2019 IEEE International Conference on Web Services (ICWS)

FASS: A Fairness-Aware Approach for Concurrent Service Selection with Constraints

Songyuan Li[†], Jiwei Huang[‡*], Bo Cheng[†], Lizhen Cui[§], Yuliang Shi[§‖]
[†]*State Key Laboratory of Networking and Switching Technology*
Beijing University of Posts and Telecommunications, Beijing 100876, China
[‡]*Department of Computer Science and Technology, China University of Petroleum - Beijing, Beijing 102249, China*
[§]*School of Software, Shandong University, Jinan 250101, China*
[‖]*Dareway Software Co., Ltd., Jinan 250101, China*
Email: lisy@bupt.edu.cn, huangjw@cup.edu.cn, chengbo@bupt.edu.cn, clz@sdu.edu.cn, shiyuliang@sdu.edu.cn

***Abstract*—The increasing momentum of service-oriented architecture has led to the emergence of divergent delivered services, where service selection is meritedly required to obtain the target service fulfilling the requirements from both users and service providers. Despite many existing works have extensively handled the issue of service selection, it remains an open question in the case where requests from multiple users are performed simultaneously by a certain set of shared candidate services. Meanwhile, there exist some constraints enforced on the context of service selection, e.g. service placement location and contracts between users and service providers. In this paper, we focus on the QoS-aware service selection with constraints from a fairness aspect, with the objective of achieving max-min fairness across multiple service requests sharing candidate service sets. To be more specific, we formulate this problem as a lexicographical maximization problem, which is far from trivial to deal with practically due to its inherently multi-objective and discrete nature. A fairness-aware algorithm for concurrent service selection (FASS) is proposed, whose basic idea is to iteratively solve the single-objective subproblems by transforming them into linear programming problems. Experimental results based on real-world datasets also validate the effectiveness and practicality of our proposed approach.**

***Keywords*-service selection; Quality of Service (QoS); max-min fairness; selection constraints; concurrent service execution.**

I. Introduction

Nowadays, service selection has become a key building block of Service-Oriented Architecture (SOA) along with the prevalence of services computing technology. It implies the process of gaining target service from various candidate services, whose objective is to match both functional and non-functional requirements. With the increasing scale of web services, candidate services with equivalent functionality are simultaneously provided for selection, but vary in non-functional properties (i.e., Quality of Service (QoS)).

The common goal of QoS-aware service selection is to elect the target service with the optimal end-to-end QoS, which is inherently an optimization problem. There have been a great number of existing works [1], [2] proposing efficient service selection schemes, especially for web and cloud systems. While most existing work in the literatures primarily deals with finding the single target service from candidate services for one user, however, little focus has been on the service selection scenario with multiple service requests addressed by users simultaneously [3]. Multiple service requests submitted by users are required for concurrent service running at the service platform.

For this case, service requests proposed by divergent users may have various constraints. For example, when mobile communication users request for establishing links with the base station (BS), there have been the selection rule (e.g. location-aware [4]) restricting the range of deliverable BS. In the fields of content distribution, content users attributable to multiple Internet Service Providers have hard constraints about the Content Distribution Netoworks that they can access to [5]. Besides, users and service providers reach an agreement in contract, specifying that users can merely use the paid services. Therefore, the constraints should be fully considered especially for concurrent service selection.

Furthermore, multiple service requests may share the limited amount of candidate services with the identical functional capacities but different QoS levels. Given this, multiple service requests are inherently competing for the candidate services with each other for the purpose of obtaining a higher QoS. Therefore, it necessitates a fairness-aware selection mechanism when pooling the candidate services.

In this paper, we put forward a fairness-aware service selection scheme, addressing the problem of multiple QoS-aware service selection with constraints. Our service selection approach is carefully designed from the perspective of service ecosystem [6]. On the one hand, users are usually willing to gain a better service with a higher QoS at a reasonable price. On the other hand, each service request gains a better service with a higher QoS without degrading the QoS of other service requests, which ensures the fairness of concurrent service selection. It is helpful for holding all the existing users in the ecosystem and attracting more users from the outside with the fair policy. With the growing scale of users, service providers will gain more revenue motivating them to develop services with higher QoS. In this way, the loop of sustainable SOA development is built up.

Highlights of our contributions are as follows. We firstly

*Jiwei Huang is the corresponding author. Email: huangjw@cup.edu.cn.

978-1-7281-2717-0/19/$31.00 ©2019 IEEE
DOI 10.1109/ICWS.2019.00051

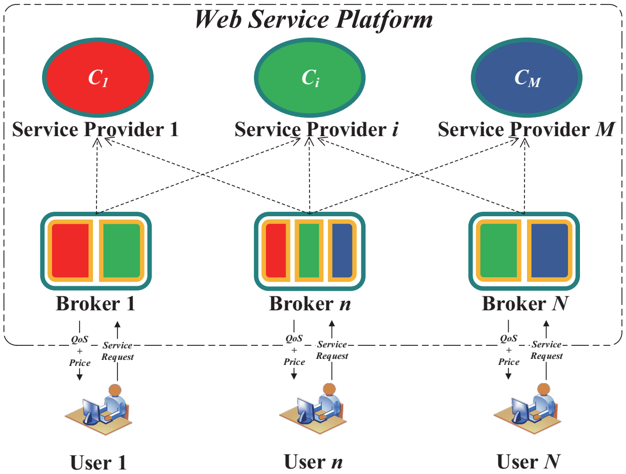

Figure 1: Overview of Service Selection Model.

outline our basic model of multiple service selection with constraints and formulate the max-min fairness (MMF) optimization objective as a lexicographical maximization problem. In virtue of the multi-objective and discrete characteristics of the lexicographical problem, it is often a puzzle to work out the exact solution. Through extensively investigating the structure of lexicographical problem, we find out the properties of separable convex objective and totally unimodular linear constraints. Thanks to these two properties, we transform the lexicographical maximization problem into a range of equivalent linear programming (LP) subproblems. The target services for multiple service requests through finite iterations of LP, where MMF is achieved. Finally, our proposed approach is validated through experiments.

II. Problem Formulation

A. Concurrent Service Selection Model with Constraints

We consider a set of multiple service requests $\mathcal{N} = \{1, ..., N\}$ submitted by users to a web service platform for concurrent execution, as depicted in Fig. 1. Substaintial candidate services are released by third-party service providers. Given the QoS preference for each service request, the service broker is responsible for finding out the personalized target service from the numerous released services.

Without loss of generality, it is assumed that the candidate service sets from M third-party service providers can be categorised into $\mathcal{M} = \{C_1, ...C_M\}$. The candidate set contains a variety of services $j \in C_i$, where $1 \leq i \leq M$. As discussed above, the candidate set are sharable with constraints amongst multiple service request. The service selection constraint for service request n is indicated by the constraint set S_n. The element $i \in S_n$ implies the enabled types of services which service request n can elect. From the standpoint of service providers, the set of service requests authorized by provider i is characterized with $\mathcal{N}_i$. For simplicity, response time is applied as the only QoS criteria in this paper. For each service j in C_i, the response time is measured as the value of $Q_{i,j}$.

The selection of candidate service is formulated by a binary variable $x^n_{i,j}$, where 1 means the j^{th} service in the candidate set C_i is elected by the service request n and 0 indicates the opposite. Decision variables of the service request n is represented by $\boldsymbol{x}_n = \{x^n_{i,j} | i \in S_n, j \in C_i\}$, and all of variables $x^n_{i,j}$ forms the solution space $\boldsymbol{\Theta}$. The execution time τ_n for service request n is calculated as (1).

$$\tau_n = \sum_{i \in S_n} \sum_{j \in C_i} x^n_{i,j} Q_{i,j} \tag{1}$$

Given diverse QoS requirements from users, a tailored Service Level Agreement (SLA) is highly required for a flexible service selection scheme [7]. To be more specific, an SLA is defined by the QoS committed by the service provider and associated payment which the user is obliged to afford. In this work, we assume that the user pays for the service in the pattern of pay-per-use. The pricing model of pay-per-use has been widely accepted in the field of cloud service [8], and so is in the case for service computing [9]. Customers wish to be served by a better service with a higher QoS even though they are reasonably asked for more money. In the pay-per-use model, the payment for service request n mainly consists of two parts, one of which is the basic payment a_n for launching the service which the another is the maximum extra bonus b_n for delivering a better service . A baseline of QoS criteria (i.e., response time) $Q^{(ref)}_n$ is addressed here, reflecting the user n' basic QoS requirements. If user n obtains a service outperforming the QoS baseline $Q^{(ref)}_n$, then a basic payment a_n and an extra bonus should be charged. Otherwise, the user n will pay at most a_n without any extra bonus. Therefore, the user n's payment is calculated as (2) when selecting the j^{th} candidate service from the candidate set C_i.

$$\pi^n_{i,j} = a_n + b_n \cdot (1 - \frac{Q_{i,j}}{Q^{(ref)}_n} \cdot x^n_{i,j}) \tag{2}$$

Basic payment a_n is positively correlated to the severity of QoS requirements (i.e., QoS baseline $Q^{(ref)}_n$). Extra bonus is considered to be linearly increasing with QoS improvements [10]. The user n's payment is expressed by (3).

$$\pi_n = a_n + b_n \cdot (1 - \sum_{i \in S_n} \sum_{j \in C_i} \frac{Q_{i,j}}{Q^{(ref)}_n} \cdot x^n_{i,j}) \tag{3}$$

B. Lexicographical Problem Formulation Achieving MMF

Since the candidate services released by service providers are shared by multiple service requests waiting for concurrent execution, our design purpose is to take each service request into consideration and motivate all of them to obtain the target service with high and acceptably fair QoS. To be more specific, our service selection scheme applies max-min fairness (MMF) across multiple service requests.

Definition 1 (Max-Min Fairness): A service selection scheme satisfies max-min fairmess (MMF), if it is impossible to increase the i^{th} lowest payment across N service requests even though removing the service requests whose payment is strictly higher than the i^{th} lowest payment, note that $i \in \mathcal{N}$.

In the context of concurrent service selection, we seeks to maximize the lowest payment amongst the multiple requests,then to optimize the second lowest without impacting the

Algorithm 1 FASS: Service Selection across Multiple Requests with Max-Min Fairness.

```
Initialize $\widetilde{\mathcal{N}} \leftarrow \mathcal{N}$;
while $\widetilde{\mathcal{N}} \neq \emptyset$ do
    $\boldsymbol{x} \leftarrow LP(\mathcal{A}, \mathcal{B}, \mathcal{Q}^{(ref)}, \mathcal{Q}, \Theta)$;
    $\boldsymbol{x}_{n^*} \leftarrow \underset{n \in \mathcal{N}}{\operatorname{argmin}}\ \pi_n$ ;
    Fix the variable subset $\boldsymbol{x}_{n^*}$ ;
    Set $x^n_{i,j} \leftarrow 0$, in the case of arbitrary $n \neq n^*$ ;
    $\Theta \leftarrow \Theta \backslash \{x^n_{i,j} \,|\, n = n^*\}$
    $\Theta \leftarrow \Theta \cap \{\, x^n_{i,j} \,|\, x^{n^*}_{i,j} = 1, i \in S_{n^*}, j \in C_i\}$ ;
    $\widetilde{\mathcal{N}} \leftarrow \widetilde{\mathcal{N}} \backslash \{n^*\}$;
end while
return $x^n_{i,j}, \forall n \in \mathcal{N}, i \in S_n, j \in C_i$;
```

previous one, and so forth. Until all the service requests have been optimized, the procedure of service selection will be terminated with an MMF service selection scheme obtained.

In the area of multi-criteria optimization, lexicographical techniques [11] grants the highest optimization priority to the most important objective, matching the interests of max-min fairness. As a result, our service selection scheme based on max-min fairness can be rigorously formulated as a lexicographical maximization problem, theoretically defined as the objective function (4) subject to (5)-(7). In the scenario of our work, there exist two main types of constraints which are user constraints and provider constraints. The user constraints (5) ensure that each customer's request should elect just only one service from available candidates of her own. The provider constraints (6) imply that different user has to select different services from service providers.

$$\underset{x^n_{i,j} \in \boldsymbol{\Theta}}{\textit{lexmax}} \quad \boldsymbol{\pi} = (\pi_1, \pi_2, ..., \pi_N) \tag{4}$$

subject to,

$$\sum_{i \in S_n} \sum_{j \in C_i} x^n_{i,j} = 1, \ \forall n \in \mathcal{N} \tag{5}$$

$$\sum_{n \in \mathcal{N}_i} x^n_{i,j} \leq 1, \ \forall i \in \mathcal{M}, \forall j \in C_i \tag{6}$$

$$x^n_{i,j} \in \{0, 1\}, \ \forall n \in \mathcal{N}, \, i \in S_n, \, j \in C_i \tag{7}$$

The objective in the lexicographical maximization problem is a payment vector $\boldsymbol{\pi} \in \mathbb{R}^N$, each element of which represents the payment of a specified user submitting the service request n. Optimal $\boldsymbol{\pi}^*$ is lexicographically no smaller than any feasible $\boldsymbol{\pi}$. It signifies that the first smallest element of $\boldsymbol{\pi}^*$ (i.e., the lowest payment across multiple requests) should be the maximum amongst all feasible solutions $\boldsymbol{\pi}$. In the case of all $\boldsymbol{\pi}$ with the same lowest payment, the second lowest payment in $\boldsymbol{\pi}^*$ is applied for maximization. The rest is in a similar fashion. Through solving this lexicographical problem iteratively, an optimal service selection plan is worked out achieving the max-min fairness.

III. Computing Service Selection Plan

A. Iterative MMF Optimization Framework

An iterative MMF optimization framework namely *FASS* is put forward in the first step. Both payment parameters and QoS baselines are tracked for each service request. The service assignments for all N service requests are iteratively accomplished one after another according to the non-decreasingly order of service payments. In the first round of iterations, the service request n^* with the lowest payment is prioritized for service selection and payment optimization, treated as a subproblem implemented by a Linear Programming (LP) problem in the Section III-B.

Once the candidate service optimizing the service request n^*'s payment is picked out, there are several settings ready for the next iteration round. In brief, we freeze the service assignment of optimized request n^*. First, the family of decision variables $\{x^n_{i,j} \,|\, n = n^*\}$ holds as unchanged, and lowers the dimension of the solution space $\boldsymbol{\Theta}$ by one. Second, the solution space $\boldsymbol{\Theta}$ should be also reduced by the decision variables relevant to the selected candidate services, formulated by $\{\, x^n_{i,j} \,|\, x^{n^*}_{i,j} = 1, i \in S_{n^*}, j \in C_i\}$. After the service request with the lowest payment having been optimized, the next round is launched aimed to optimizing the service request with the second lowest payment. Preparing for the afterwards round, we conduct the settings of solution space $\boldsymbol{\Theta}$ analogous to what is done at the first round.

Such iterative process repeats until all the service requests obtain the target service of her own. The iteration algorithm terminates, indicating the arise of concurrent service selection scheme with max-min fairness. It should be noticed that the service selection scheme is obtained through deterministic finite iterative rounds. Algorithm 1 illustrates pseudocode for concurrent service selection achieving MMF.

B. LP Transformation Towards the Lowest Payment Maximization

The lexicographical optimization problem (4) is an integer optimization with multi-objectives, which is NP-hard to solve the problem directly. Given that, we resolve the problem (4) into N single-objective subproblems optimizing the lowest payment. The optimization goal of single-objective subproblem is formulated as (8).

$$\underset{x^n_{i,j} \in \boldsymbol{\Theta}}{\textit{max}} \ \underset{n \in \mathcal{N}}{\textit{min}} \, (a_n + b_n \times (1 - \frac{\tau_n}{Q^{(ref)}_n})) \tag{8}$$

Thanks to the possible value for $x^n_{i,j}$ confined to $\{0, 1\}$, the execution time τ_n for service request n, previously formulated by (1), can be also expressed as (9).

$$\tau_n = \underset{i \in S_n, j \in C_i}{\textit{max}} \ x^n_{i,j} Q_{i,j} \tag{9}$$

Then, let (9) substituted into the objective function (8), then we have the single-objective problem represented in another non-linear form (10), subject to (5)-(7).

$$\underset{x^n_{i,j} \in \boldsymbol{\Theta}}{\textit{max}} \ \underset{n \in \mathcal{N}, i \in S_n, j \in C_i}{\textit{min}} \ a_n + b_n \cdot (1 - \frac{Q_{i,j}}{Q^{(ref)}_n} \cdot x^n_{i,j}) \tag{10}$$

Integral Optimum Guarantee. A linear programming problem will yield an optimal solution in integers, if it has a totally unimodular (TU) coefficient matrix [12]. In our problem domain, the coefficient matrix of constraints (5) and (6) is carefully investigated and verified the property of total unimodularity, in order to further determine whether the deletion of the integrality constriants (7) impacts on the optimal service selection of the problem (4).

Lemma 1: The matrix formed by the coefficients of constraints (5) and (6) is total unimodular.

The proof of Lemma 1 can be found in a longer version of this paper [13]. It follows that our problem has an integral

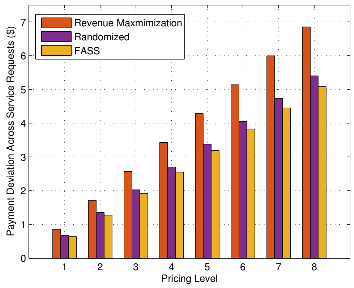

Figure 2: Payment Deviation under Different Algorithms.

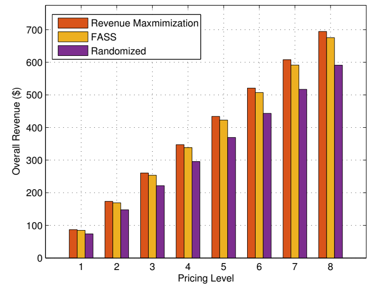

Figure 3: Overall Revenue under Different Algorithms.

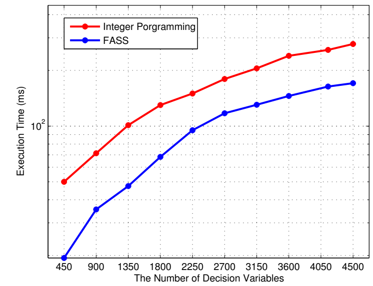

Figure 4: Algorithm Execution Time at Different Scales.

optimum as long as any optimum exists, providing the legality basis of linear relaxation on the integer constraints (7). The integer constraints get relaxed as (11).

$$x_{i,j}^n \in \mathbb{R}^+,\ \forall n \in \mathcal{N},\ i \in S_n,\ j \in C_i \tag{11}$$

Equivalent Convex Objective. The optimal service selection scheme of the problem (10) can be attained by solving the following lexicographical problem as (12). The common goal of this problem is to maximize the lowest payment across multiple service request, which is specifically the minimum element in $\boldsymbol{\varpi}$. Thus, it shows that the optimal decision variable $\boldsymbol{x}^*$ derived from the problem (12) is equivalent to the optimal solution of the problem (10).

$$\underset{x_{i,j}^n \in \boldsymbol{\Theta}}{lexmax} \quad \boldsymbol{\varpi} = (\pi_{i,j}^n \,|\, n \in \mathcal{N}, i \in S_n, j \in C_i) \tag{12}$$

In order to eventually supply a linear objective function, a tailored separable convex objective function $\xi(\boldsymbol{\varpi})$ is defined as (13), served as an intermediate transformation of objective function. The k^{th} element of $\boldsymbol{\varpi}$ is labeled by ϖ_k.

$$\xi(\boldsymbol{\varpi}) = \sum_{k=1}^{|\boldsymbol{\varpi}|} |\boldsymbol{\varpi}|^{-\varpi_k} = \sum_{k=1}^{K} K^{-\varpi_k} \tag{13}$$

Lemma 2: $\xi(\cdot)$ reverses the original partial order of lexicographically no greater than ($\succeq$), which is mathematically represented as $\boldsymbol{\varpi}(\boldsymbol{x}^*) \succeq \boldsymbol{\varpi}(\boldsymbol{x}) \Leftrightarrow \xi(\boldsymbol{\varpi}(\boldsymbol{x}^*)) \leq \boldsymbol{\varpi}(\boldsymbol{x})$.

The proof of Lemma 2 can be found in a longer version of this paper [13]. It follows that

$$\underset{x_{i,j}^n \in \boldsymbol{\Theta}}{lexmax}\, \boldsymbol{\varpi} \Longleftrightarrow \underset{x_{i,j}^n \in \boldsymbol{\Theta}}{min}\, \xi(\boldsymbol{\varpi}) = \sum_{n \in \mathcal{N}} \sum_{i \in S_n} \sum_{j \in C_i} K^{-\pi_{i,j}^n} \tag{14}$$

where $\xi(\boldsymbol{\varpi})$ is a summation of the term $K^{\pi_{i,j}^n}$ which is a convex function in terms of the single variable $x_{i,j}^n$. Therefore, solving the problem (10) is equivalent to solving the following problem (13) with constraints (5), (6) and (11).

$$\underset{x_{i,j}^n \in \boldsymbol{\Theta}}{min} \sum_{n \in \mathcal{N}} \sum_{i \in S_n} \sum_{j \in C_i} K^{-[a_n + b_n \times (1 - \frac{Q_{i,j}}{Q_n^{(ref)}} \cdot x_{i,j}^n)]} \tag{15}$$

LP Transformation. Given the properties of separable convex objective and totally unimodular linear constraints holding as true, we introduce the λ-technique [14] for optimality-equivalent Linear Programming (LP) transformation from the problem (19) in order to obtain the target service selection scheme with high efficiency. In our problem, each convex function $K^{-\pi_{i,j}^n}$ is transformed with λ-technique into another form $\psi_{i,j}^n(x_{i,j}^n)$, formulated as follows.

$$\psi_{i,j}^n(x_{i,j}^n) = \sum_{p \in \{0,1\}} K^{-[a_n + b_n \times (1 - \frac{Q_{i,j}}{Q_n^{(ref)}} \cdot p)]} \lambda_{i,j}^{n,p} \tag{16}$$

The domain of decision variable $x_{i,j}^n$ is migrated from a discrete space $\{0,1\}$ to a continuous positive real space by the means of traversing each possible value $x_{i,j}^n \in \{0,1\}$, and newly introducing a couple of weighted variables $\lambda_{i,j}^{n,0}, \lambda_{i,j}^{n,1} \in \mathbb{R}^+$ subject to (17) and (18).

$$\lambda_{i,j}^{n,0} + \lambda_{i,j}^{n,1} = 1, \quad \forall n \in \mathcal{N},\ i \in S_n,\ j \in C_i \tag{17}$$

$$x_{i,j}^n = \lambda_{i,j}^{n,1}, \quad \forall n \in \mathcal{N},\ i \in S_n,\ j \in C_i \tag{18}$$

Jointly considering the linear relaxation on the integer constraints, the linear programming problem is eventually obtained as (19).

$$\underset{\boldsymbol{x},\boldsymbol{\lambda}}{min} \sum_{n \in \mathcal{N}} \sum_{i \in S_n} \sum_{j \in C_i} K_0 \cdot \lambda_{i,j}^{n,0} + K_1 \cdot \lambda_{i,j}^{n,1} \tag{19}$$

subject to (5)-(6), (17)-(18) and

$$x_{i,j}^n,\ \lambda_{i,j}^{n,0},\ \lambda_{i,j}^{n,1} \in \mathbb{R}^+ \quad \forall n \in \mathcal{N},\ i \in S_n,\ j \in C_i$$

$$K_0 = K^{-(a_n + b_n)},\ K_1 = K^{-[a_n + b_n \times (1 - \frac{Q_{i,j}}{Q_n^{(ref)}})]}$$

Theorem 1: An optimal service selection scheme derived from the problem (19) coincides with an optimal scheme derived from the problem (4).

The proof of Theorem 1 can be found in a longer version of this paper [13]. From now on, the optimal service selection scheme across multiple service requests maximizing the lowest payment can be computed with efficient LP algorithms (e.g., Simplex Algorithm) and solvers (e.g., CPLEX [15]).

IV. Experiments and Results

A. Experimental Setup

We respectively set the amount of service providers and service requests as $\mathcal{M} = 9$ and $\mathcal{N} = 10$. The candidate service associated with the QoS value originates from WS-Dream dataset [16], which measures response time for 5,825 types of real-world web services from disparate locations. Nine amongst 5,825 types of web services are randomly chosen as service providers. At the user side, 10 users simultaneously make service requests, each of which corresponds to a broker responsible for regulating the service selection process. The privilege for service selection is restricted to specified service providers. Our simulator is implemented in C++, invoking IBM CPLEX [15] to solve our LP problems.

B. Experimental Results

To extensively investigate the optimality and fairness of our proposed algorithm, we tune the providers' pricing policy (i.e., $a_n + b_n$) to evaluate the payment deviation across multiple requests and overall revenue of service providers, depicted in Fig. 2 and Fig. 3. The pricing policy is set as 8 levels from 1 to 8, where a higher pricing level indicates that the candidate service is more highly priced.

Our proposed algorithm, referred to as *FASS*, is compared with two baselines - *Revenue Maximization* and *Randomized*. The *Revenue Maximization* algorithm refers to the algorithm whose objective is to maximize the overall revenue including all of users' payments, ignoring the variation of service assignment amongst users, while the *Randomized* algorithm randomly selects a service for execution. The *Randomized* algorithm are executed over 1,000 runs.

On the one hand, smaller payment deviation amongst individuals guarantees the fairness of concurrent service selection. Thanks to the notion of max-min fairness, our *FASS* algorithm is at the minimum payment deviation. The *Randomized* algorithm takes the second place, whereas the *Revenue Maximization* algorithm performs with the maximum payment deviation, much less for fairness guarantee. On the other hand, service providers which attain higher revenue due to offering services gain higher profits. *Revenue Maximization* algorithm optimizes the overall revenue from all service requests, served as the *optimal* baseline in our comparison study. The *Randomized* algorithm acquires the least revenue because of its blind selecting behavior. Our *FASS* algorithm does not top the list, nevertheless, there simply exist tiny gaps away from the baseline of *Revenue Maximization*. Notwithstanding a little sacrifice of revenue gains, our *FASS* algorithm achieves the fairness guarantee across multiple service requests.

Furthermore, we evaluate the practicality of our proposed algorithm by measuring the execution time of various algorithms under different problem scales. Since it is sharply time consuming to solve lexicographical problem (4), the running times of integer programming (i.e. $x_{i,j}^n \in \{0,1\}$) and our FASS algorithm is elected to conduct comparative analysis. The running times of both algorithms are demonstrated in Fig. 4, with the number of decision variables from 450 to 4,500. Each data point representing the execution time is averaged over 20 runs. Under the growth of problem scale, the execution time of both algorithms are kept as nearly linear increase. Compared with integer programming, our FASS algorithm performs much faster over 153% to 258%. The procedure of service selection for FASS can be accomplished below tens or hundreds of milliseconds. It follows that our FASS algorithm is efficient in practice.

V. Conclusions and Future Work

Fairness is an important issue in service selection when multiple users share multiple candidate services in a service ecosystem. We study the QoS-aware service selection problem with constraints from a globally fairness viewpoint. With the objective of achieving max-min fairness across the entire system, we formulate the service selection as a lexicographical maximization problem. An efficient algorithm is designed to solve such problem with acceptably low overhead by introducing λ-technique and linear relaxation.

There are several avenues for future work. Dynamic service composition scheme might be designed based on the basic idea proposed in this paper. Experimental results obtained from real-life environments should provide us with more insights of the user/system behaviors and algorithm optimization. Pricing schemes and gaming among service providers/users in real-world systems are interesting problems for researchers in this community to study.

Acknowledgment

This work is supported by the National Key Research and Development Program of China (Nos. 2018YFB1003804 and 2016YFC0303707), the National Natural Science Foundation of China (No.61772479), the Fundamental Research Funds for the Central Universities (No. 2462018YJRC040), the Natural Science Foundation of Shandong Province of China for Major Basic Research Projects (No. ZR2017ZB0419), and the TaiShan Industrial Experts Program of Shandong Province of China (No. tscy20150305).

References

[1] H. Wu, S. Deng, W. Li, M. Fu, J. Yin, and A. Y. Zomaya, "Service selection for composition in mobile edge computing systems," in *IEEE International Conference on Web Services*, 2018, pp. 355–358.

[2] M. S. Saleem, C. Ding, X. Liu, and C. H. Chi, "Personalized decision-strategy based web service selection using a learning-to-rank algorithm," *IEEE Transactions on Services Computing*, vol. 8, no. 5, pp. 727–739, 2017.

[3] K. Bessai, S. Youcef, A. Oulamara, and C. Godart, "Bi-criteria strategies for business processes scheduling in cloud environments with fairness metrics," in *IEEE Seventh International Conference on Research Challenges in Information Science*, 2013, pp. 1–10.

[4] G. Chen, L. Xiao, J. Li, and L. Zhou, "Dynamic base station selection strategies based on location fingerprinting," in *Antennas and Propagation*, 2015, pp. 474–477.

[5] V. K. Adhikari, Y. Guo, F. Hao, and M. Varvello, "Unreeling netflix: Understanding and improving multi-cdn movie delivery," in *IEEE Conference on Computer Communications*, 2012, pp. 1620–1628.

[6] G. Castelli, M. Mamei, A. Rosi, and F. Zambonelli, "Engineering pervasive service ecosystems:the sapere approach," *Acm Transactions on Autonomous and Adaptive Systems*, vol. 10, no. 1, pp. 1–27, 2015.

[7] J. Yan, R. Kowalczyk, J. Lin, M. B. Chhetri, S. K. Goh, and J. Zhang, "Autonomous service level agreement negotiation for service composition provision," *Future Generation Computer Systems*, vol. 23, no. 6, pp. 748–759, 2007.

[8] "Amazon web services," https://aws. amazon.com/.

[9] K. Kofler, I. U. Haq, and E. Schikuta, "User-centric, heuristic optimization of service composition in clouds," in *International Euro-Par Conference on Parallel Processing*, 2010, pp. 405–417.

[10] D. E. Irwin, L. E. Grit, and J. S. Chase, "Balancing risk and reward in a market-based task service," in *IEEE International Symposium on High PERFORMANCE Distributed Computing*, 2004, pp. 160–169.

[11] A. V. Zykina, "A lexicographic optimization algorithm," *Automation and Remote Control*, vol. 65, no. 3, pp. 363–368, Mar 2004.

[12] B. H. Korte and J. Vygen, "Combinatorial optimization: Theory and algorithms," *Algorithms and Combinatorics*, vol. 21, no. 5, 2006.

[13] S. Li, J. Huang, B. Cheng, L. Cui, and Y. Shi, "Fass: A fairness-aware approach for concurrent service selection with constraints," https://arxiv.org/pdf/1905.03857.pdf, 2018.

[14] R. R. Meyer, "A class of nonlinear integer programs solvable by a single linear program," *Siam Journal on Control and Optimization*, vol. 15, no. 6, pp. 935–946, 1976.

[15] "Cplex optimizer: High-performance mathematical programming solver," https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/.

[16] Z. Zheng, Y. Zhang, and M. R. Lyu, "Investigating qos of real-world web services," *IEEE Transactions on Services Computing*, vol. 7, no. 1, pp. 32–39, 2014.