





Quality of Service Ranking by Quantifying Partial Compliance of Requirements

Agustín Eloy Martínez Suñé¹  and Carlos Gustavo Lopez Pombo^{1,2} 

¹ Departamento de Computación, Universidad de Buenos Aires,
Buenos Aires, Argentina

{aemartinez, clpombo}@dc.uba.edu.ar

² Instituto de Investigación en Ciencias de la Computación (ICC), CONICET-UBA,
Buenos Aires, Argentina

Abstract. While there is not much discussion on the importance of formally describing and analysing quantitative requirements in the process of software construction; in the paradigm of API-based software systems it could be vital. Quantitative attributes can be thought as attributes determining the *Quality of Service* – QoS provided by a software component published as a service. In this sense, they play a determinant role in classifying software artifacts according to specific needs stated as requirements. In previous works we presented an efficient, and fully automatic, analysis technique for establishing *Service Level Agreements* – SLA. Such a proposal relies on describing QoS contracts as convex specifications, and compliance checking is performed by the application of an analysis algorithm based on state of the art techniques used in hybrid system verification. Such a technique succeeds in offering a procedure for determining SLA but fails in the more realistic scenario where, potentially, no service fully satisfies the requirements. In this scenario the running application may still prefer to invoke the service that offers the best chances of successfully executing with values for QoS attributes meeting the requirements satisfactorily.

In this work we propose and implement a metric for automatically quantifying partial satisfaction of QoS requirements, leading to a way of ranking services according to such notion of partial compliance.

1 Introduction

Distributed software resulting from paradigms such as *Service-Oriented Computing (SOC)* or the API's economy is based on the idea of constructing software artifacts by composing services provided by third parties and registered in

Research partly supported by the European Unions Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 778233. Carlos G. Lopez Pombo's research is supported by Universidad de Buenos Aires through grant UBACyT 20020170100544BA, and Consejo Nacional de Investigaciones Científicas y Técnicas through grant PIP 11220130100148CO.

© IFIP International Federation for Information Processing 2020

Published by Springer Nature Switzerland AG 2020

S. Bliudze and L. Bocchi (Eds.): COORDINATION 2020, LNCS 12134, pp. 181–189, 2020.

https://doi.org/10.1007/978-3-030-50029-0_12

repositories. This envisages a generation of applications which, at run-time, are transparently reconfigured by the intervention of a dedicated middleware with the capability to bind a running application that has certain requirements to a service capable of fulfilling them, subject to the negotiation of a *Service Level Agreement* – SLA; in this way, services can collectively fulfill certain business goals [4].

Requirements can be formalised as contracts between software components, and satisfaction of such contracts is usually dealt with by checking whether a judgment of the form $Pr \vdash Rq$ holds or not, where Pr is the provision contract and Rq is the requirements contract. In the literature, functional requirements have been identified as those describing what the system has to do, while non-functional ones are assumed to characterise how the system develops the behaviour described by the functional ones.

In this work we focus on *Quantitative requirements* as a proper subclass of non-functional requirements of a software system. From our perspective, quantitative requirements formalise the admissible values of specific attributes that can be interpreted over a metric space [2]. Such attributes are referred to as *Quantitative attributes*. For practical reasons, the real numbers constitute a good candidate over which these can be interpreted, formalised and analysed.

In this sense, quantitative requirements might be used to characterise the *Quality of Service* – QoS provided by a software component available as service. While services may have the same functional behaviour, they might differ on their non-functional one (for example, a service may offer low-speed computation at a very low cost while another, functionally equivalent one, might be faster but more onerous); a motivation shared with other works like [14]. Therefore, quantitative requirements might be considered as a way to classify functionally equivalent services by their QoS.

In [12] we presented an efficient, and fully automatic, analysis procedure for checking a judgment of the shape $Pr \vdash Rq$ where Pr is the provision contract and Rq is the requirements contract, thus serving the purpose of establishing SLA. Such an approach guarantees the selection of a service satisfying the requirements of the executing application but it does not provide any insight when there is no service whose provision contract is fully compliant with the requirement contract. A geometrical interpretation of the judgment $Pr \vdash Rq$ expresses that all the satisfying values of real attributes of Pr also satisfy Rq . Thus, a negative answer just means that there is at least one value satisfying Pr and not satisfying Rq pushing the application to abort its execution because there is no possible SLA on the required QoS.

In this paper we propose an automatic procedure¹ for evaluating partial compliance of QoS contracts by estimating what we call the *inclusion ratio*: the intersection of the set of satisfying values of the provision and requirement contracts interpreted relative to the size of the set of values satisfying the provision contract. This metric leads to a natural way of ranking services. While the problem of ranking services according to their QoS has been studied and different

¹ A demonstration video of the tool can be found on <http://bit.do/QoS-Rank>.

approaches to tackle the problem have been proposed, in general, these are based on monitoring run-time behavior of specific attributes [1,9] or by socially sharing the users' opinion [11,15] but, to the best of our knowledge, there is none focussing on QoS ranking where the QoS is formally specified and SLA is guaranteed through an automated procedure.

2 Formalisation and Analysis of Quantitative Attributes

The selection of a service requires establishing an SLA, part of which is the QoS agreement. Given a formal QoS requirement contract Rq ; a service that declares a formal QoS provision contract Pr will be a good candidate only if $Pr \implies Rq$ holds. In [12] we adopted *monotone Satisfiability Modulo Convex (SMC) formulae* [13] as specification language and proposed a method to check QoS compliance. *SMC formulae* are defined as quantifier-free formulae in conjunctive normal form, with atomic propositions ranging over a subset of the propositional variables and convex constraints². Then, given two SMC formulas Pr and Rq , the proposed method to check whether $Pr \vdash Rq$ holds or not is done by determining if the formula $Pr \wedge \neg Rq$ is not satisfiable.

Such a procedure does not provide any hint on what to do when no service registered in the repository fully satisfies the requirements. That is, no QoS provision contract satisfies the QoS requirements contract.

A different way of formalising QoS contracts is as a set of real values satisfying the specification. Next we present this alternative formalisation of quantitative non-functional requirements. Throughout the rest of the paper, letters \mathcal{I} , \mathcal{J} and \mathcal{K} will be used to denote index sets.

Definition 1. *Let \mathcal{X} be a set of real variables. We define a QoS specification on quantitative attributes \mathcal{X} to be $\langle \mathcal{X}, \alpha \rangle$, with α being a formula with shape $\bigvee_{i \in \mathcal{I}} \mathbb{P}_i$. Each \mathbb{P}_i is a convex polytope expressed as $\bigwedge_{k \in \mathcal{K}_i} f_{i,k}(\vec{x}) \mathcal{R} 0$, where $\mathcal{R} \in \{<, \leq\}$ such that each $f_{i,k}(\vec{x}) \mathcal{R} 0$ is a convex constraint. The set of QoS specifications over the set of quantitative attributes \mathcal{X} will be referred to as $QoS_spec(\mathcal{X})$.*

Let $\langle \mathcal{X}, \alpha \rangle \in QoS_spec(\mathcal{X})$. We define a valuation for α as $v : \mathcal{X} \rightarrow \mathbb{R}$. Satisfaction is a relation $\models \subseteq [\mathcal{X} \rightarrow \mathbb{R}] \times QoS_spec(\mathcal{X})$ defined as $\vec{x} \models \alpha$ if and only if there exists $i \in \mathcal{I}$, such that for all $k \in \mathcal{K}_i$, $f_{i,k}(\vec{x}) \mathcal{R} 0$. The set of values satisfying α is defined as $\llbracket \alpha \rrbracket = \{ \vec{x} \mid \vec{x} \models \alpha \}$.

Essentially, the set of admissible values of the quantitative attributes \mathcal{X} in a QoS_spec is characterised by a set of convex polytopes over \mathcal{X} . Notice that, under some reasonable assumptions, *SMC formulae* can be translated to specifications in $QoS_spec(\mathcal{X})$ by simply iterating the satisfying valuations of the boolean abstraction of the formulae, and collecting the polytopes determined by the positive literals.

² The interested reader will find the formal definition in [13].

A specification like this can capture many quantitative aspects of a software system. For example, the formulae below show a specification of an API-based software application requiring a service to be paid exclusively for the time it is used. Consider as relevant quantitative attributes a) *perSec*: the cost per second of the session, and b) *maxWait*: the maximum waiting time. Then, such attributes can be formalised by a QoS specification $\langle\langle\mathcal{X}\rangle, \alpha\rangle$ where $\mathcal{X} = \{perSec, maxWait\}$. To characterise a service that is more expensive when it is faster a possible α would be the disjunction of the following formulae:

$$\begin{aligned} 0 < maxWait \leq 100 \wedge 0.1 \leq perSec < 0.3, \\ 100 < maxWait \leq 1000 \wedge 0.0 \leq perSec \leq 0.1 \end{aligned}$$

A geometrical interpretation of the formula $Pr \implies Rq$ leads to the possibility of measuring different degrees of partial compliance, which range between total compliance (all values allowed by Pr are accepted by Rq) and no compliance at all (no value allowed by Pr is accepted by Rq).

Based on the previous definitions, proving the formula $Pr \vdash Rq$ is to check whether $\llbracket Pr \rrbracket \subseteq \llbracket Rq \rrbracket$. We propose to compute the volume of the intersection between $\llbracket Pr \rrbracket$ and $\llbracket Rq \rrbracket$ relative to the volume of $\llbracket Pr \rrbracket$, referred to as *inclusion ratio*. This indicator quantifies what is the percentage of the QoS values allowed by the provision contract that is actually accepted by the requirements contract. For us, this indicator serves the purpose of quantifying the partial compliance of Rq by Pr ; therefore, functionally compliant services can be chosen from a ranking built by using inclusion ratio as the ordering criterion.

2.1 On the Complexity of the Volume Computation

In the literature, the volume of a polytope is invariably defined in terms of the sum of the volumes of a certain decomposition of such polytope into a family of convex ones [6]. Fortunately, Definition 1 assists us by already expressing the set of values in a QoS contract as the union of the sets of values of each member of a family of convex polytopes, expressed as $Pr = \bigvee_{i \in \mathcal{I}} P_i$ and $Rq = \bigvee_{j \in \mathcal{J}} R_j$. As neither family is guaranteed to be a partition, volumes must be computed following the *Principle of inclusion-exclusion* [10, Chap. 4]. Then, the inclusion ratio is calculated as follows:

First: Compute the volume (denoted as $\#$) of intersection between Pr and Rq :

$$\# \left(\bigcup_{i \in \mathcal{I}} \llbracket P_i \rrbracket \cap \bigcup_{j \in \mathcal{J}} \llbracket R_j \rrbracket \right) = \sum_{\emptyset \neq \mathcal{K} \subseteq \mathcal{I} \times \mathcal{J}} (-1)^{|\mathcal{K}|-1} \# \left(\bigcap_{(i,j) \in \mathcal{K}} (\llbracket P_i \rrbracket \cap \llbracket R_j \rrbracket) \right) \quad (1)$$

Second: Compute the volume of P :

$$\# \left(\bigcup_{i \in \mathcal{I}} \llbracket P_i \rrbracket \right) = \sum_{\emptyset \neq \mathcal{K} \subseteq \mathcal{I}} (-1)^{|\mathcal{K}|-1} \# \left(\bigcap_{k \in \mathcal{K}} \llbracket P_k \rrbracket \right) \quad (2)$$

Third: Compute the inclusion ratio of Pr in Rq as: $\frac{\#(\llbracket Pr \rrbracket \cap \llbracket Rq \rrbracket)}{\#(\llbracket Pr \rrbracket)}$.

In the late 80's Dyer et al. [3] and Khachiyan [8] proved the complexity of computing the volume of a convex polytope to be **#P-Hard**. The first estimation algorithm was presented by Kannan et al. in [7]. Later in [5], an estimation algorithm based on the *Multiphase Monte-Carlo algorithm* is proposed and the authors show that it can efficiently handle instances of dozens of dimensions with high accuracy. The complexity is shown to be $O^*(m \cdot n^3)$, where n is the dimensions, m is the number of constraints, and the Soft- O notation (O^*) omits logarithmic factors. The estimation method tool is distributed under the name **PolyVest**.

The implementation of the algorithm presented above for computing the inclusion ratio requires $2^{|\mathcal{I}| \cdot |\mathcal{I}|} + 2^{|\mathcal{I}|}$ invocations to the function computing the volume of a convex polytope. Such complexity forces us to consider a further dimension of approximation in order to make the calculation of the inclusion ratio viable in practice. In the next section we study the appropriateness of considering only some of the intersections involved in the calculation as an estimation of the volume.

3 Experimental Results

Our first research question is related to the complexity of the exact computation of the volume of a polytope \mathcal{P} . For the sake of understanding the limits imposed by such complexity, the research question will be stated in terms of a single polytope decomposed as a family of convex polytopes.

RQ1: How does the complexity of $\#(\bigcup_{i \in \mathcal{I}} \mathcal{P}_i)$ impacts in practice?

From our experiments³ it is possible to identify the exponential nature of the computation and that the majority of cases could not be completed within the time budget, hence the method cannot be applied to more realistic case-studies. The applicability of the technique, therefore, rests on the possibility of computing a good and efficient approximation of the volume operator. A natural way of doing this is by considering only a limited number of terms of the sum in Eq. 1. The next proposition formalises the intuition that subsequent partial sums represent an upper, and a lower, bound for the exact value of the volume.

Proposition 1. *Let $P = \{\mathcal{P}_i\}_{i \in \mathcal{I}}$ and $0 < n < |\mathcal{I}|$, with $n \equiv 0 \pmod{2}$ then*

$$\sum_{\substack{\emptyset \neq \mathcal{K} \subseteq \mathcal{I} \\ 0 < |\mathcal{K}| \leq n}} (-1)^{|\mathcal{K}|-1} \# \left(\bigcap_{k \in \mathcal{K}} \mathbb{P}_k \right) \leq \#(P) \leq \sum_{\substack{\emptyset \neq \mathcal{K} \subseteq \mathcal{I} \\ 0 < |\mathcal{K}| \leq n+1}} (-1)^{|\mathcal{K}|-1} \# \left(\bigcap_{k \in \mathcal{K}} \mathbb{P}_k \right).$$

Computing an approximation of a magnitude requires the introduction of a metric for measuring the effectiveness of such estimation. As usual, the relative error between the magnitude x and an approximate value x_0 is defined to be $\frac{|x_0 - x|}{|x|}$.

³ Synthetic case-studies were generated automatically consisting of families of 5–11 convex polytopes over 10–35 quantitative attributes. For each case the procedure was run over 20 different instances, with a 1 h time budget.

The next proposition provides upper bounds for the relative error that do not depend on the exact value v of the volume but on two consecutive approximations: $\{v_{k-1}, v_k\}$ or $\{v_k, v_{k+1}\}$.

Proposition 2. *Let $v, v_{k-1}, v_k, v_{k+1} \in \mathbb{R}$, then:*

- if $0 < v_{k-1} < v < v_k$, then $\frac{v_k - v}{v} \leq \frac{v_k - v_{k-1}}{v_{k-1}}$ (denoted ρ_k),
- if $v_k > v > v_{k+1} > 0$, then $\frac{v_k - v}{v} \leq \frac{v_k - v_{k+1}}{v_{k+1}}$ (denoted σ_k)
- if $v_{k-1} \geq v_{k+1} > 0$, then $\rho_k \geq \sigma_k$.

Using these bounds, and considering Proposition 1, we can design an algorithm for computing the approximate volume of a polytope, by computing successive approximations until the desired value for the relative error is guaranteed. Notice that if we require a relative error equal to 0, the complexity of the algorithm is still $O(2^n)$, but when the bound is big enough to let the algorithm converge in the second approximation, the complexity drops to $O(n^2)$. Our second research question analyses the practical impact of approximating the volume of a polytope using these bounds.

RQ2: How do the partial sums of the terms in the sum of Eq.1 converge to the value $\#(\bigcup_{i \in \mathcal{I}} \mathcal{P}_i)$ with 1% and 5% tolerance?

Results⁴ show that the number of successive approximations needed for a calculation with an error within 1% tolerance is between 2 and 5, while for an error within 5% tolerance is between 2 and 3. In both scenarios the estimated computation greatly outperformed the exact one, being able to compute the majority of cases within the time budget.

A natural follow-up from these results is to explore the scalability of the procedure for volume estimation. Table 1 show the maximum size of the family of polytopes whose approximate volume can be computed in one hour, considering relative errors with bounds set to 1% and 5%. *Size* denotes the number of polytopes in the family and *App* the number of successive approximations needed in each case.

As we mentioned in Sect. 2, efficiently computing the volume of these types of polytopes is just a means for quantifying partial compliance of QoS contracts.

RQ3: how does the performance gain obtained by approximating the volume impacts the computation of the inclusion ratio?

Our results⁵ show that as the size of the contracts increases the efficiency gain between 0%, 1% and 5% tolerance is dramatically better. There are cases where the approximate volume computation is performed in 4% of the time required by the exact volume computation, and several cases for which the exact inclusion ratio fails to be computed within the 10 h time budget but becomes feasible when a bound for the relative error is introduced.

⁴ The experiment was conducted over the same dataset as for **RQ1**.

⁵ Case studies were generated following the same guidelines as for **RQ2**. A time budget of 10 h is considered.

Table 1. Upper limit for computation of approximated volume.

	Number of quantitative attributes								
	10			15			20		
	Size	App.	Time	Size	App.	Time	Size	App.	Time
1%	11	4.95	1452.42	15	3	3067.82	22	2	3320.67
5%	15	3.72	1857.42	24.92	2	1218.55	22	2	3320.67

	Number of quantitative attributes					
	25			30		
	Size	App.	Time	Size	App.	Time
1%	14	2	3481.29	9	2	3120.49
5%	14	2	3481.29	9	2	3120.49

Inclusion ratio provides a way of ranking services according to partial satisfaction. Thus, it is worth knowing how an approximate computation of the volume propagates the relative error to the computation of the inclusion ratio. In the end, this will impact the actual selection of a service candidate. The next proposition establishes lower and upper bounds for the distance between approximate inclusion ratio and exact inclusion ratio.

Proposition 3. *Given two volume approximations v_k, w_k , where the exact volumes are v, k , respectively. Let e be the bound for the relative error (i.e., $\frac{|v-v_k|}{|v|} \leq e$ and $\frac{|w-w_k|}{|w|} \leq e$) then:*

1. $(1 - e) \cdot v \leq v_k \leq (1 + e) \cdot v$ (respectively $(1 - e) \cdot w \leq w_k \leq (1 + e) \cdot w$), and
2. $\frac{(1-e) \cdot v}{(1+e) \cdot w} \leq \frac{v_k}{w_k} \leq \frac{(1+e) \cdot v}{(1-e) \cdot w}$ (equivalently $\frac{1-e}{1+e} \cdot \frac{v}{w} \leq \frac{v_k}{w_k} \leq \frac{1+e}{1-e} \cdot \frac{v}{w}$).

Consider, as an example, the bounds obtained from Proposition 3 for the approximate inclusion ratio, when approximate volumes are computed with relative error bound to be smaller than 5% and 1%, as we did in the previous experiments. Let eir be the exact inclusion ratio, $ir5$ the inclusion ratio computed with a relative error below 5% and $ir1$ the inclusion ratio computed with a relative error below 1%; then, we obtain that $0.905 \cdot eir \approx \frac{0.95}{1.05} \cdot eir \leq ir5 \leq \frac{1.05}{0.95} \cdot eir \approx 1.105 \cdot eir$ and $0.98 \cdot eir \approx \frac{0.99}{1.01} \cdot eir \leq ir1 \leq \frac{1.01}{0.99} \cdot eir \approx 1.02 \cdot eir$, respectively.

Our final research question focuses on effectively ranking different provision contracts using the inclusion ratio as a metric for quantifying partial satisfaction.

RQ4: What is the impact of the proposed approximation in the construction of a ranking of a set of services?

For this experiment we performed 3 examples of ranking 20 provision contracts by partial satisfaction of a single requirements contract. The datasets were synthetically generated following the guidelines used for generating the case-studies used in **RQ3**. Each ranking correspond to a different size of the families of polytopes as follows: a) $|Pr| = 2$, $|Rq| = 4$, b) $|Pr| = 3$, $|Rq| = 3$, c) $|Pr| = 4$, $|Rq| = 2$.

For each dataset we performed the ranking order using the exact inclusion ratio, and also using 1% and 5% bounds for the relative error. The results of the experiment show that in all of the examples the ranking obtained by using the exact volume computation is preserved when resorting to approximate volume computations.

4 Conclusions and Further Work

We extended previous work where we proposed a formalisation of QoS contract as convex specifications [13]. Under a geometrical interpretation, a contract can be seen as a family of convex polytopes characterising the admissible values for the quantitative attributes and a judgement of the shape $Pr \vdash Rq$ as polytope inclusion. From this perspective, we proposed a way of QoS ranking services by quantifying the volume of the intersection between $\llbracket Pr \rrbracket$ and $\llbracket Rq \rrbracket$, under the name *inclusion ratio*. Since this indicator quantifies the percentage of values allowed by the provision contract that are actually accepted by the requirement contract we sustain that it serves the purpose of quantifying partial compliance. The exponential nature of computing the volume of a family of convex polytopes forced us to propose a volume approximation technique based on imposing an upper bound to the relative error.

We evaluated the performance of the approximated volume computation against the exact volume computation, the scalability of the approximate volume computation and the performance of computing the inclusion ratio with respect to the exact computation. Finally, we showed examples of the impact of approximate inclusion ratio in ranking services by comparing the ideal ranking, obtained by resorting to the exact inclusion ratio, and the one obtained by resorting to approximate inclusion ratio, for different bounds to the relative error.

This technique for evaluating partial satisfaction of QoS contracts triggers what we believe is one of the most important questions regarding the automation of service broking. Under partial satisfaction of QoS contracts, we would like to establish preferences among different quantitative attributes of the different candidates. For example, if we consider two services with the same inclusion ratio, a low budget application would like to express its preference for cheaper services, even at the cost of degraded performance, over a more expensive and efficient one. This problem will be addressed in the near future aiming at a more realistic view of an automatic procedure for checking QoS compliance for determining SLA in a service-based architecture.

References

1. Al-Masri, E., Mahmood, Q.H.: QoS-based discovery and ranking of web services. In: Guo, K. (ed.) Proceedings of 16th International Conference on Computer Communications and Networks, IEEE ICCCN 2007, pp. 529–534. IEEE Computer Society, August 2007

2. Bryant, V.: Metric Spaces: Iteration and Application. Mathematical Systems Theory. Cambridge University Press, Cambridge (1985)
3. Dyer, M.E., Frieze, A.M.: On the complexity of computing the volume of a polyhedron. *SIAM J.* **17**(5), 967–974 (1988)
4. Fiadeiro, J.L., Lopes, A., Bocchi, L.: An abstract model of service discovery and binding. *Formal Aspects Comput.* **23**(4), 433–463 (2011). <https://doi.org/10.1007/s00165-010-0166-z>
5. Ge, C., Ma, F.: A fast and practical method to estimate volumes of convex polytopes. In: Wang, J., Yap, C. (eds.) FAW 2015. LNCS, vol. 9130, pp. 52–65. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19647-3_6
6. Gritzmann, P., Klee, V.: On the complexity of some basic problems in computational convexity. In: Bisztriczky, T., McMullen, P., Schneider, R., Weiss, A.I. (eds.) Polytopes: Abstract, Convex and Computational. ASIC, vol. 440, pp. 373–466. Springer, Dordrecht (1994). https://doi.org/10.1007/978-94-011-0924-6_17
7. Kannan, R., Lovász, L., Simonovits, M.: Random walks and an $o^*(n^5)$ volume algorithm for convex bodies. *Random Struct. Algorithms* **11**(1), 1–50 (1991)
8. Khachiyan, L.G.: The problem of calculating the volume of a polyhedron is enumerably hard. *Russ. Math. Surv.* **44**(3), 199–200 (1989)
9. Liu, Y., Ngu, A.H., Zeng, L.: QoS computation and policing in dynamic web service selection. In: Feldman, S.I., Uretsky, M., Najork, M., Wills, C.E. (eds.) Proceedings of 13th International Conference on World Wide Web - WWW 2004, pp. 66–73. ACM Press, May 2004
10. Liu, Z.: Introduction to Combinatorial Mathematics. McGraw-Hill Book Company, New York City (1968)
11. Mao, C., Chen, J., Towey, D., Chen, J., Xie, X.: Search-based QoS ranking prediction for web services in cloud environments. *Future Gener. Comput. Syst.* **50**, 111–126 (2015)
12. Martínez Suñé, A.E., Lopez Pombo, C.G.: Automatic quality-of-service evaluation in service-oriented computing. In: Riis Nielson, H., Tuosto, E. (eds.) COORDINATION 2019. LNCS, vol. 11533, pp. 221–236. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22397-7_13
13. Shoukry, Y., Nuzzo, P., Sangiovanni-Vincentelli, A.L., Seshia, S.A., Pappas, G.J., Tabuada, P.: SMC: satisfiability modulo convex optimization. In: Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control, pp. 19–28. ACM Press, New York (2017)
14. Strunk, A.: QoS-aware service composition: a survey. In: Brogi, A., Pautasso, C., Papadopoulos, G.A. (eds.) Proceedings of 8th IEEE European Conference on Web Services, ECOWS 2010, pp. 67–74. IEEE Computer Society, December 2010
15. Zheng, Z., Wu, X., Zhang, Y., Lyu, M.R., Wang, J.: QoS ranking prediction for cloud services. *IEEE Trans. Parallel Distrib. Syst.* **24**(6), 1–50 (2013)