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

Songyuan Li[1], Jiwei Huang[2], Bo Cheng[1], Lizhen Cui[3], Yuliang Shi[3,4]

1 Beijing University of Posts and Telecommunications

2 China University of Petroleum – Beijing

3 Shandong University

4 Dareway Software Co. Ltd.

lisy@bupt.edu.cn

July 8 -13, 2019, Milan, Italy

Outline

- **Background**
- Concurrent Service Selection Model
- Lexicographical Problem Formulation Achieving MMF
- Efficient Solution to an MMF-Aware Service Selection Plan
- Evaluation
- Conclusion and Future Work

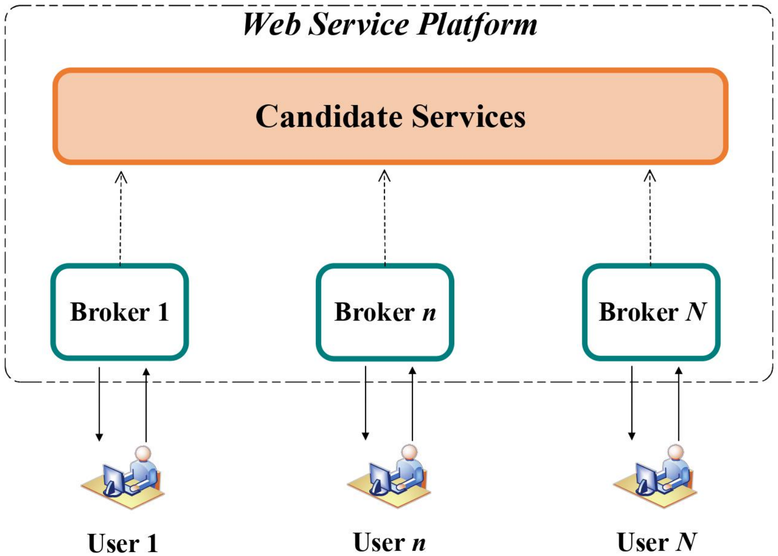
Background

❖ QoS-Aware Service Selection

- Candidate services with equivalent functionality but varying in **non-functional** properties (e.g, **QoS**)
- Gain the target service from various candidates with the **optimal QoS**
- An optimization problem
  - ✓ Integer Programming
  - ✓ NP-hard

Background

- Concurrent Service Selection
  - Service selection with multiple service requests addressed by users simultaneously
  - Concurrent service running for execution



Background

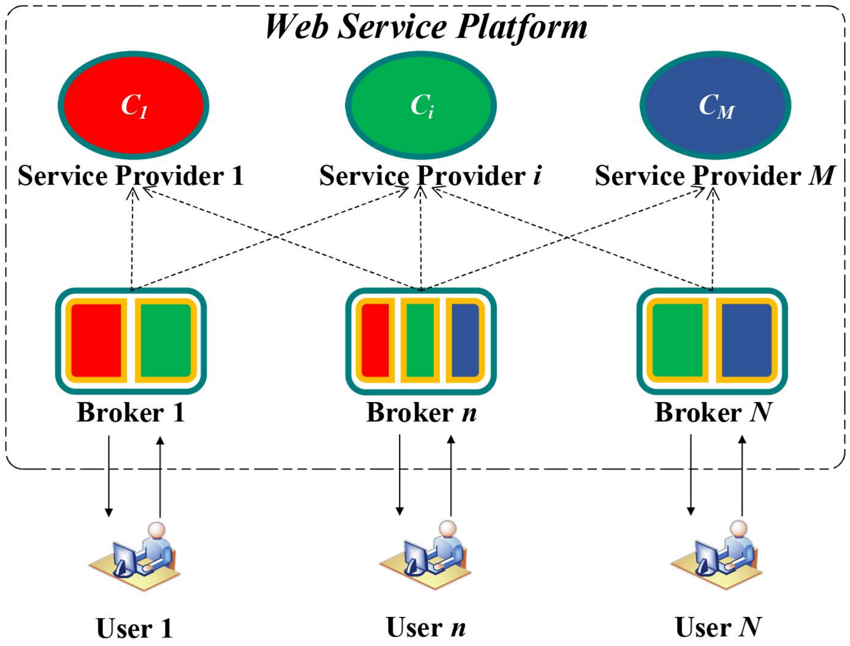
❖ Service Selection with Constraints

- Service requests restricted to a specific set of candidate services
- Wide application in several aspects
  - ✓ Location-aware BS mobile communications
  - ✓ Content-aware Internet Service Providers
  - ✓ Service contracts between users and providers
  - ✓ …

Background

❖Service Selection with Constraints

- Service requests restricted to a specific set of candidate services



Background

❖ Fairness Issues

- Multiple requests **share** the **limited** amount of candidate services
- Multiple requests **compete** each other for obtaining a service with higher QoS

It necessitates a **fairness**-aware selection mechanism when pooling a shared set of candidate services.

Background

❖ Fairness Issues *(Service Ecosystem's View)*

- **User Side:** Acquire the target service with **high** and acceptably **fair** QoS
- **Provider Side:** Fairness guarantee helps providers hold all the existing users and attract more users, earning **more revenues** for developing services with higher QoS
- **The loop of sustainable SOA development** is built up

Background

❖Our Contributions

- Basic model formulation of concurrent service selection with constraints
- Lexicographical problem formulation for max-min fairness (MMF) objective
- Iterative LP-based optimization framework
- Empirical Experiments Based on Real-World Data

Outline

- Background
- **Concurrent Service Selection Model**
- Lexicographical Problem Formulation Achieving MMF
- Efficient Solution to an MMF-Aware Service Selection Plan
- Evaluation
- Conclusion and Future Work

Basic Modeling Setting

- Multiple Service Requests:

$$\mathcal{N} = \{1, 2, ..., N\}$$

- *M* Third-Party Service Providers:

$$\mathcal{M} = \{C_1, C_2, ... C_M\}$$

- Service Selection Constraint for Service Request *n*:

S_n: The set of service providers for service request n authorized to select service from

$\mathcal{N}_i$: The set of service requests which are authorized service accessed by the service provider i

Basic Modeling Setting

- QoS value of the j^{th} service in the candidate set C_i:

$$Q_{i,j}$$ Response Time

- Execution time of service request n :

$$\tau_n = \sum_{i \in S_n} \sum_{j \in C_i} x_{i,j}^n Q_{i,j}$$

$x_{i,j}^n$: binary decision variable

Service Payment Model

- Reference QoS value of service request n:

$$Q_n^{(ref)}$$

- Basic payment for launching service

$$a_n$$

- Maximum extra bonus for obtaining a better service outperforming the reference QoS

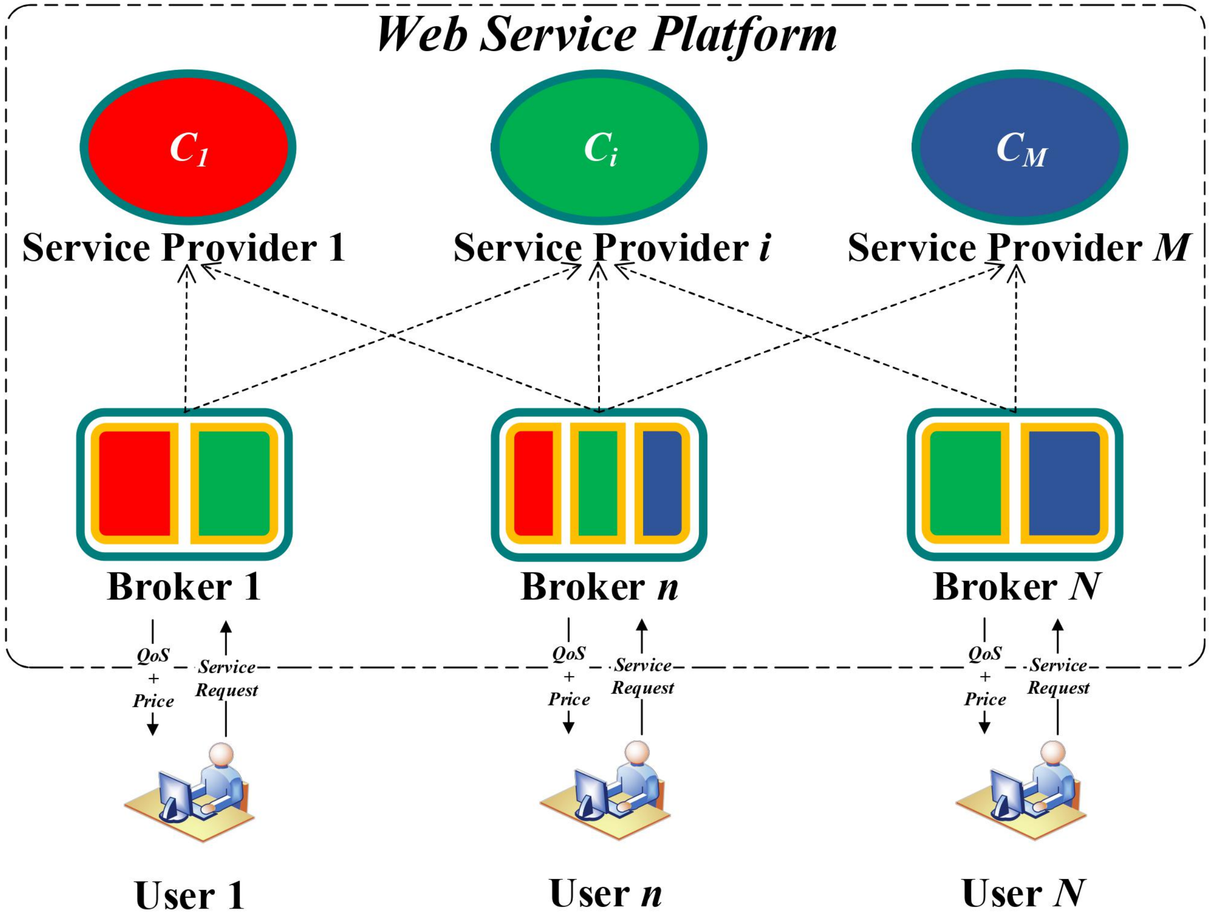
$$b_n$$

Service Payment Model

- Reference QoS value of service request n: $Q_n^{(ref)}$
- Basic payment for launching service: a_n
- Maximum extra bonus for obtaining a better service outperforming the reference QoS: b_n
- User n's overall payment:

$$\pi_n = a_n + b_n \cdot (1 - \sum_{i \in S_n} \sum_{j \in C_i} \frac{Q_{i,j}}{Q_n^{(ref)}} \cdot x_{i,j}^n)$$

Concurrent Service Selection Model



Outline

- Background
- Concurrent Service Selection Model
- **Lexicographical Problem Formulation Achieving MMF**
- Efficient Solution to an MMF-Aware Service Selection Plan
- Evaluation
- Conclusion and Future Work

Max-Min Fairness (MMF)

❖ Def. Max-Min Fairness

- A service selection scheme satisfies max-min fairness (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 ith lowest payment, note that $i \in \mathcal{N}$

MMF Optimization

- Maximize the **lowest** payment amongst the multiple requests
- Then, optimize the **second lowest** **without impacting the previous one**, and **so forth**.
- Until **all** of the service requests have been optimized, the procedure of service selection will be **terminated** with an MMF service selection scheme obtained

Lexicographical Problem Formulation

❖Lexicographical techniques grants the highest optimization priority to the most important objective.

$$\underset{x_{i,j}^{n}\in\Theta}{lexmax}\ \boxed{\pi=(\pi_1,\pi_2,......,\pi_N)}$$ Payment Vector

subject to,

$$\sum_{i\in S_n}\sum_{j\in C_i} x_{i,j}^{n}=1,\ \forall n\in\mathcal{N}$$

User Constraints

$$\sum_{n\in\mathcal{N}_i} x_{i,j}^{n}\leq 1,\ \forall i\in\mathcal{M},\ \forall j\in C_i$$

Provider Constraints

$$x_{i,j}^{n}\in\{0,1\},\ \forall n\in\mathcal{N},\ i\in S_n,\ j\in C_i$$

Discrete Domain

Lexicographical Problem Formulation

$$\underset{x_{i,j}^{n} \in \Theta}{lexmax} \quad \pi = (\pi_1, \pi_2, \ldots\ldots, \pi_N)$$

- The first smallest element of π^* (i.e., the lowest payment across multiple requests) should be the maximum amongst all feasible solutions π.
- With all π maximizing the same lowest payment, the second lowest payment in π^* is applied for maximization.

Outline

- Background
- Concurrent Service Selection Model
- Lexicographical Problem Formulation Achieving MMF
- **Efficient Solution to an MMF-Aware Service Selection Plan**
- Evaluation
- Conclusion and Future Work

Iterative MMF Optimization Framework

- The service assignments for all N service requests are iteratively accomplished one after another according to the **non-decreasingly order of service payments**
- The service request n^* with the i^{th} lowest payment is prioritized for service selection and payment optimization, treated as a subproblem implemented by a Linear Programming (LP) problem. **(i orderd by $1,\ldots,N$)**
- Freeze the service assignment of optimized request at the end of each round.

5: Fix the variable subset $\boldsymbol{x}_{n^*}$;

6: Set $x_{i,j}^{n} \leftarrow 0$, in the case of arbitrary $n \neq n^*$;

7: $\Theta \leftarrow \Theta \backslash \{x_{i,j}^{n} \,|\, n = n^*\}$ $\triangleright$ Lower the dimension of solution space Θ by one

8: $\Theta \leftarrow \Theta \cap \{x_{i,j}^{n} \,|\, x_{i,j}^{n^*} = 1, i \in S_{n^*}, j \in C_i\}$; $\triangleright$ Reduce the solution space Θ

Iterative MMF Optimization Framework

❖Pseudo Code

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

Input: Basic Payment $\mathcal{A}$, Extra Bonus $\mathcal{B}$, and QoS Baseline $\mathcal{Q}^{(ref)}$;

Output: Service Assignment $x_{i,j}^{n}, \forall n \in \mathcal{N}, i \in S_n, j \in C_i$;

```
Initialize Ñ ← N;
while Ñ ≠ ∅ do
    x ← LP(A, B, Q^(ref), Q, Θ);   ▷ Solve the LP problem (21) to obtain the scheduling plan x
    x_{n*} ← argmin_{n∈N} π_n ;   ▷ Obtain workflow n*'s optimal scheduling plan
    Fix the variable subset x_{n*} ;
    Set x^n_{i,j} ← 0, in the case of arbitrary n ≠ n* ;
    Θ ← Θ\{x^n_{i,j} | n = n*}   ▷ Lower the dimension of solution space Θ by one
    Θ ← Θ ∩ {x^n_{i,j} | x^{n*}_{i,j} = 1, i ∈ S_{n*}, j ∈ C_i};   ▷ Reduce the solution space Θ
    Ñ ← Ñ\{n*};   ▷ Hold all but n* and prepare for the next round
end while
return x^n_{i,j}, ∀n ∈ N, i ∈ S_n, j ∈ C_i;
```

LP Transformation

- Equivalent Convex Objective
- Integral Optimum Guarantee
  - Totally unimodular (TU) coefficient matrix
- Integral Relaxation
  - λ - technique

Equivalent Convex Objective

- Convex Objective Function

$$\sum_{n\in\mathcal{N}}\sum_{i\in S_n}\sum_{j\in C_i} K^{-\pi_{i,j}^n}$$

- $\underset{x_{i,j}^n\in\Theta}{lexmax}\ \boldsymbol{\pi}=(\pi_1,\pi_2,\ldots\ldots,\pi_N) \Longleftrightarrow \underset{x_{i,j}^n\in\Theta}{min}\sum_{n\in\mathcal{N}}\sum_{i\in S_n}\sum_{j\in C_i} K^{-\pi_{i,j}^n}$

  is proved in Lemma 2.

K: length of vector payment vector formed by the terms
$\pi_{i,j}^n, \forall n\in\mathcal{N}, i\in S_n, j\in C_i$

Integral Optimum Guarantee

- Totally unimodular (TU) coefficient matrix
  - User and provider constraints form up a matrix $\mathbf{A}_{s\times t}$ with $s = N + \sum_{i\in\mathcal{M}} |C_i|$, $t = \sum_{n\in\mathcal{N}} \sum_{i\in S_n} |C_i|$
  - TU matrix should meet two sufficient conditions
    - $x^n_{i,j} \in \{0, 1\}$
    - $|\sum_{i_1\in R_1} a_{i_1 j} - \sum_{i_2\in R_2} a_{i_2 j}| \leq 1$ when R_1 = $\{1, 2, ..., N\}$ and R_2 = $\left\{N+1, \ldots N+\sum_{i\in M}|C_i|\right\}$, $\forall j \in \{1, 2, ..., t\}$
  - Detailed proof can be found in Lemma 1.

$\sum_{i\in S_n}\sum_{j\in C_i} x^n_{i,j} = 1, \ \forall n \in \mathcal{N}$	User Constraints
$\sum_{n\in\mathcal{N}_i} x^n_{i,j} \leq 1, \ \forall i \in \mathcal{M}, \forall j \in C_i$	Provider Constraints
$x^n_{i,j} \in \{0,1\}, \ \forall n \in \mathcal{N}, \ i \in S_n, \ j \in C_i$	Discrete Domain

北京郵電大學
BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS

Integral Relaxation

❖ Convex function $K^{-\pi_{i,j}^{n}}$ is transformed with λ - technique

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

$$= K^{-(a_n+b_n)}\lambda_{i,j}^{n,0} + K^{-[a_n+b_n\times(1-\frac{Q_{i,j}}{Q_n^{(ref)}})]}\lambda_{i,j}^{n,1}$$

where,

$$\sum_{p\in\{0,1\}}\lambda_{i,j}^{n,p} = \lambda_{i,j}^{n,0} + \lambda_{i,j}^{n,1} = 1$$

$$x_{i,j}^{n} = \sum_{p\in\{0,1\}} p\cdot\lambda_{i,j}^{n,p} = \lambda_{i,j}^{n,1}$$

Linear Programming (final)

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

subject to,

$$\sum_{i\in S_n} \sum_{j\in C_i} x_{i,j}^{n} = 1, \ \forall n \in \mathcal{N}$$

$$\sum_{n\in\mathcal{N}_i} x_{i,j}^{n} \leq 1 \ \ \forall i \in \mathcal{M}, \forall j \in C_i$$

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

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

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

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

Outline

- ❖ Background
- ❖ Concurrent Service Selection Model
- ❖ Lexicographical Problem Formulation Achieving MMF
- ❖ Efficient Solution to an MMF-Aware Service Selection Plan
- ❖ **Evaluation**
- ❖ Conclusion and Future Work

Evaluation

- WSDream Dataset
  - Measure response time for various types of real-world web services from disparate locations
  - In our setting, $\mathcal{M} = 9$ and $\mathcal{N} = 10$
- IBM CPLEX
  - An efficient LP solver released by IBM
  - Invoked in C++ environment on Ubuntu 18.04

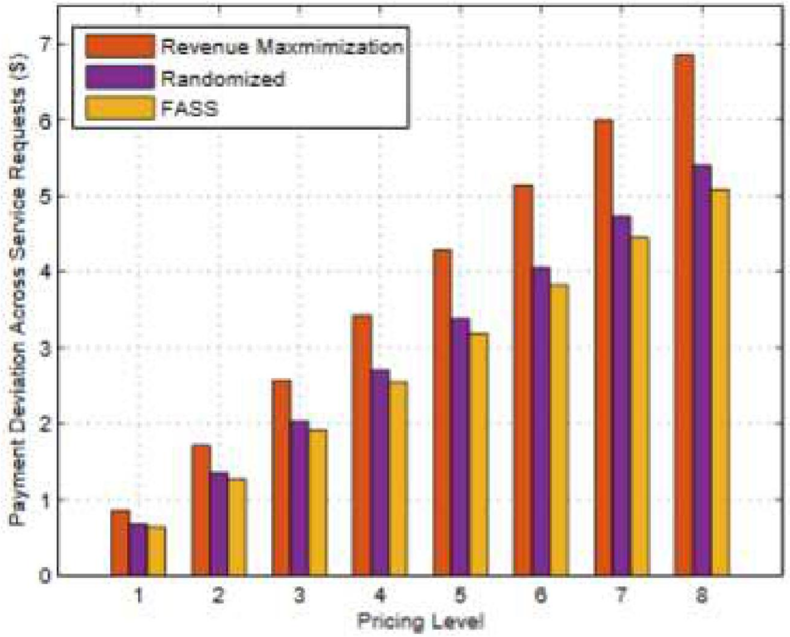
Evaluation

❖ State-of-the-Art Solutions

- Revenue Maximization Algorithm
  - Optimization objective is to maximize the overall revenue including all users' payments
  - Ignoring the deviation of charged money amongst users
- Randomized Algorithm
  - Randomly select a service from available candi-dates for execution

Evaluation

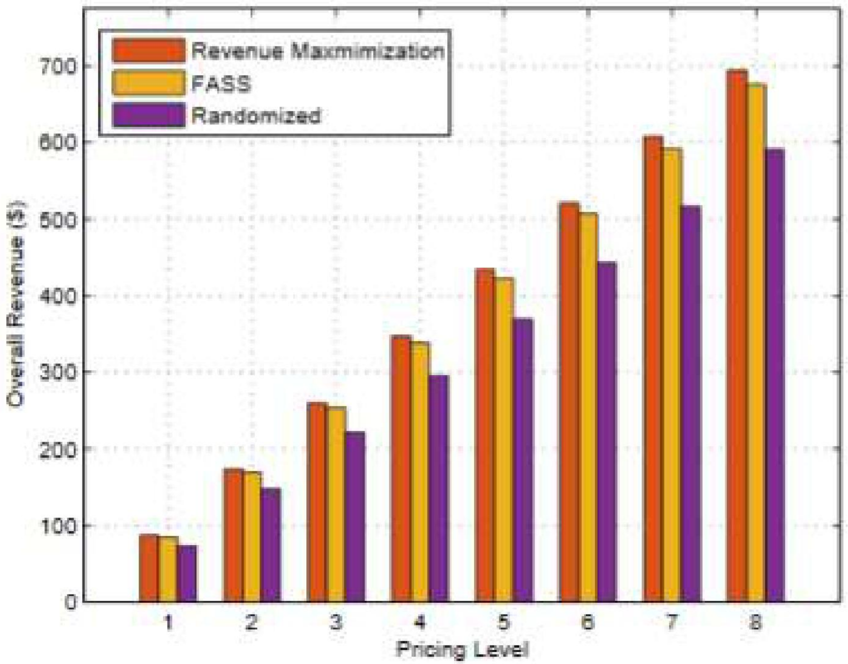
❖Payment Deviation under Different Algorithms



Payment Deviation under Different Algorithms.

Evaluation

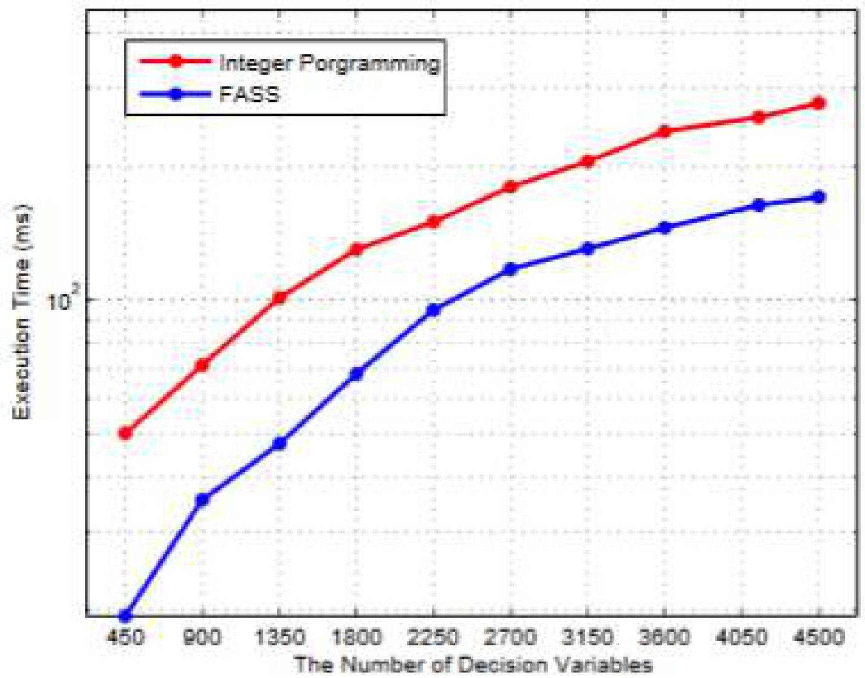
❖Overall Revenue under Different Algorithms



Overall Revenue under Different Algorithms.

Evaluation

❖Algorithm Execution Time at Different Scales



Algorithm Execution Time at Different Scales.

Outline

- Background
- Concurrent Service Selection Model
- Lexicographical Problem Formulation Achieving MMF
- Efficient Solution to an MMF-Aware Service Selection Plan
- Evaluation
- **Conclusion and Future Work**

Conclusion

- Formulating the QoS-aware service selection problem from a globally fairness viewpoint, where service selection constraints are also fully considered
- Lexicographical problem achieving MMF
- Solving problem with acceptably low overhead by introducing λ-technique and linear relaxation
- Simulation experiments based on real-world data

Future Work

- Dynamic service composition scheme
- Deployment in real-life environments
- Pricing schemes
- Gaming among service providers/users

Thank you!

Q&A

Songyuan Li
Beijing University of Posts and Telecommunications
Beijing, China
lisy@bupt.edu.cn

网络与交换技术国家重点实验室
State Key Laboratory of Networking
and Switching Technology

北京邮电大学
BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS