ウインターワークショップ

Webサービス合成時に生じる品質変化を考慮した
アルゴリズムの提案

早稲田大学理工学部CS学科
深澤研究室4年　渡辺　敦

発表内容

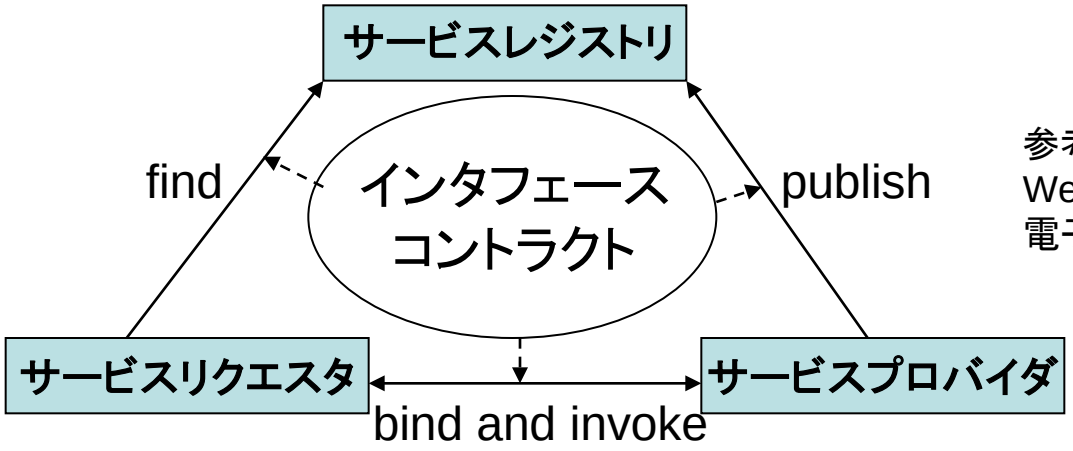
研究の概要
背景
問題点
提案手法
評価
まとめ

研究の概要

- Webサービスの合成時に生じる品質変化を考慮したアルゴリズムの提案
  - 合成サービス実行時のサービス選択時間が特定の条件の元で減少する

背景

- SOA(サービス指向アーキテクチャ)が注目され、大規模サービスを開発する際にサービス合成が利用されている
  - 合成サービス実行時のサービス選択時間を短縮する必要がある

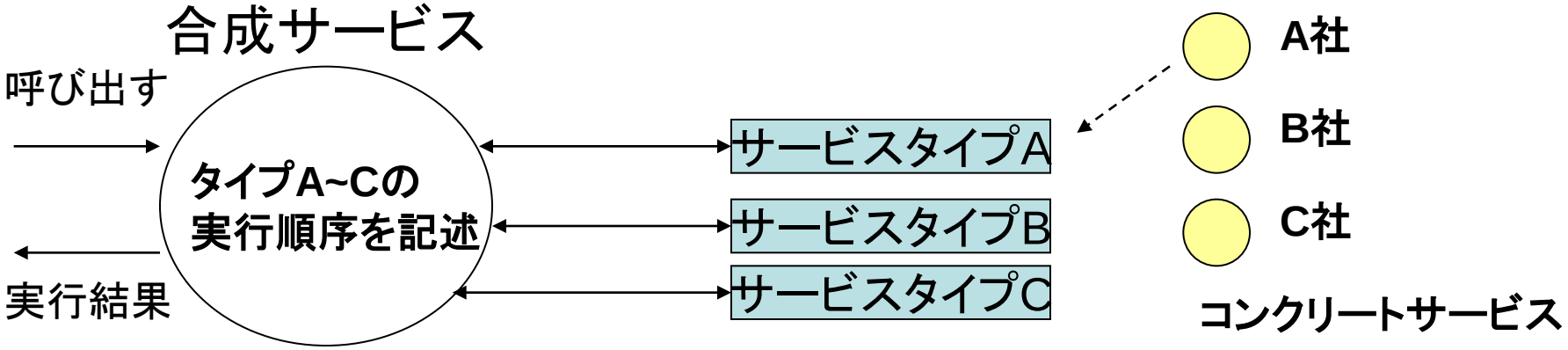


サービス指向アーキテクチャ

参考文献
Webサービスコンピューティング
電子情報通信学会発行

サービス合成

- サービス合成とは？
  - 複数のサービスを組み合わせて新しいサービス(合成サービス)を開発すること
  - 各サービスタイプに割り当てるサービスを選択



サービス選択

- 本研究では、サービスタイプが既知で、各サービスタイプに割り当てるコンクリートサービスを選択する事と定義する
  - 機能面のサービス選択はできていると仮定
  - サービスタイプが求める機能を持つコンクリートサービス群から、QoS(Qoality of Service)を考慮し、選択
  - QoSの例としては、価格や応答時間

品質変化問題

- サービス合成のQoSには、コンクリートサービスの組み合わせによって品質が向上する場合がある

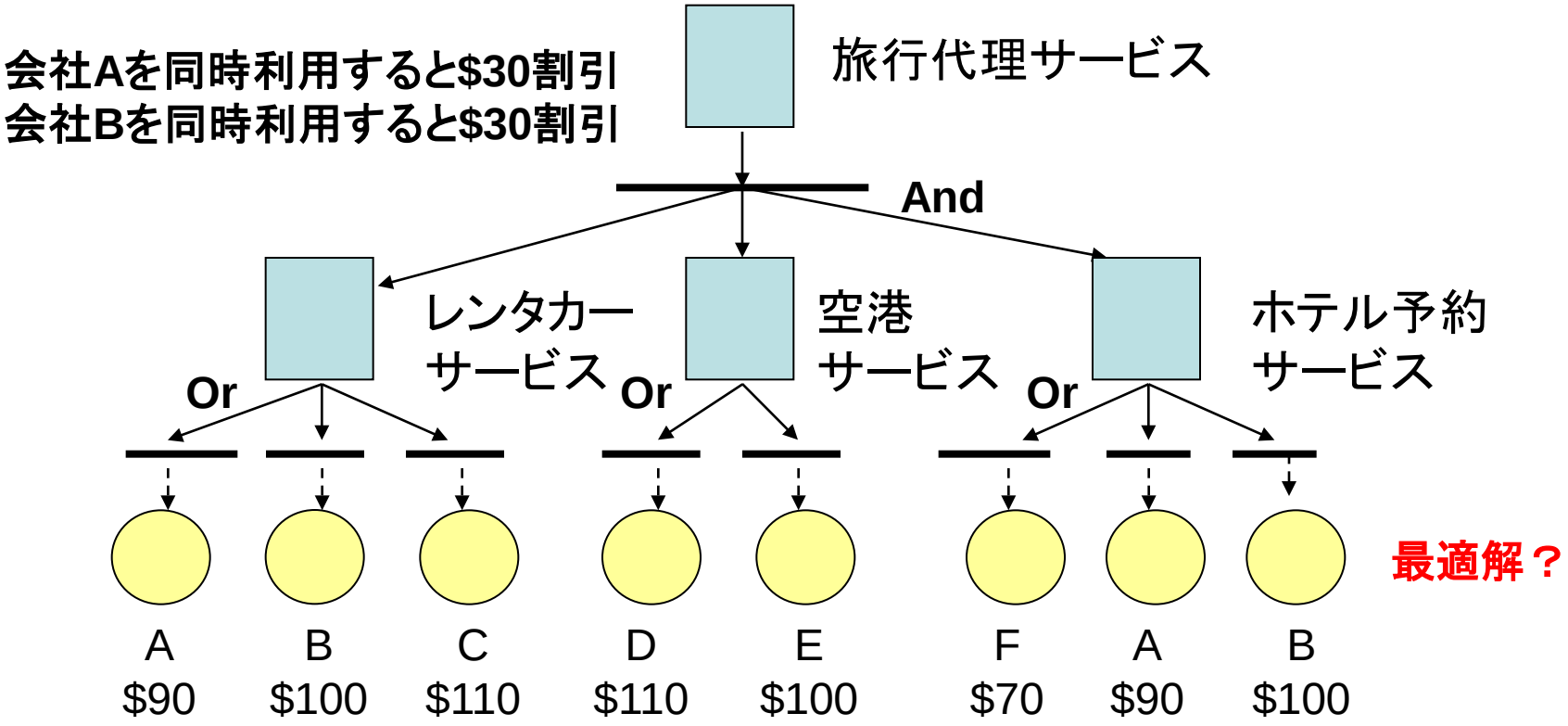
  ⇒品質変化問題と独自に定義

  品質変化の具体例

  – 特定の組み合わせで割引があるサービス
  – 具体的には、会社Aのサービスとサービスを利用すると、割引が発生する状況が考えられる

シナリオ例

- 旅行代理サービスは3つのサービスの合成サービスとする



問題点

- 個々のサービス選択時に、一意にQoSの最適解を求める事ができず、サービスの組み合わせを考慮する必要がある
  - 合成解を求める時間が指数時間になり、サービス数が増えると、合成時間が膨大になる

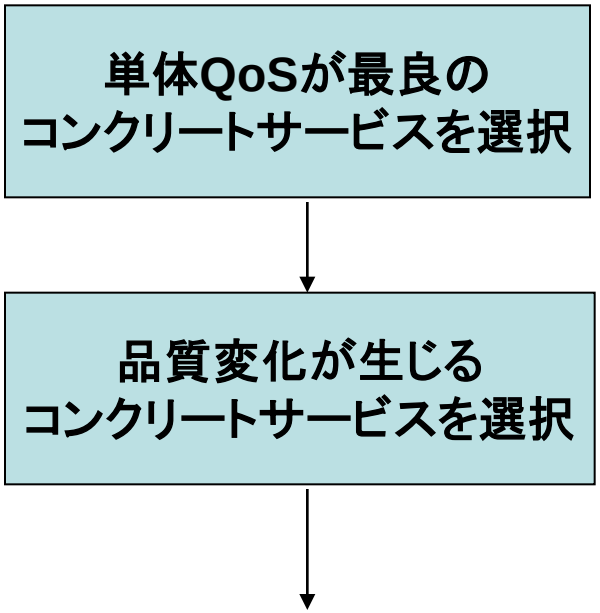
提案手法の概要

- 個々のサービスタイプに対し、最適なQoSを持つ合成サービスと、品質変化がある合成サービスだけを扱うことで、考慮する合成サービス数を削減する手法の提案

- サービスプロバイダ・タイプ数に対して多項式オーダーで選択する

- 入力
  - 品質変化リスト、サービスタイプ、各サービスタイプが求める機能を持つコンクリートサービス群
- 出力
  - QoSが最良な合成サービス

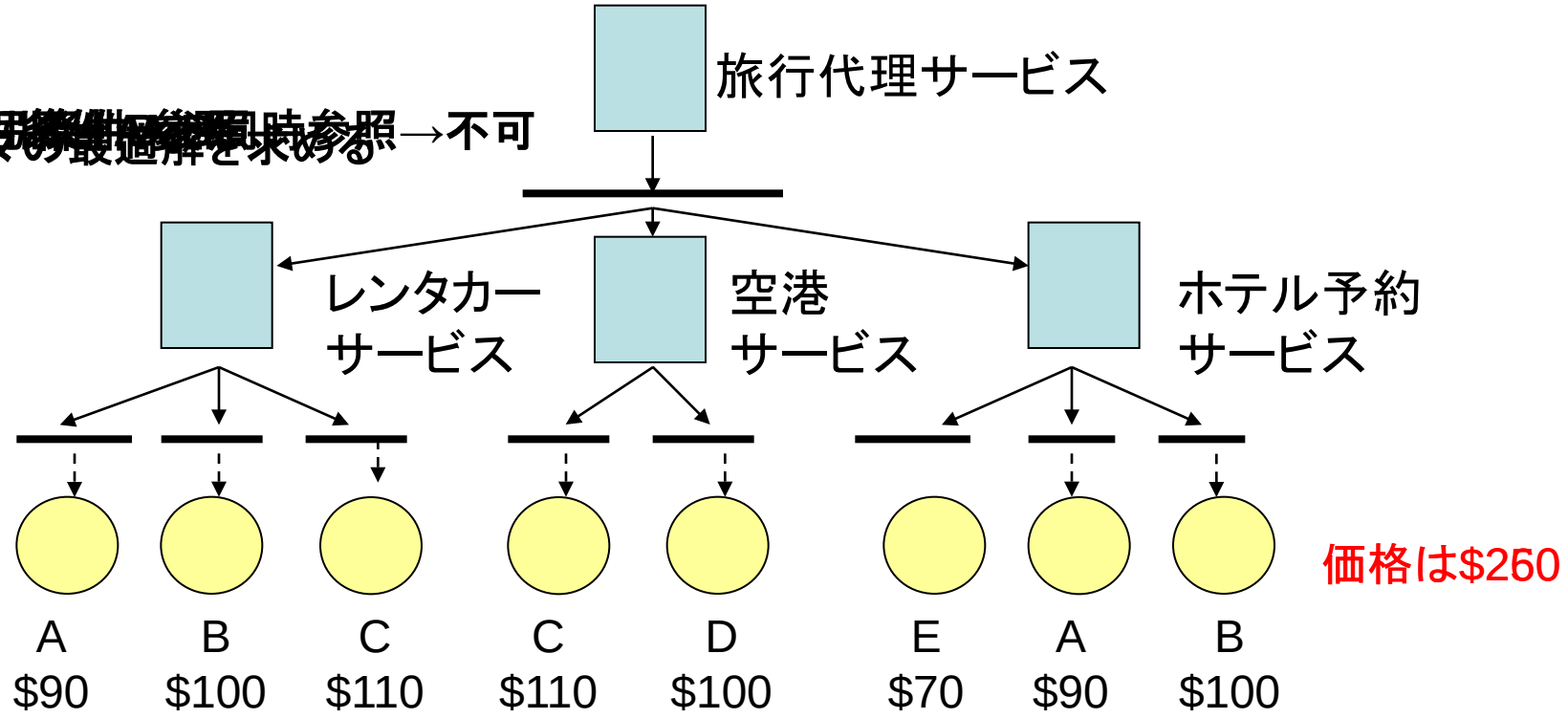
アルゴリズム



※品質変化が生じない
サービスタイプには、単体QoS最良
コンクリートサービスを選択

品質変化を考慮し、QoSが最適な合成サービスを出力

シナリオ例に適応


旅行代理サービス
レンタカーサービス
空港サービス
ホテル予約サービス
A $90
B $100
C $110
C $110
D $100
E $70
A $90
B $100
価格は$260

計算量

- 提案手法の考慮するサービス合成数は、個々の最適解と品質変化が発生する場合の2つ
  - 個々の最適解を求める計算量
    - n個のサービスタイプ、各々のコンクリートサービス数m個の場合

$$\sum_{m=1}^{n} m$$

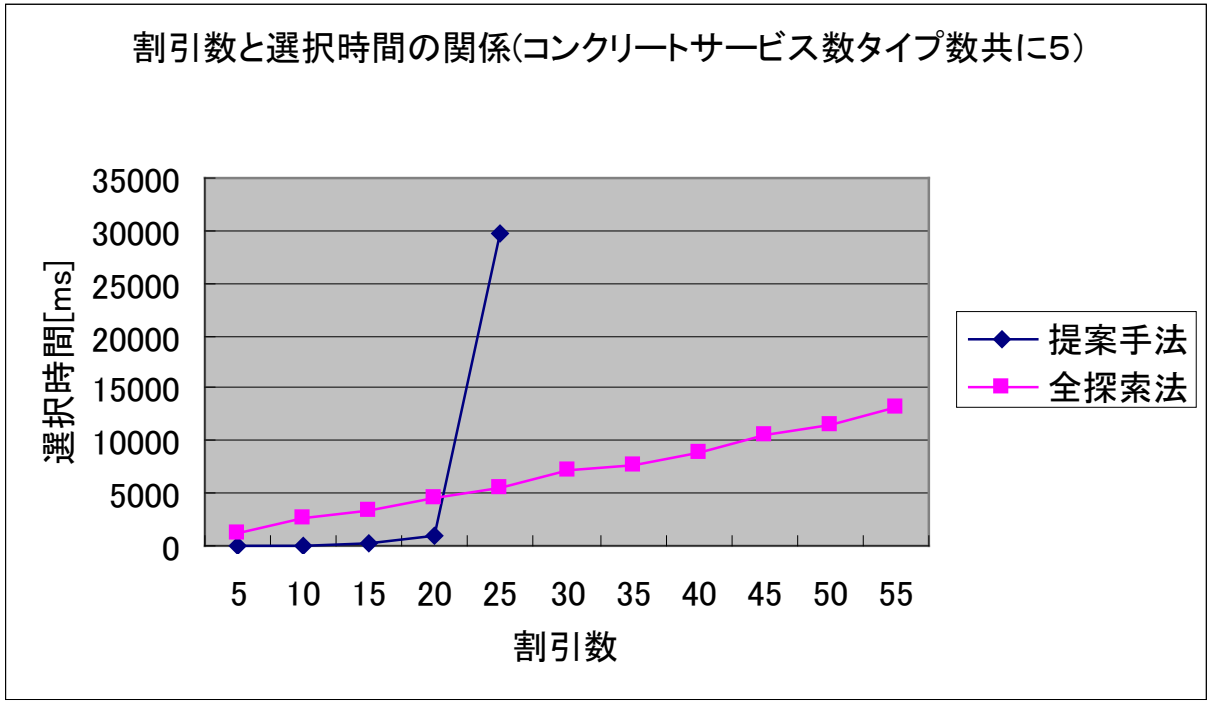
  - 品質変化がk個ある場合は、2^k+1個考慮
    - 品質変化の組み合わせ爆発が起こる

- 全探索は、$\prod_{m=1}^{n} m$ の解を求める必要がある

評価

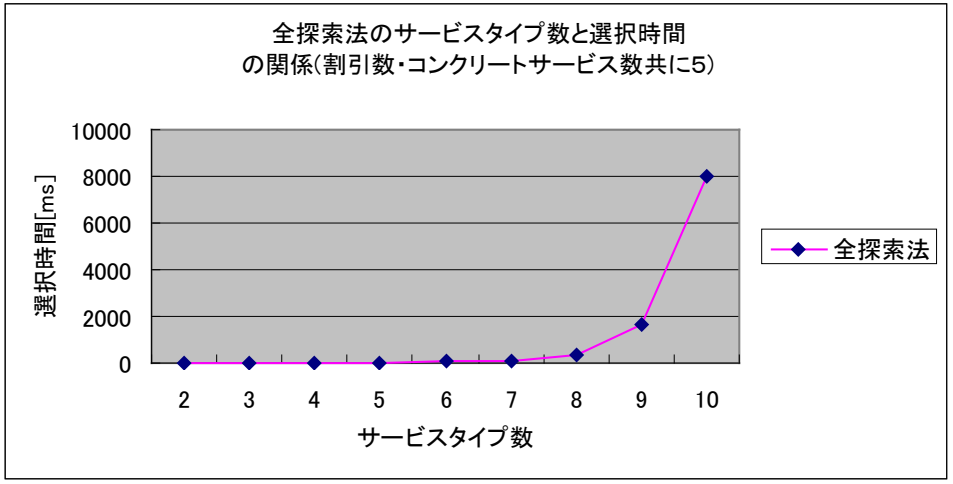
- 評価項目
  - 品質変化の数に対する選択時間の変化
  - サービスタイプ数に対する選択時間の変化
  - サービスプロバイダ数に対する選択時間の変化

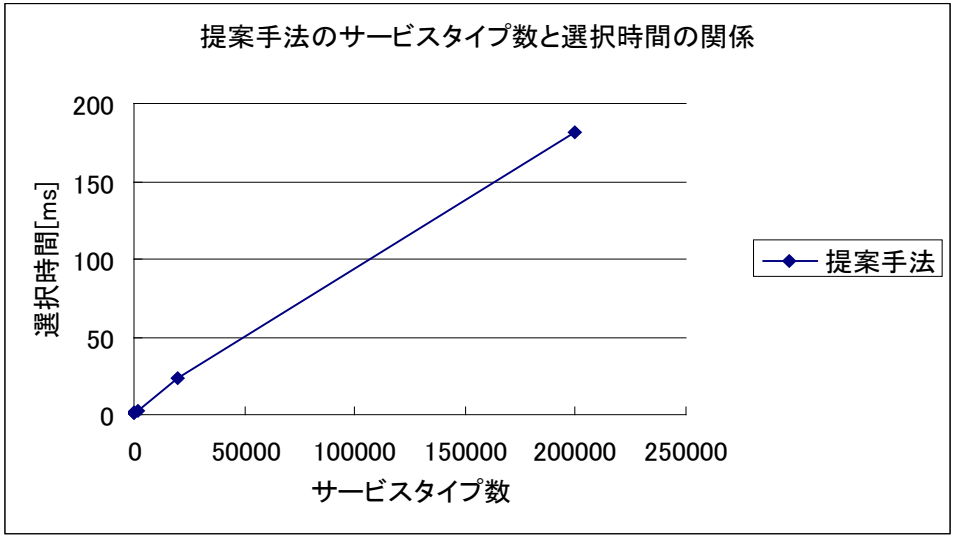
割引数と選択時間の関係



提案手法の実行時間は割引数nに対して、n^2になるため、割引数が大きくなると、全探索よりも、悪くなる

サービスタイプ数と選択時間の関係

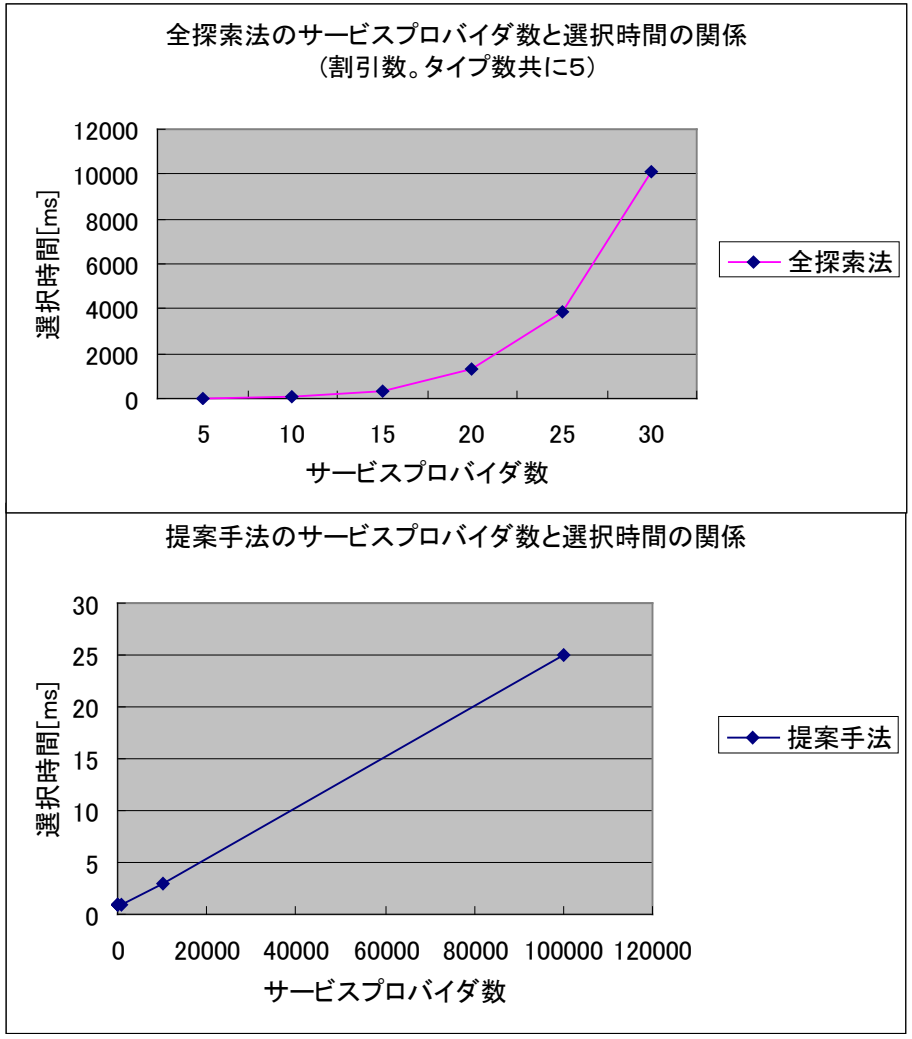




全探索は、サービスタイプ数に対して指数オーダーになる

提案手法は、サービスタイプ数に対して多項式オーダーになる

コンクリートサービス数と選択時間



全探索は、コンクリートサービス数に対して指数オーダーになる

提案手法は、コンクリートサービス数に対して多項式オーダーになる

※各サービスタイプの機能をもつコンクリートサービス数は同じ

考察

- 本手法の利点は、コンクリートサービス数とサービスタイプ数に対して多項式オーダー
- 悪い点は、品質変化に対して指数オーダー

- 品質変化数nがサービスタイプ数t・コンクリートサービス数cとする
  - 2^n < n*t^c の時に効果的な手法

研究の立ち位置

- 関連研究として、リクエスタのQoS制約を満足する合成サービスを見つける手法が多数あるが、本研究では、QoSを最大化する問題を扱った

まとめ

- 本手法により、合成サービス実行時のサービス選択時間が特定の条件の元で減少する
- 今後は、有効な枝刈り方法の研究を行う

- 参考文献
  - Webサービスコンピューティング　電子情報通信学会発行

付録

- 本手法で最適解が求まる事の証明
- 品質変化を同時に2つ以上利用すると、更に良いQoSが見つかることの証明
  - 品質変化の組み合わせを考慮する必要が生じる

ワード

擬似コード

Algorithm 1 : Searching the best optimal composite service
Input : Service[] abstractProcessComposionPlan, QoSList[] CombinationQoSList[],Service ConcreteServices,

Output: Service[] BQCS //the best QoS composite service

```
 1 Service[] IOCS ← IndividualOptimalCompositeService(
abstractProcessComposionPlan,ConcreteServices,RequestQoS);
 2 BQCS ← IOCS;
 3 int BQ ← CalculateQoS(BQCS); // The best QoS
 4 FOR i = 1 to CombinationQoSList.size()^2 DO
 5 Service[] CBQCS ← changeCS(IOCS,CombinationQoSList,i);
//change individual optimal composite service
 6   int CQ  ← CalculateQoS(CBQCS);
 7   IF CQ > BQ THEN
 8     BQ ← CQ;
 9     BQCS ← CBQCS;
10   END IF
11 END FOR
12 Return BQCS;
```