of branches (i.e. each AND-split has a corresponding AND-join and each (X)OR-split has a corresponding (X)OR-join). Formally, a structured workflow is defined as follows.

Definition 1 (Structured Workflow). *A structured workflow wf on a set of tasks $T = \{t_1, \ldots, t_n\}$ is defined recursively as:*

- *a task $t_i \in T$ is a structured workflow wf. Both the entry and exit points of wf are t_i in this case.*
- *a sequence pattern $(wf_1, \ldots, wf_\ell)$ of structured workflows $wf_1, \ldots, wf_\ell$. The entry point of wf is the entry point of wf_1, the exit point of wf is the exit point of wf_ℓ, and for all k, $1 \leqslant k < \ell$, the exit point of wf_k has a transition to the entry point of wf_{k+1}.*
- *an AND pattern of structured workflows AND-split$(wf_1, \ldots, wf_\ell)$AND-join where the entry point of wf is AND-split, the exit point of wf is AND-join, and with transitions between AND-split and the entry points of each wf_i and between the exit point of each wf_i and AND-join.*
- *a XOR pattern of structured workflows XOR-split$(wf_1, \ldots, wf_\ell)$XOR-join where the entry point of wf is XOR-split, the exit point of wf is XOR-join, and with transitions between XOR-split and the entry points of each wf_i and between the exit point of each wf_i and XOR-join.*

- *an OR pattern of structured workflows OR-split($wf_1, \ldots, wf_\ell$)OR-join where the entry point of wf is OR-split, the exit point of wf is OR-join, and with transitions between OR-split and the entry points of each wf_i and between the exit point of each wf_i and OR-join.*

Definition 2 (Structured Workflow Without Sharing Tasks). *A structured workflow without sharing is a structured workflow in which every task has one incoming and one outgoing transition.*

We represent workflows as trees of patterns recursively as in Definition 1 where a leaf node represents a task and a non-leaf node is either SEQ for sequential pattern, AND for parallel pattern, XOR (resp. OR) for a choice of one (resp. of one or several) out of several alternative branches.

The control flow describes the execution ordering of the tasks through different patterns. Informally, a subset of tasks $T' \subseteq T$ satisfies the control flow of a workflow wf if T' satisfies the control flow of at least one wf_i in the case of OR pattern, of all wf_i in the case of AND or sequence patterns, or of exactly one wf_i in the case of XOR pattern. More formally:

Definition 3 (Control Flow Satisfaction). *Consider a workflow wf on a set of tasks T. The control flow satisfaction of wf by a subset of tasks $T' \subseteq T$ is defined recursively as:*

- *when wf is a task t_i, then T' satisfies the control flow of wf iff $t_i \in T'$.*
- *when wf is a sequence of workflows $(wf_1, \ldots, wf_\ell)$, then T' satisfies the control flow of wf iff T' satisfies the control flow of each workflow wf_i.*
- *when wf is AND-split($wf_1, \ldots, wf_\ell$)AND-join, then T' satisfies the control flow of wf iff T' satisfies the control flow of each workflow wf_i.*
- *when wf is XOR-split($wf_1, \ldots, wf_\ell$)XOR-join, then T' satisfies the control flow of wf iff T' satisfies the control flow of exactly one workflow wf_i.*
- *when wf is OR-split($wf_1, \ldots, wf_\ell$)OR-join, then T' satisfies the control flow of wf iff T' satisfies the control flow of at least one workflow wf_i.*

2.2 Services and Quality of Service

Services are software components that encapsulate atomic functionality. Since executing a task only one service is needed among service candidates then non-functional properties such as Quality of Service (QoS) are considered. Denoting by $\mathcal{C}$ the set of criteria and assuming a fixed ordering between them, services are described by their QoS vectors of non-negative rational values as follows.

Definition 4 (Service). *A service s is a tuple (f, c) where f represents its offered functionality, and $c = (c_1(s), \ldots, c_p(s))$ represents its QoS vector where $c_i(s)$ is the QoS value for criterion i.*

Consider a workflow wf composed of a set of n tasks $T = \{t_1, \ldots, t_n\}$. For each task $t_j \in T$, there is a set of m_j candidate services $S_j = \{s_{j,1}, \ldots, s_{j,m_j}\}$.

A vector $c(s_{j,\ell}) = (c_1(s_{j,\ell}), \ldots, c_p(s_{j,\ell}))$ is associated to each service $s_{j,\ell}$ that represents the evaluation of a service $s_{j,\ell}$ on criterion c_i, for $i = 1, \ldots, p$. Consider the set $T' \subseteq T$ satisfying the control flow of *wf* and the set S of selected services, we can compute the QoS values of a workflow *wf* from the QoS values of the selected services $s_{j,\ell} \in S$ as shown in Table 1. Note that, for an exclusive pattern (XOR-split/-join pattern) there is no aggregation function between wf_i, $i = 1, \ldots, \ell$ since exactly one wf_i satisfies the control flow. For all the other patterns, the QoS of the *wf* is the QoS of the selected services for the workflows wf_i that satisfy the control flow. As shown in Table 1, we classify into five categories the most used QoS criteria according to their aggregation functions that could be summation ($\sum$ or *sum*), product ($\prod$ or *prod*), minimum (*min*) and maximum (*max*). Optimization objective could be minimization (MIN) or maximization (MAX).

Table 1. Aggregation Functions

QoS category	Sequence pattern	AND or OR pattern	Objective function
cat 1	$\sum_{t_j \in T', s_{j,\ell} \in S \cap S_j} c_i(s_{j,\ell})$	$\sum_{t_j \in T', s_{j,\ell} \in S \cap S_j} c_i(s_{j,\ell})$	Min or Max
cat 2	$\sum_{t_j \in T', s_{j,\ell} \in S \cap S_j} c_i(s_{j,\ell})$	$max\{c_i(s_{j,\ell}) : t_j \in T', s_{j,\ell} \in S \cap S_j\}$	Min
cat 3	$\prod_{t_j \in T', s_{j,\ell} \in S \cap S_j} c_i(s_{j,\ell})$	$\prod_{t_j \in T', s_{j,\ell} \in S \cap S_j} c_i(s_{j,\ell})$	Min or Max
cat 4	$min\{c_i(s_{j,\ell}) : t_j \in T', s_{j,\ell} \in S \cap S_j\}$	$min\{c_i(s_{j,\ell}) : t_j \in T', s_{j,\ell} \in S \cap S_j\}$	Max
cat 5	$max\{c_i(s_{j,\ell}) : t_j \in T', s_{j,\ell} \in S \cap S_j\}$	$max\{c_i(s_{j,\ell}) : t_j \in T', s_{j,\ell} \in S \cap S_j\}$	Min

The QoS values of a leaf in the tree of patterns are equal to the QoS values of the selected service $s_{j,\ell}$ in the candidate set S_j. The QoS values of an non-leaf node are computed based on the QoS criteria values of its children and their associated aggregation functions. The QoS values of non-leaf nodes are computed recursively.

2.3 Problem Statement

The QoS-aware Service Selection problem is an optimization problem that aims at binding every leaf node in the tree of patterns representing a workflow task to at most one of its service candidates. Formally, the problem can be stated as follows.

The QoS range values of a criterion c_i of category *cat 3* is in $\mathbb{Q} \cap (0, 1]$, and of categories *cat 1*, *cat 2*, *cat 4* and *cat 5* is in $\mathbb{N}$ for any service s.

The decision problem associated with QoS-aware Service Selection consists of p bounds $b_1, \ldots, b_p \in \mathbb{Q}$ to decide if there exists a subset of tasks $T' \subseteq T$ satisfying the control flow of *wf* and a service $s_{j,\ell} \in S_j$ associated to each task $t_j \in T'$ such that the value c_i associated with this solution is smaller than or

QoS-aware Service Selection:

Input: A workflow *wf* on a set of tasks $T = \{t_1, \ldots, t_n\}$, a set of candidate services S_j for each task $t_j \in T$ (of size at most m) and a vector $c = (c_1, \ldots, c_p)$ of p criteria such that $c_i(s) \in \mathbb{Q}$, $\forall s \in S_j$, $j = 1, \ldots, n, i = 1, \ldots, p$.

Output: A subset of tasks $T' \subseteq T$ satisfying the control flow of *wf* and a subset S of services such that a service $s_{j,\ell} \in S_j \cap S$ is associated to each task $t_j \in T'$ in order to optimize each c_i, $1 \leqslant i \leqslant p$.

equal to (resp. greater than or equal to) b_i in the case of a minimization (resp. maximization) criterion c_i, for $i = 1, \ldots, p$.

The solution for our problem is a subset $T' \subseteq T$ containing only leaves of the tree of patterns (i.e. only tasks) satisfying the control flow of *wf* in contrast with the other papers in the area where all leaves of the tree (i.e. all tasks of T) are selected in T' even if no service is selected for a task.

Before giving some complexity notions, we define the size on an input of QoS-aware Service Selection. Given an input instance I of the problem, its maximum value, denoted by $V(I)$, is $max\{c_i(s) : s \in S_j, j = 1, \ldots, n, i = 1, \ldots, p\}$ and its size, denoted by $|I|$, is $O(nm \log V(I))$.

3 Complexity Upper Bounds

In this section, we give some positive results concerning QoS-aware Service Selection. We show that the problem is polynomial time solvable if there is only one criterion and pseudo-polynomial time solvable if there are a fixed number of criteria. Moreover, we will show in the next section that pseudo-polynomiality is also tight, i.e. one cannot expect polynomiality.

When the number of criteria is one (i.e. $p = 1$), then QoS-aware Service Selection is a single criterion optimization problem and we search for an optimal solution. When the number of criteria is more than one ($p \geqslant 2$), then QoS-aware Service Selection is a p-criteria optimization problem. In this case, there is typically no optimal solution that is the best for all the criteria. Therefore, the standard situation is that any solution can always be improved on at least one criterion. The solutions of interest, called efficient solutions, are those such that any other solution which is better on one criterion is necessarily worse on at least one other criterion. The vectors associated to efficient solutions are called nondominated vectors.

Theorem 1. QoS-aware Service Selection *is solvable in linear time for instances with only one criterion.*

Theorem 2. *An instance of* QoS-aware Service Selection *on p criteria, $p \geqslant 2$, with one of them of cat 4 or cat 5, can be reduced in polynomial time to several instances on $p - 1$ criteria.*

Corollary 1. QoS-aware Service Selection *is polynomial time solvable for instances with two criteria where one of them is of cat 4 or cat 5.*

Theorem 3. *For a fixed number of criteria p,* QoS-aware Service Selection *is solvable in pseudo-polynomial time.*

4 Complexity Lower Bounds

In this section, we prove that the positive results of the previous section are tight, i.e. QoS-aware Service Selection is weakly NP-hard. We will show that it is the case even for very restricted instances.

For an instance of QoS-aware Service Selection with two criteria c_1 and c_2 we denote by Max/Max (or Min/Min) when c_1 and c_2 are to be maximized (or minimized) and by Min/Max (resp. Max/Min) when c_1 is to be minimized (resp. maximized) and c_2 to be maximized (resp. minimized).

Theorem 4. *When $p = 2$ and the criteria c_1 and c_2 are both of cat 1 or both of cat 3, then* QoS-aware Service Selection *is* NP*-hard even when*

1. *one AND pattern, 2 services per task, and* Max/Max *(or* Min/Min*, or* Min/Max*).*
2. *one sequence pattern, 2 services per task, and* Max/Max *(or* Min/Min*, or* Min/Max*).*
3. *a sequence of XOR patterns,1 service per task, and* Max/Max *(or* Min/Min*, or* Min/Max*).*
4. *one OR pattern, 1 service per task, and* Min/Max*.*

5 Conclusion

In this work, we established NP-hardness proofs for QoS-aware Service Selection, which hold even in very restrictive cases. However, we show that this hardness result is counter-balanced by the existence of a pseudo-polynomial time algorithm, which is able to solve optimally the problem in polynomial time for small values of QoS services, which is the common case in real instances.

For future work, we suggest investigating the complexity of the problem in case of two criteria of cat 1 and cat 3. However, one would easily notice that if there are three criteria (at least two of cat 1, or at least two of cat 3), then the problem is clearly NP-hard due to Theorem 4. It would also be interesting to investigate the potential relation between the shape of a workflow and the complexity of the problem. Of particular interest are cases where the problem becomes strongly NP-hard. Another interesting direction would be to establish approximation algorithms that return solutions with a priori guarantee of quality in polynomial time even for large values of QoS services.

Acknowledgements. We are grateful for the support by the bilateral research cooperation CEDRE between France and Lebanon (grant number 30885TM).

References

1. Alrifai, M., Risse, T., Nejdl, W.: A hybrid approach for efficient web service composition with end-to-end QoS constraints. ACM Trans. Web **6**(2), 7:1–7:31 (2012)
2. Ardagna, D., Pernici, B.: Global and local QoS guarantee in web service selection. In: Bussler, C.J., Haller, A. (eds.) BPM 2005. LNCS, vol. 3812, pp. 32–46. Springer, Heidelberg (2006)
3. Bonatti, P.A., Festa, P.: On optimal service selection. In: Proceedings of the 14th International Conference on World Wide Web, pp. 530–538 (2005)
4. Canfora, G., Di Penta, M., Esposito, R., Villani, M.L.: An approach for QoS-aware service composition based on genetic algorithms. In: Proceedings of the 7th GECCO, pp. 1069–1075. ACM (2005)
5. El Haddad, J.: Optimization techniques for QoS-aware workflow realization in web services context. In: Lacroix, Z., Vidal, M.E. (eds.) RED 2010. LNCS, vol. 6799, pp. 134–149. Springer, Heidelberg (2012)
6. Gabrel, V., Manouvrier, M., Murat, C.: Web services composition: complexity and models. Discrete Appl. Math. (2014). http://dx.doi.org/10.1016/j.dam.2014.10.020
7. Kiepuszewski, B., ter Hofstede, A.H.M., Bussler, C.J.: On structured workflow modelling. In: Wangler, B., Bergman, L.D. (eds.) CAiSE 2000. LNCS, vol. 1789, pp. 431–445. Springer, Heidelberg (2000)
8. Moghaddam, M., Davis, J.G.: Service selection in web service composition: a comparative review of existing approaches. In: Bouguettaya, A., Sheng, Q.Z., Daniel, F. (eds.) Web Services Foundations, pp. 321–346. Springer, New York (2014)
9. Schuller, D., Miede, A., Eckert, J., Lampe, U., Papageorgiou, A., Steinmetz, R.: QoS-based optimization of service compositions for complex workflows. In: Fantinato, M., Yang, J., Weske, M., Maglio, P.P. (eds.) ICSOC 2010. LNCS, vol. 6470, pp. 641–648. Springer, Heidelberg (2010)
10. Schuller, D., Polyvyanyy, A., García-Bañuelos, L., Schulte, S.: Optimization of complex QoS-aware service compositions. In: Kappel, G., Motahari-Nezhad, H.R., Maamar, Z. (eds.) Service Oriented Computing. LNCS, vol. 7084, pp. 452–466. Springer, Heidelberg (2011)
11. Strunk, A.: Qos-aware service composition: a survey. In: 2010 IEEE 8th European Conference on Web Services (ECOWS), pp. 67–74. IEEE (2010)
12. Trummer, I., Faltings, B., Binder, W.: Multi-objective quality-driven service selection - a fully polynomial time approximation scheme. IEEE Trans. Softw. Eng. **40**(2), 167–191 (2014)
13. van Der Aalst, W.M., Ter Hofstede, A.H., Kiepuszewski, B., Barros, A.P.: Workflow patterns. Distrib. Parallel Databases **14**(1), 5–51 (2003)
14. Yu, T., Lin, K.-J.: Service selection algorithms for web services with end-to-end QoS constraints. Inf. Syst. E-Business. Manage **3**(2), 103–126 (2005)
15. Zeng, L., Benatallah, B., Ngu, A.H.H.: QoS-aware middleware for web services composition. IEEE Trans. Softw. Eng. **30**, 311–327 (2004)
16. Zhang, W., Chang, C.K., Feng, T., Jiang, H.: QoS-based dynamic web service composition with ant colony optimization. In: 34th Annual Computer Software and Applications Conference, pp. 493–502. IEEE (2010)