

QoS 기반 웹 서비스 선택 모형의 성능 평가

서상구*

Performance Evaluation of QoS-based Web Services Selection Models

Sangkoo Seo *

요약

공개 웹 서비스의 수요가 증가하면 동일한 기능을 제공하는 웹 서비스가 여러 개 존재하게 될 것이다. 이들은 가격, 응답 시간, 가용성 등 다양한 QoS 특성들을 가지게 되는데, 주어진 QoS 제약을 만족하며 가장 적합한 서비스를 선택하는 것은 매우 중요한 작업이다. 본 논문에서는 비즈니스 프로세스의 병행 분기와 응답시간 제약을 중점적으로 고려하여 서비스의 선택 모형을 연구하였다. 다차원 다중선택 Knapsack 모형을 기본으로, 각 프로세스 실행 경로 별로 응답시간 제약조건을 갖는 방안과 전체 서비스 타입에 대하여 하나의 응답시간 제약조건을 갖는 방안, 그리고 특정 경로에 대해서만 응답시간 제약 조건을 적용하는 방안을 제시하고, 실험을 통하여 서비스 수, 병행 분기의 수 그리고 제약 조건 값의 크기 등의 변화에 따른 각 방안의 성능을 살펴보았다. 실험에 사용된 데이터를 기준으로 볼 때, 파레토 우세를 이용하여 후보 서비스 수를 줄이는 것은 매우 효과적이며, 중소규모의 복합 웹 서비스에서는 전체 서비스 타입에 대하여 하나의 응답시간 제약을 설정한 방안이 실행 시간과 탐색 해의 최적 해 근접도 면에서 효율적인 것으로 분석되었다.

Abstract

As the number of public Web Services increases, there will be many services with the same functionality. These services, however, will vary in their QoS properties, such as price, response time and availability, and it is very important to choose a best service while satisfying given QoS constraints. This paper brings parallel branching and response time constraint of business processes into focus and investigates several service selection plans based on multidimensional multiple choice Knapsack model. Specifically, proposed in the paper are a plan with response time constraints for each execution flow, a plan with a single constraint over the whole service types and a plan with a constraint on a particular execution path of a composite Web Services. Experiments are conducted to observe the performance of each plan with varying the number of services, the number of branches

• 제1저자 : 서상구

• 접수일 : 2007. 7. 20, 심사일 : 2007. 7. 24, 심사완료일 : 2007. 8. 16.

* 광운대학교 경영대학 경영정보학과 교수

※ 본 연구는 2006년도 광운대학교 교내학술연구비 지원에 의하여 연구되었음.

and the values of response time constraint. Experimental results show that reducing the number of candidate services using Pareto Dominance is very effective and the plan with a constraint over the whole service types is efficient in time and solution quality for small to medium size problems.

▶ Keyword : Web Services, Service Selection, Parallel Branching, Optimization Model

1. 서론

정보 시스템 구축의 새로운 패러다임으로 많은 관심을 받고 있는 서비스 지향 아키텍처(SOA: Service Oriented Architecture)는 서비스 제공자의 서비스를 동적으로 탐색하여 실행시키는 것을 목표로 한다[1][2][3]. SOAP, WSDL 및 UDDI 표준을 바탕으로 하는 웹 서비스(Web Services)는 서비스 지향 아키텍처를 구현하는 가장 대표적인 응용 형태로 인정받고 있다[1][3][4]. BPEL 프로세스와 같이 하나의 서비스 내에서 실행 흐름에 따라 외부의 다른 서비스 호출을 포함하는 복합 웹 서비스의 개발은 응용 설계자가 실행 흐름과 함께 필요한 웹 서비스 타입들을 구성한 후, 각 서비스 타입에 적합한 후보 서비스를 응용 개발자 또는 응용 전문가가 결정하는 것이 일반적인 방식이다[5]. 앞으로 다양한 분야에서 점차 많은 수의 공개 웹 서비스들이 등장할 것으로 예상되는데 이럴 경우 개발자 또는 응용 전문가가 직접 웹 서비스를 탐색하여 가장 적합한 후보 서비스를 선택하는 작업은 용이하지 않다[5][6]. 따라서 웹 서비스 선택을 자동화하기 위한 많은 연구가 진행되어 왔는데, 이러한 웹 서비스의 선택문제는 두 가지 유형으로 나뉜다. 첫째는 서비스의 내용, 입력출력 데이터 등을 고려한 의미적 매칭 기법을 이용하여 주어진 기능적 요구조건에 부합되는 웹 서비스 종류를 식별하는 문제이다[7][8][9]. 두 번째 유형은 주어진 기능적 요구사항을 만족하는 웹 서비스가 다수 개 있을 경우에 가격, 응답시간 등과 같은 비 기능적 요소, 즉, QoS(Quality of Service)를 고려하여 가장 적합한 웹 서비스를 선택하는 문제이다.

본 논문에서는 기능적 요구사항을 만족하는 여러 후보 서비스들에 대하여, QoS 항목 가운데에서도 응답 시간과 웹 서비스의 병행 분기(parallel branching)를 중점적으로 고려하여 서비스를 선택하는 모형의 성능비교를 연구하였다. QoS에 기반한 웹 서비스의 선택과 관련하여 많은 연구가 진행되어 왔다[2][10][11][12]. 대표적으로, [12]에서는 복합 웹 서비스 구축을 위한 미들웨어 플랫폼인 AgFlow에서의 QoS 모델과 웹 서비스 선택 알고리즘을 소개하였다.

복합 웹 서비스에 대한 후보 서비스의 선택 방법으로 지역 최적화 방법과 전역 계획 최적화 방법을 제안하였는데, 지역 최적화 방법에서는 복합 웹 서비스에 포함된 각 서비스 타입 별로 가장 적합한 웹 서비스를 개별적이고 독립적으로 선택한다. 전역 계획 방법에서는 총 비용, 총 실행 시간 등에 대한 제약조건을 고려하여 복합 웹 서비스의 각 실행 경로에 대하여 최적의 후보 서비스를 선택하도록 하였다. [11]에서는 복합 웹 서비스의 총 지연시간에 대한 제약조건을 충족시키는 서비스 집합을 선택하기 위하여 여러 실행 경로 가운데 최적화 목표 함수 값이 가장 큰 경로를 최적 실행 경로로 결정하는 알고리즘을 제안하였다.

대부분의 앞선 연구들은 하나의 서비스 타입에 대한 서비스를 선택하기 위하여 다중 목표 제약에 따른 최적 선택을 시도하였거나, 여러 실행 경로 가운데 주어진 제약조건과 목표치를 가장 만족하는 실행 경로를 선택하는 것을 전제로 하고 있다. 그러나 실제 웹 서비스의 실행 흐름은 실행 시점의 데이터 값에 따라 동적으로 결정되기 때문에 미리 선택된 최적 경로와 다를 수 있다. 또, 병행분기가 있을 때, 각 경로 별로 최적 후보 서비스를 선택하였더라도 여러 경로에 중복된 서비스 타입에서 서로 다른 후보 서비스가 결정되었다면 그 중 하나만 선택해야 하므로 최적 해를 얻을 수 없을 뿐 아니라 전체 제약조건을 위배할 수도 있게 된다. 본 연구에서는 Knapsack 모형에 기반하여 복합 웹 서비스의 선택 문제를 모형화하되, 병행 분기에 영향을 받는 응답시간 제약을 중심으로 그 표현 방안과 성능을 실험하여 비교 분석하였다. 구체적으로, 후보 서비스의 수, 병행 분기의 수, 응답 제약 값의 크기 등이 각 방안의 해의 탐색 시간 및 최적 해와의 근접도 등에 미치는 영향을 살펴 보았다.

본 논문은 다음과 같이 구성된다. 2장에서는 복합 웹 서비스 QoS 모델 및 최적해 기본 모형을 설명한다. 3장에서는 병행분기에 대하여 응답시간 제약조건을 표현하는 선택 모형을 제시하고, 4장에서는 이들 방안에 대한 실험 결과를 설명한다. 그리고 5장에서 논문을 맺는다.

II. 복합 웹 서비스 선택 문제

2.1 복합 웹 서비스의 QoS 모형

웹 서비스의 QoS 기반 선택 문제란 각각의 서비스 타입에 해당되는 후보 서비스들 가운데 하나씩의 후보 서비스를 결정하되, 웹 서비스에 대한 총 기대이익이 가능한 최대화되며 이와 동시에 가격 총액, 전체 응답시간 등에 대한 제약 조건이 주어질 경우 이를 만족하도록 하는 문제이다. 여기서 기대이익이란 응용환경에 따라 다양하게 적용할 수 있다. 각 후보 서비스에 대한 기대 이익은 주로 여러 QoS 값을 종합적으로 고려하여 계산된다[11][12]. 즉, 가격, 응답시간, 지명도 등의 QoS 기준에 대하여 가중치를 부여하고 각 후보 서비스의 QoS 값을 동일한 범위 내의 값으로 변환한 후 이에 가중치를 곱하여 모두 더한 값을 각 후보 서비스에 대한 기대이익으로 계산할 수 있다.

BPEL 프로세스와 같은 복합 웹 서비스는 하나 이상의 단순 또는 복합 웹 서비스들이 순차, 반복, 분기 등의 실행 흐름에 따라 구성된다[4]. 순차 흐름에 대한 가격, 응답시간 등의 값은 단순 합계로 정의되고 가용성과 같은 QoS 값은 로그함수를 적용한 값의 합산으로 정의될 수 있다. 병행 분기는 각 분기가 각각 실행되기 때문에 가격의 경우 분기에 포함되는 모든 웹 서비스의 가격의 합으로 정의되고, 응답시간은 각 분기별로 포함된 서비스의 응답시간 합계 가운데 최대인 분기의 값을 취한다. 조건 분기의 경우에는 각 분기가 선택적으로 실행되기 때문에, 각 분기의 순차흐름에 대하여 계산된 QoS값에 그 분기의 실행 확률 값을 곱한 기대 값을 해당 분기의 QoS 값으로 재귀적으로 정의함으로써, 병행 분기와 동일하게 모형화할 수 있다. 반복 흐름은 최대 반복 횟수만큼 반복 블록내의 웹 서비스 흐름이 중복되어 있다고 간주함으로써 QoS값을 정의할 수 있다.

2.2 복합 웹 서비스 선택 문제의 기본 모형

복합 웹 서비스의 최적 구성 문제는 후보 서비스를 선택한다는 점에서 Knapsack 문제와 유사하다. 복합 웹 서비스에 포함된 각 서비스 타입을 다중선택 Knapsack의 각 그룹으로 보면, 서비스 타입에 속하는 후보 서비스는 그룹에 속하는 개체에 해당되며, 하나의 서비스 타입에 대하여 오직 하나의 후보 서비스가 결정되는 점에서 다중선택

Knapsack 문제와 유사하다[13]. 그리고 웹 서비스의 가격, 응답시간 등의 QoS 속성들과 복합 웹 서비스의 실제화에 대한 제약조건은 다차원 Knapsack의 제약 조건으로 간주할 수 있다. 따라서 2.1절의 QoS 모형에 대한 가정을 바탕으로, 다차원 다중선택 Knapsack 모형을 이용하여 복합 웹 서비스 최적구성 문제를 아래와 같이 수식화할 수 있다.

$$\text{Maximize } \sum_{i=1}^M \sum_{j=1}^{N_j} p_{ij} x_{ij} \dots\dots\dots (1)$$

$$\text{Subject to } \sum_{i=1}^M \sum_{j=1}^{N_j} q_{ijk} x_{ij} \leq W_k \text{ for } 1 \leq k \leq K \dots\dots\dots (2)$$

$$\sum_{j=1}^{N_j} x_{ij} = 1 \text{ for } 1 \leq i \leq M \text{ and } x_{ij} \in \{0, 1\} \dots\dots (3)$$

수식에 사용되는 기호와 의미는 아래 표 1과 같다. 목표함수 식 (1)은 복합 웹 서비스의 이익이 최대가 되도록 모든 후보 서비스들의 이익을 합산하는데, 제약 조건 식 (3)에 의하여 하나의 서비스 타입에서 하나의 후보 서비스가 결정되도록 제한한다. QoS 제약에 대한 제약 조건 식 (2)은 서비스 타입 별로 하나씩 결정되는 후보 서비스들의 각 QoS 값의 합계가 제약조건 값을 만족하도록 한정하고 있다.

표 1. 수식에 이용되는 기호와 정의
Table 1. Symbols and definitions for expressions

표기	의미
M	복합 웹 서비스에 포함된 서비스 타입 수
N _i	서비스 타입 ti에 속한 후보 서비스 수
K	QoS 제약 항목의 수
Q _k	웹 서비스의 QoS 항목 (1 ≤ k ≤ K)
W _k	QoS 항목 Q _k 의 제약 값 (1 ≤ k ≤ K)
t _i	복합 웹 서비스에 포함된 서비스 타입 (1 ≤ i ≤ M)
s _{ij}	서비스 타입 ti의 후보 서비스 (1 ≤ i ≤ M, 1 ≤ j ≤ N _i)
x _{ij}	후보 서비스 s _{ij} 에 대한 이진 결정변수
p _{ij}	후보 서비스 s _{ij} 의 이익 (기대이익)
q _{ijk}	후보 서비스 s _{ij} 의 서비스 항목 Q _k 의 값 (1 ≤ i ≤ M, 1 ≤ j ≤ N _i , 1 ≤ k ≤ K)

2.3 병행 분기와 응답시간

병행 분기에 대한 응답시간 QoS 값은 2.1절에서 설명한 바와 같이 가장 실행시간이 긴 분기를 병행 진행되는 서비스의 응답시간 QoS 합으로 대표된다고 하였다. 따라서 각 서비스 타입에 후보 서비스를 선정할 때에도 이 점을 고려

하여야 한다. 그러나 앞 절의 최적화 기본 모형에서 제약 조건식은 병행 작업에 대한 응답시간을 고려하지 않고 있다. 즉, 경로 별 서비스의 응답시간 합계가 아닌 선택된 모든 후보 서비스의 응답시간 합계가 제약조건 값과 비교되고 있다. 또한, 병행 분기의 한 경로의 선택 대상 후보 웹 서비스들의 응답시간이 다른 경로의 대상 후보들의 응답시간보다 월등히 작다면 그러한 경로의 웹 서비스를 선택할 때에는 응답시간은 중요하지 않다는 점을 고려하여야 한다.

III. 웹 서비스 선택 모형

3.1 IP-U: 병행분기를 고려하지 않는 모형

병행분기를 고려하지 않고 모든 후보 서비스들의 실행시간에 대하여 하나의 응답시간 제약조건을 적용하는 2.2절의 최적화 기본 모형을 이하 IP-U 라 부른다. IP-U를 이용하여 구해진 해의 목표 함수 값은 최적 해에 비하여 같거나 작게 될 것이다. 그러나 실제 구해진 해에 대한 전체 응답시간은 최적 해에 대하여 구해진 응답시간보다 같거나 짧기 때문에 문제의 크기와 복잡도에 따라 적절한 탐색시간과 최적 해와의 근접도가 절충된다면 나름대로 유효한 방안이 될 수 있을 것이다. 또한 다른 방법들의 탐색 성능을 비교할 수 있는 기본 모형이므로 방안의 하나로 설정하고 실험 대상에 포함하기로 하였다.

3.2 IP-M: 실행경로별 제약조건을 갖는 모형

병행 작업이 있을 경우에는 가장 실행시간이 긴 경로에 대하여 응답시간 제약조건이 적용되어야 한다. 후보 서비스들 가운데 어떤 서비스가 선택되더라도 전체 응답시간 제약조건이 만족되어야 하므로 모든 실행 가능한 경로, 즉, 복합 웹 서비스의 시작부터 종료까지 병행분기에 의하여 나뉘지는 모든 서로 다른 경로에 대하여 응답시간 제약조건을 각각 적용해야 한다. 이를 2.2절의 최적화 모형에 적용하기 위해서는 목표 함수식과 이진 변수의 제약 식은 동일하되, 제약조건 식을 가격 등과 같이 모든 서비스 타입에 대하여 적용할 수식과 각 실행 경로에 대하여 적용할 수식으로 분리하여야 한다. 이에 따라 새로 작성된 최적화 모형을 IP-M으로 부르고, IP-M은 응답시간을 포함한 모든 제약조건을 만족하면서 목표함수 값이 최대가 되도록 각 서비스 타입으로부터 하나씩의 후보 서비스를 결정한다. IP-M 모형의 최적화 수식은 지면관계상 생략한다.

3.3 IP-P: 파레토 우세를 적용한 모형

IP-P방안은 각 서비스 타입의 후보 서비스 가운데 최적 해에 포함될 기회가 없는 후보들을 제거한 후에, IP-M 모형을 적용한 것으로써, IP-M 보다 짧은 탐색시간으로 동일한 최적 해를 생성하는 것을 목표로 한다. 이때 제거 대상 후보는 다중목표 최적화의 한 가지 방법으로 이용되는 파레토 우세 판별을 통하여 결정된다[14]. 파레토 우세(Pareto Dominance)란, 하나의 해 s_i 가 다른 해 s_j 보다 적어도 한 가지 이상의 기준에서 우월하며 동시에 s_j 보다 열등한 기준이 하나도 없을 때, s_i 가 s_j 보다 우세한 것으로 정의된다. 웹 서비스의 선택에 이를 적용하면, 각각의 서비스 타입에 속한 후보 서비스들 가운데 서비스의 이득과 QoS 제약조건 측면에서 모두 자신보다 유리한 서비스가 존재하는 후보 서비스는 선택 대상에서 제외하여도 무방하다는 결과를 얻게 된다. 예를 들어, QoS 제약조건으로 응답시간이 주어지고 있다고 가정할 때, 동일한 서비스 타입에 속한 서비스 s_{i1} 와 s_{i2} 에 대하여, $[s_{i1} \text{의 이득}] \geq [s_{i2} \text{의 이득}]$ 과 $[s_{i1} \text{의 응답시간}] \leq [s_{i2} \text{의 응답시간}]$ 의 두 비교식이 동시에 만족한다면, 둘 사이에 s_{i1} 이 s_{i2} 보다 최소한 동등하거나 우세하다는 관계가 성립한다. 하나의 서비스 타입에 대하여 하나의 후보 서비스만 선택되어야 하며, s_{i2} 가 선택된 해에 대하여 s_{i1} 을 대신 선택하면 제약조건을 여전히 만족하면서 전체 이득은 같거나 증가하기 때문에 s_{i2} 는 선택 대상에서 배제해도 해의 선택에는 영향을 미치지 않는다. s_{i2} 와 같은 열세 후보를 사전에 제거한다면 선택 대상의 수가 줄어들어서 탐색시간을 매우 감소시킬 수 있을 것이다.

3.4 IP-S: 하나의 경로에 제약 적용한 모형

IP-P는 제약 조건의 수가 많을 때에는 열세 후보를 제거하는 효과가 미미하여 최적해 탐색 시간의 감소에 기여하기 어려울 수 있다. 또한 여러 개의 실행 경로가 있는 복합 웹 서비스의 경우에, IP-M 또는 IP-P에서처럼 각각의 경로에 대하여 응답시간 제약을 적용하더라도 실제로 응답시간 제약에 영향을 주는 것은 일부 경로에 국한될 경우가 많고, 그러한 경로가 사전에 예상되는 경우도 많을 것이다. 본 절에서는 이러한 측면에서, 병행분기에 의하여 여러 경로를 포함하는 복합 웹 서비스에서 하나의 경로에 대해서만 주어진 응답시간 제약조건을 적용하는 방안을 고려하였다.

여러 실행경로 가운데 응답시간 제약조건을 적용할 하나의 경로를 택하는 것에는 다양한 방법이 있을 수 있는데, 응답시간이 가장 클 것으로 예상되는 경로를 선택하여 제약

조건을 적용함으로써 최적 해에 가까운 해를 얻을 수 있을 것이다. 만약 각 후보 서비스의 예상 실행시간의 최소, 최대, 평균시간 등이 SLA(Service Level Agreement) 계약으로 등으로 주어져 있다면 서비스 타입 별로 최대 또는 평균 실행시간 값의 평균값을 구하고, 이들의 합이 가장 큰 경로를 선택할 수 있을 것이다. 각 서비스 타입들의 후보 서비스의 실행시간들이 비슷하고, 일반적으로 서비스 타입 수가 많은 경로일수록 전체 응답시간이 클 것이라고 가정한다면 최장 경로에 대하여 응답시간 제약을 적용하는 것이 무난하므로, 이하 본 논문에서는 최장경로를 선택하는 방안으로 가정한다. IP-S 모형의 수식은 목표함수 및 이진결정변수 제약은 나머지 방안들과 동일하되, 응답시간 제약조건식은 결정된 하나의 경로에 대해서만 주어진다. 단, 구해진 해에 대해서는 전체 제약조건의 만족여부를 확인해야 하며, 위배시에는 재탐색이 필요하다.

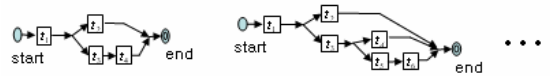
IV. 실험

4.1 실험 환경

3장에서 설명한 모형을 이용하여 해를 탐색할 때의 탐색 시간 등의 성능을 살펴보고자 한다. 각 모형을 서로 비교하기 위해서는 해를 탐색하는 프로그램을 각각 작성하는 대신 동일한 프로그램을 이용하는 것이 바람직하므로, 본 연구에서는 기존의 최적해 탐색 프로그램을 이용하기로 하였다. 실험을 위해서는 각 문제 모형으로부터 최적 탐색 프로그램의 입력 데이터를 생성해야 한다. 문제의 크기, 즉, 복잡도는 서비스 타입의 개수, 각 서비스 타입에 속하는 후보 서비스의 수, QoS 제약 항목의 수, 그리고 서비스 타입의 포함하는 분기의 구성 형태에 의하여 결정된다고 할 수 있다. 본 연구는 병행 분기에 대한 응답시간 제약조건 모형의 달리하는 방안들의 성능을 분석하는 것이 주요 목적이기 때문에 QoS 제약 항목으로서 응답시간 한 가지만 고려하였다. 이렇게 함으로써 다른 QoS 제약 항목이 해의 탐색에 미치는 영향을 줄이고, 응답시간 제약이 해의 탐색 성능에 미치는 영향을 보다 직접적으로 알아보고자 하였다. QoS 제약 항목을 추가하더라도 본 실험의 절차와 방식은 그대로 적용될 수 있다. 복합 웹 서비스의 구성 형태는 하나의 병행분기 기본형을 이용하여, 이를 반복적으로 중첩시켜서 실험에 사용될 문제를 생성하였다. 이때 반복 중첩 시에 병행분기 수 뿐 아니라 실행 경로의 길이도 조금씩 늘어나도록

하는 것이 필요하다. 또한 각 방안의 탐색 성능을 다양한 문제 크기에서 관찰할 수 있도록 기본 문제 형식을 간단히 설정하였다.

아래 그림 1의 (a)는 실험에 이용된 문제의 기본형으로서 하나의 병행분기와 t1, t2, t3, t4 네 개의 서비스 타입으로 구성된다. 그림 (b)는 그림 (a)에서 서비스 타입 t4가 새로운 병행분기로 확장된 형태로서, 두 개의 병행분기와 t1, t2, ..., t6의 서비스 타입으로 구성된다. 그 다음에는 t6가 마찬가지로 방법으로 새로운 병행분기로 확장되어, t1, t2, ..., t8의 서비스 타입으로 구성된다. 본 실험에서는 총 5개까지의 병행 분기가 중첩된 형태까지 고려하였으며, 이 경우, 서비스 타입은 t1, t2, ..., t12가 된다.



(a) 분기 수 1일 때 (b) 분기 수 2일 때
 그림 1. 실험 데이터를 위한 웹 서비스에
 Fig. 1 Example Web Services for experiment

해를 구하는 프로그램으로 OPBDP 최적해 탐색기를 이용하였다[15]. OPBDP는 Davis-Putnam 최적해 기법에 기초하고 있으며 정수 계수의 목표함수와 주어진 제약조건 부등식에 대하여 이진 결정변수의 해와 최종 목표 함수값을 구하는 프로그램이다. OPBDP는 SunOS (Starfire) 환경에서 실행되었다.

OPBDP의 입력 데이터로서 목표함수 수식과 제약조건식이 필요하고, 주어진 서비스 실행 흐름에 대하여 이들 수식을 생성하는 프로그램이 필요하다. 이를 위해서는 먼저 서비스 실행 흐름을 표현하는 방법이 요구되는데, 본 연구에서는 BPEL의 activity 태그와 그 표기법을 활용하였다 [4]. 즉, 순차적 흐름은 <seq> ... </seq>로 나타내고, 병행 분기는 <flow> ... </flow>로, 그리고 서비스 호출은 <invoke> task </invoke>로 표현하였다. BPEL의 XML 문서 형태의 서비스 구성 데이터로부터 각 모형별 목표함수와 제약 조건식을 생성하기 위하여 이벤트처리 방식의 SAX 인터페이스를 이용하여 Java 프로그램을 작성하였다. 프로그램의 인수는 BPEL 태그 파일과 서비스 타입 별 후보 수인데, BPEL 태그 파일은 데이터 기본형부터 분기가 5회 적용된 5개의 파일을 미리 준비하였다. 후보의 수는 10부터 50까지 5씩 증가하는 값이다. 프로그램의 실행 결과로 목표 함수 식과 IP-M, IP-P, IP-S, IP-U 네 가지 모형에 대한 제약조건식이 생성된다. 각 후보 서비스의 이득 값은 10 ~ 30의 임의 값으로, 응답시간 값은 5 ~ 20 사이의 임의의

값으로 생성하였다. 임의값에 대한 편차의 영향을 줄이기 위하여, 분기 수와 후보 수가 주어졌을 때 문제 인스턴스를 2개씩 생성하여 OPBDP를 각각 실행한 후, 실행 시간과 목표 함수 값의 평균값을 측정하였다.

4.2 후보 서비스 수의 변화에 대한 실험

4.2.1 후보 서비스 개수 변화에 대한 실행 시간의 비교

서비스 타입당 후보 서비스의 개수를 변화시키면서 IP-M, IP-U, IP-S, IP-P 각각의 방법에서 해를 구하는데 소요된 실행시간을 측정하였다. 즉, 병행 작업의 중첩 정도를 고정했을 때 후보 서비스 개수가 해의 탐색에 어느 정도의 부하를 가하는지 알아보고자 하였다. 실험에서 병행 중첩의 수를 2로 하였는데, 이는 중첩수가 2보다 큰 경우에는 IP-M과 IP-S 방법의 실행시간이 너무 급격히 증가하였기에 가능한 많은 측정 결과를 비교하기 위하여 중첩의 수가 크지 않도록 한 것이다. 이때 실험 대상 모형에서 서비스 타입의 수는 6개, 병행 실행 경로의 수는 3개이다. IP-M과 IP-P의 경우에, 병행경로에 대한 응답시간 제약식 3개, 서비스 타입에 대한 결정변수 제약식 6개가 생성된다. 응답시간 제약 값은 모두 동일한 값(40)으로 하여 탐색조건이 일정하도록 하였다.

측정 결과는 그림 2과 같다. 제약 조건에서 병행 작업을 고려하지 않은 IP-U와 전처리 과정을 거친 IP-P를 제외한 두 가지 방법은 실행 시간이 일찍 가파르게 상승하였다.

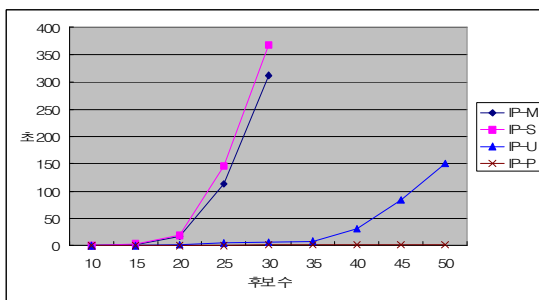


그림 2. 실행 시간의 비교
Fig. 2 Comparison of execution time

최장 경로에 대한 응답시간 제약을 적용한 IP-S와 병행을 고려하지 않은 채 응답시간 제약을 적용한 IP-U는 둘 다 하나의 제약 조건식만 갖는다는 점은 동일하지만, IP-S가 IP-U보다 제약 조건식에서 항의 개수가 1/3정도 적은 차이가 있다. 즉, 6개의 서비스 타입 가운데 최장 경로에 포함되지 않은 2개의 서비스 타입의 후보들에 대한 응답시간

값 항목들이 제외된 것이다. 그러나 실험 결과는 IP-S의 경우 실행시간이 빠르게 증가한 반면, IP-U는 완만히 증가하였는데, 이는 제약조건의 항의 개수가 줄어들었으므로 탐색 공간이 넓어짐으로 인하여 IP-S가 훨씬 많은 실행시간을 소요한 것으로 보인다. 실험에서 IP-P, 즉, 우세한 서비스에 의하여 최적 해에 포함될 수 없는 후보들을 전처리 작업을 통하여 제거한 방법은 매우 뛰어난 결과를 보였다.

각 서비스 개수에 대하여 서비스 타입 별로 상당 수의 항들이 우세 서비스에 의하여 제거되었고, 후보 수가 많을수록 그 비율은 더 높았는데, 그림 3은 제약수가 1~3개일 때 IP-M과 대비하여 IP-P의 서비스 수의 비율을 보여주고 있다. 서비스 개수가 적고 제약 개수가 많을수록 유효 후보, 즉, 비열세(non-dominated) 후보의 비율은 높고, 반대로, 서비스 개수가 많고 제약 개수가 적을수록 유효 후보의 비율이 낮아지는 것을 확인할 수 있다. 서비스 수가 10개일 때, 제약 개수에 따라 유효 서비스의 비율이 각각 33%, 48%, 73% 정도에 머물렀다. 서비스 수가 증가할수록 유효 서비스 비율이 각각 줄어들 뿐 아니라 이들 간의 격차도 점차 줄어드는 것을 알 수 있다. 즉, 서비스 수가 50일 때에는 비율이 각각 4%, 14%, 23%였다. 후보 서비스 수가 증가함에도 불구하고 유효 후보 수는 오히려 감소한 이유는 많은 수의 후보 수가 생성될 때 “매우” 파레토 우세한 후보가 발생할 확률이 높고, 그러한 후보에 의하여 많은 열세 후보들이 제거되기 때문이다. 후보 서비스 수가 증가할수록 “매우” 파레토 열세인 후보가 발생할 확률도 마찬가지로 높아지지만, 이러한 후보는 자신만 제거될 뿐이기 때문에 전체적인 유효 서비스 비율에 큰 영향을 미치지 못하는 것이다. 한편, 전처리에 소요되는 시간은 1초 이내로써 거의 무시할 수 있는 수준이었다.

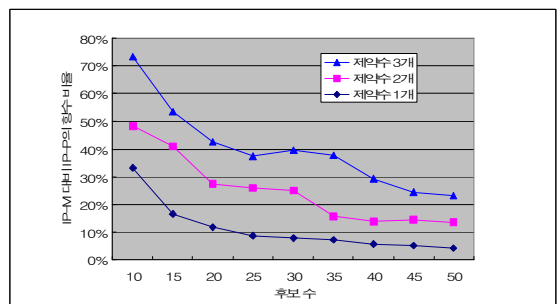


그림 3. IP-P의 서비스 수의 비율
Fig. 3 Ratio of the number of services of IP-P

4.2.2 후보 수 변화에 대한 해의 목표함수 값 비교

OPBDP 프로그램은 탐색해, 즉, 이진 변수들 가운데 값이 1인 변수와 함께 그때의 목표함수의 값(Global Maximum)을 결과 값으로 제공하는데, 본 절에서는 4.2.1 절의 실험에서 각 방법에 대한 실험 결과로서 목표함수 값을 비교한다. 따라서 병행 중점 수, 후보 수, 응답시간 제약 값 등 실험 환경은 4.2.1절과 동일하다. IP-P가 최적해이므로 최적 해에 대한 결과 값에 대한 IP-S와 IP-U의 결과 값의 비율을 그래프로 나타내었다. 즉, IP-P를 1로 보았을 때 IP-U와 IP-S의 최적 해에 대한 근접도를 분석한 것이다. IP-M은 해의 결과값이 IP-P와 동일하며, IP-S는 후보 수가 30일 때 까지만 측정된 것이다.

그림 4에서 보는 바와 같이 IP-U의 경우에 후보 숫자가 많을수록, 즉, 이득 값과 응답시간 값이 다양한 후보들이 많이 존재할수록 최적 해에 근접하는 결과를 얻을 확률이 높아진다는 사실을 확인할 수 있다. IP-S의 경우에는 모두 최적 해를 탐색했는데, 이는 본 실험에서 서비스 후보들의 응답시간 값의 생성범위가 동일하게 주어졌기 때문에 최장 길이의 경로를 대상으로 응답시간 제약조건에 의하여 후보가 결정됨으로 인한 자연스런 결과이다. 서비스 타입 별로 응답시간 값의 편차가 매우 크다면 최장 길이보다 짧은 병행 경로에서 제약조건을 위반하여 재탐색을 실시해야 하는 일이 발생할 것이다. 후보 수가 20개 이상부터 IP-U는 최적 해에 95% 이상 근접하는 결과를 보였다. 이러한 근접도는 후보 별 응답시간 값의 범위와 후보 수와 밀접한 관계가 있을 것으로 보인다. 또한 이 경우 병행 작업 경로를 고려하지 않은 해이기 때문에, 선택된 후보 서비스를 이용하여 실행시간을 다시 계산하면 응답시간 제약 값보다 짧은 예상 응답시간 값을 나타내게 될 것이다.

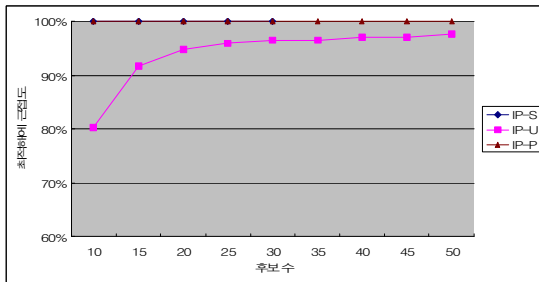


그림 4. 해의 근접도 비교
Fig. 4 Comparison of closeness to optimum

4.3 병행 분기 수의 변화에 대한 실험

4.3.1 분기 수의 변화에 대한 실행 시간의 비교

앞선 4.2절과 반대로, 서비스 타입 별 후보 서비스의 개수를 고정시킨 채 병행 분기가 중첩되는 수를 1에서 5까지 1씩 증가시키며 네 가지 방법의 해의 탐색시간을 측정하였다. 후보 서비스의 수는 10개로 고정하였고, 응답시간 제약 값은 분기 수가 1일 때에는 30, 5일 때에는 70이 되도록 10씩 증가시켰다. 그 이유는 분기 수가 커질수록 경로의 길이도 길어지기 때문에 최적 해를 구할 수 있을 정도의 충분한 값으로 조정할 필요가 있기 때문이다. 응답시간 제약 값이 해의 탐색시간에 미치는 영향에 대해서는 다음 4.4절의 실험에서 실시하였다.

그림 5는 각 모형에 대한 실행 시간을 보여준다. 분기 수가 3까지는 IP-S와 IP-M은 수 초 내에 해를 구하였으나 4부터 실행시간이 급격히 증가하였다. 전체 후보 수가 크지 않음에도 불구하고 오랜 탐색시간을 보인 것은 문제의 복잡도가 높다는 것을 뜻한다. 병행 분기를 고려하지 않은 IP-U는 4.2.1절의 실험에서와 마찬가지로 IP-S보다 실행시간의 증가 속도가 느렸고, 분기수가 5일 때에는 10분을 초과하는 과다한 수행시간을 보였다. IP-P는 전처리를 통하여 일관되게 짧은 실행시간을 보였으며, 제거되는 항의 수는 4.2절의 실험 결과와 비슷하였다.

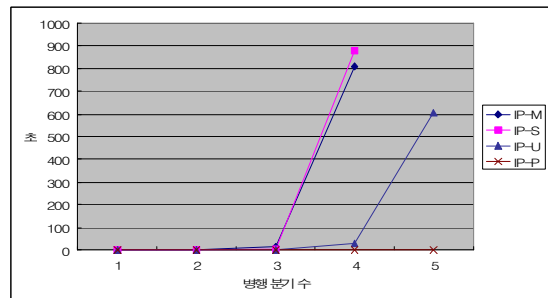


그림 5. 분기 수에 대한 실행시간 비교
Fig. 5 Comparison of execution time over the number of branches

4.3.2 분기 수의 변화에 대한 해의 결과값 비교

앞선 4.3.1절의 실험에 대하여 탐색 결과인 목표 함수 값을 비교한 것으로서, 병행 분기 수가 증가할 때 IP-U의 최적 해에 대한 근접도를 비교하였다. IP-S는 4.2절의 실험에서와 마찬가지로 최적해가 탐색되었다. IP-U는 그림 6에서 나타나 있듯이 병행 분기 수를 고정하고 후보 서비스 수만 변화시킨

4.2.2절의 결과와는 다소 다른 특성을 보였다. 분기 수가 1일 때에는 여러 번의 실험에도 불구하고 97% 정도의 근접도를 보였고, 병행 분기 수가 2일 때에는 4.2.2절의 실험 결과와 비슷한 근접도를 나타냄으로써 일관성을 보여주었다. 분기수가 늘어남에 따라 최적 해와의 근접도는 80% 중반 정도에 머물렀다. 이는 분기 수가 늘어날수록 전체 후보들을 대상으로 응답시간 제약을 적용하는 IP-U의 탐색결과가 최적해로부터 점차 멀어진다는 것을 보여주고 있다. 하지만 본 실험 데이터를 기준으로 볼 때, 실행시간을 고려하면 분기 수가 4 정도까지는 IP-M 또는 IP-S의 대안이 될 수도 있을 것이다.

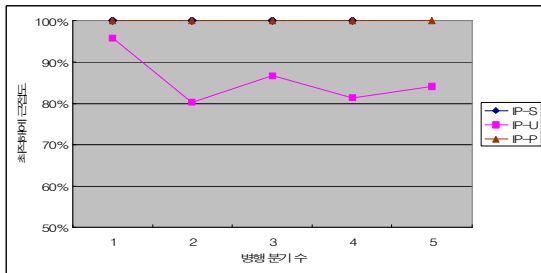


그림 6. 분기 수에 대한 최적해 근접도 비교
Fig. 6 Comparison of closeness to optimum over the number of branches

4.4 제약 값의 변화에 대한 실험

본 절에서는 응답시간 제약 조건 값의 변화가 해의 탐색 성능에 미치는 영향을 알아본다. IP-M과 IP-S는 실행 시간이 급격히 증가하기 때문에 IP-P와 IP-U를 대상으로 실행 시간과 탐색 해의 최적 해에 대한 근접도를 측정하였다. IP-U의 실행 시간을 폭넓게 측정하기 위하여 문제 크기는 병렬 분기 수를 2로 고정하고, 후보 서비스의 수 10, 20, 30, 40 각각에 대하여 응답시간 제약 값을 30부터 70까지 5씩 증가시키며 IP-P와 IP-U의 데이터를 생성하여 폭넓게 실험을 실시하였다.

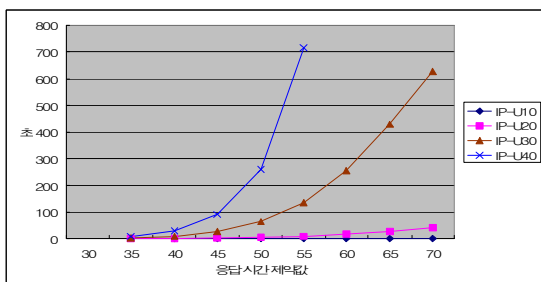


그림 7. 제약 값의 변화에 대한 실행시간의 비교
Fig. 7 Comparison of execution time over constraint values

그림 7은 각 후보 수에 대한 IP-U의 실행 시간을 보여주고 있다. 그림의 범례에서 IP-U10 등의 숫자 10은 후보 서비스의 수를 뜻한다. 전체적으로, 후보 수가 많을수록, 그리고 제약 값이 클수록 실행시간이 더 오래 걸렸다. 후보 수가 10개일 때 (즉, IP-U10)에는 제약 값이 증가함에 따라 실행시간이 미세하게 증가했지만 1~2초 이내에 실행을 완료하였다. IP-U20은 제약 값이 60일 때 최적 해를 탐색하였고, 제약 값이 65, 70으로 증가할 때 동일한 최적 해를 탐색하는데 더 많은 시간이 걸렸다. 후보수가 30개와 40개일 때에는 실행시간이 보다 빠른 속도로 증가하였는데 둘 사이의 실행시간의 차이도 기하급수적으로 증가하였다. 이는 제약 값이 클수록 탐색 공간이 넓어진 이유이다.

그림 8는 IP-U의 최적 해에 대한 근접도를 보여주고 있다. 제약 조건 값이 30일 때에는 모든 후보 서비스 수에 대하여 IP-U가 조건을 만족하는 해를 찾지 못하였고, 제약 조건 값이 35일 때에는 후보 수 10인 경우를 제외하고 IP-U가 해를 탐색하였다. 제약 조건 값이 작을수록 그리고 후보수가 적을수록 해를 찾지 못할 확률이 높다는 것을 확인할 수 있다. IP-U는 제약 조건 값이 커질수록 최적 해에 접근하는 값을 탐색하였다. 후보 서비스의 수가 많을수록 대체로 최적 해에 보다 근접한 해를 찾았으나, IP-U20은 예외적으로 IP-U30과 IP-U40보다 우수한 결과를 보였다.

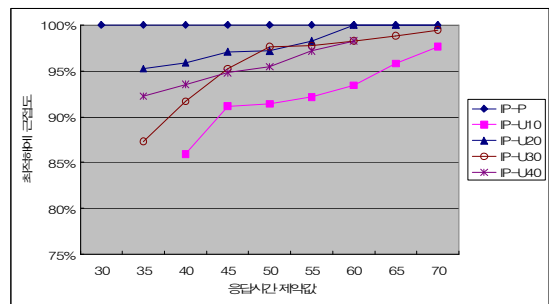


그림 8. 제약 값의 변화에 대한 최적 해 근접도 비교
Fig. 8 Comparison of closeness to optimum over constraint values

후보 서비스의 이득 값과 응답시간 값이 주어진 범위 내에서 임의 생성되었기 때문에 어느 정도 편차는 있을 수 있으나 여러 번의 실험에서 각 제약 조건 값의 경우에 대부분 우수한 결과를 보인 것은 흥미로운 일이다. 이에 대한 이유는 생성된 문제에 대한 유효 후보 서비스 수와 관련이 있는 것으로 보인다. 유효 후보 서비스, 즉, 동일한 서비스 타입에 속하면서 자신보다 이득과 응답시간이 동시에 우세한 후

보 서비스가 없는 후보 서비스의 수가 적을수록 구해지는 해는 최적 해에 빨리 접근할 것이다. 최적 해는 이들 유효 후보 서비스들 가운데서 구성된다. 실제로 IP-U10, IP-U20, IP-U30, IP-U40의 유효 후보 서비스의 수는 각각 20, 14, 22, 20개였다.

V. 결론

본 논문에서는 병행분기를 중심으로 복합 웹 서비스의 서비스 선택을 위한 응답시간 제약조건 적용 모형을 설명하고 실험을 통하여 각 방안의 성능을 살펴보았다. 실험 데이터 모형을 이용하여 후보 웹 서비스의 수, 병행 분기의 중첩 수, 제약 조건 값의 크기 등을 변화시키면서 네 가지 방안의 해의 탐색 성능을 관찰하는 폭넓은 실험을 실시하였다. 전처리를 통하여 열세 후보를 배제하는 것은 매우 효과적이며, 중소기업의 경우에 전체 서비스 타입 집합에 대하여 하나의 응답시간 제약을 적용하는 방안도 절충안이 될 수 있음을 알 수 있었다.

앞으로 다양한 분야에서 웹 서비스가 활용되면 응용 분야의 특성에 따른 QoS 제약 또한 다양하게 요구될 것이다. 이들 새로운 QoS 항목을 도출하고 복합 웹 서비스에 대한 이들 QoS 항목의 모형을 수립해 나가는 연구가 필요할 것이다. 또한 본 연구에서는 몇 가지 가정을 통하여 연구 범위를 제한하였는데, 제약조건이 매우 많고 비즈니스 프로세스의 작업흐름 유형이 복잡한 환경에서 서비스 선택 방안에 대한 연구가 계속 필요할 것으로 생각된다.

참고문헌

- [1] Erl, T., *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services*, Prentice Hall, 2004.
- [2] Kuster, U. et al, "DIANE An Integrated Approach to Automated Service Discovery, Matchmaking and Composition," *Proceedings of WWW2007 Conference*, 2007.
- [3] Weerawarana, S., et al, *Web Services Platform Architecture*, Prentice Hall, 2005.
- [4] Juric, M. B., Mathew, B. and Sarang, P., *Business Process Execution Language for Web Services*, Packt Publishing, 2004.
- [5] Oh, S.C. et al, "A Comparative Illustration of AI Planning-based Web Services Composition," *ACM SIGecom Exchanges*, Vol. 5, No. 5, Dec. 2005.
- [6] Fan, J. and Kambhampati, S., "A Snapshot of Public Web Services," *ACM SIGMOD Record*, Vol. 34, No. 1, March 2005.
- [7] Aggarwal, R., Verma, K., Miller, J. and Milnor W., "Constraint Driven Web Service Composition in METEOR-S," *Proc. of IEEE International Conference on Services Computing (SCC)*, 2004.
- [8] Benatallah, B., et al, "On automating Web Services discovery," *VLDB Journal*, Vol. 14, 2005.
- [9] Maximilien, E. M. and Singh, M. P., "A Framework and Ontology for Dynamic Web Service Selection," *IEEE Internet Computing*, September-October, 2004.
- [10] Kerrigan, M., "Web Service Selection Mechanisms in the Web Service Execution Environment," *Proceedings of ACM SAC '06 Conference*, 2006.
- [11] Yu, T. and Lin, K.-J., "Service Selection Algorithms for Web Services with End-to-End QoS Constraints," in *Proceedings of the IEEE International Conference on E-Commerce Technology*, 2004.
- [12] Zeng, L., et al, "QoS-Aware Middleware for Web Services Composition," *IEEE Transactions on Software Engineering*, Vol. 30, No. 5, May 2004.
- [13] Pisinger, D., "A Minimal algorithm for the Multiple-choice Knapsack Problem," *European Journal of Operational Research*, 83, 1995.
- [14] Freitas, A. A., "A Critical Review of Multi-Objective Optimization in Data Mining: a position paper," *ACM SIGKDD Explorations Newsletter*, Vol. 6, Issue 2, 2004.
- [15] Barth, P., "Davis-Putnam Based Enumeration Algorithm for Linear Pseudo-Boolean Optimization," *MPI-I-95-2-003, MPI Informatik*, 1995.

저 자 소 개



서상구

1984년 서울대학교 컴퓨터공학과
(학사)

1986년 KAIST 전산학과 (석사)

1995년 KAIST 전산학과 (박사)

1986년~1989년 현대전자 S/W개발부

1995년~1998년 현대전자 정보시스템사업부

1999년~현재 광운대학교 부교수

관심분야 : 데이터베이스 시스템,

웹 데이터 관리, 웹 서비스

email: skseo@kw.ac.kr