

웹 서비스의 선택과 조건 분기에 관한 연구*

서 상 구**

A Study on Web Services Selection and Conditional Branches*

Sangkoo Seo**

■ Abstract ■

IT Services market is growing rapidly in the business industry and SOA-based Web Services have been introduced as an effective vehicle for the integration of enterprise-wide applications within organizations. The number of publicly available Web Services is ever increasing recently in a variety of areas, and as the number of public Web Services increases, there will be many Web 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 right service while satisfying given QoS constraints. This paper addresses the issue of selecting composite Web Services which involves conditional branches in business processes. It is essential to have any conditional branches satisfy the global QoS constraints at service selection phase, since the branches are chosen to execute at run-time dynamically. We proposed service selection procedures for basic structure of conditional branches and explained them by examples. Experiments were conducted to analyze the impact of the number of candidate services and service types on the time of finding service solutions.

Keyword : Web Services, Service Selection, Conditional Branches

1. 서 론

서비스 지향 아키텍처(SOA: Service-Orien-

ted Architecture)는 인터넷 환경에서 정보 시스템의 통합 구축을 위한 새로운 패러다임으로서 IT서비스의 SI 및 솔루션 분야의 중요 요소로 많

* 이 논문은 2005년도 광운대학교 연구년에 의해 연구되었음.

** 광운대학교 경영대학 경영정보학과 교수

은 관심을 받고 있다[5, 10, 15]. SOAP, WSDL 및 UDDI 표준을 바탕으로 하는 웹 서비스(Web Services)는 서비스 지향 아키텍처를 구현하는 가장 대표적인 응용 형태로 인정받고 있다[5, 8, 15]. 웹 서비스는 다른 웹 서비스를 호출하지 않고 자신의 서비스 실행 결과만 반환하는 단순 웹 서비스와 BPEL 프로세스와 같이 하나의 서비스 내에서 프로세스 실행 흐름에 따라 외부의 다른 서비스 호출을 포함하는 복합 웹 서비스의 형태가 있다. 복합 웹 서비스의 개발은 응용 설계자가 실행 흐름과 함께 필요한 웹 서비스(본 논문에서는 서비스 타입이라 부른다)들을 구성한 후, 각 서비스 타입에 적합한 서비스(후보 서비스라고 부른다)를 응용 개발자 또는 응용 전문가가 결정하는 것이 일반적인 방식이다. 다양한 응용 분야에서 점차 많은 수의 공개 웹 서비스들이 등장하고 있는데 이럴 경우 개발자 또는 응용 전문가가 직접 웹 서비스를 탐색하여 가장 적합한 후보 서비스를 선택하는 작업은 용이하지 않다[6]. 따라서 웹 서비스 선택을 자동화하기 위한 많은 연구가 진행되어 왔는데, 이러한 웹 서비스의 선택문제는 두 가지 유형으로 나뉜다. 첫째는 주어진 기능적 요구 조건에 부합되는 웹 서비스 종류를 식별하는 문제로서, 서비스의 내용, 입출력 변수의 개수와 데이터 타입 등을 고려한 의미적 매칭 기법을 이용한 다양한 연구가 진행되고 있다[1, 3, 11, 12]. 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.

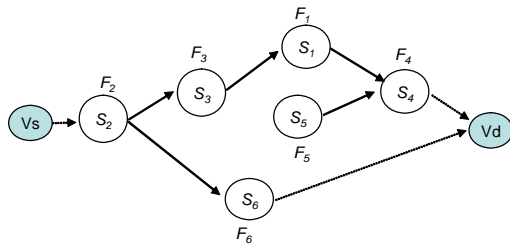
본 논문에서는 웹 서비스의 조건 분기(Conditional Branches)를 고려하여 복합 웹 서비스에 대한 후보 서비스를 결정하는 방안을 제안한다. QoS에 기반한 웹 서비스의 선택과 관련하여 많은 연구가 진행되어 왔다[4, 9, 10, 14, 16, 17, 18]. 대표적으로, [18]에서는 복합 웹 서비스 구축을 위한 전역 계획 최적화 방법을 제안하였는데,

복합 웹 서비스의 각 실행 경로에 대하여 총 비용, 총 실행 시간 등에 대한 제약조건을 고려하여 최적의 후보 서비스를 선택하도록 하였다. 여러 경로에 공통으로 존재하는 서비스 타입에 대한 후보 서비스의 결정은 실행 빈도 또는 사용자가 선호하는 것을 선택하는 방식으로 전체 복합 웹 서비스를 구현하였다. [16, 17]에서는 복합 웹 서비스의 총 지연시간에 대한 제약조건을 충족시키는 서비스 집합을 선택하는 알고리즘을 연구하였다. 이를 위하여 웹 서비스 문제를 Knapsack 문제로 모형화하고 여러 실행경로 가운데 최적화 목표 함수 값이 가장 큰 경로를 최적 실행 경로로 결정하는 알고리즘을 연구하였다.

대부분의 앞선 연구들은 하나의 서비스 타입에 대하여 서비스를 선택하기 위하여 다중 목표 제약에 따른 최적 선택을 시도하였거나, 복합 웹 서비스의 경우에는, 여러 실행 경로 가운데 주어진 제약조건과 목표치를 가장 만족하는 하나의 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, 각 경로 별로 최적 후보 서비스를 선택하였다도 여러 경로에 공통으로 존재하는 서비스 타입에 대하여 서로 다른 후보 서비스를 결정하였다면 그 중 하나만 선택해야 하는데 이런 경우 전체 QoS 제약조건을 만족시키기 위한 방안은 충분히 고려되지 않았다. 또한 많은 연구에서 복합 웹 서비스 프로세스의 모든 가능한 실행 경로에 대한 해를 탐색하는데 이는 많은 탐색시간을 요구할 수 있다. 본 연구에서는 이전 연구에서와 같이 Knapsack에 기반한 Integer Programming으로 복합 웹 서비스의 선택을 모형화하되, 조건 분기를 고려하여 서비스를 선택하는 방안을 제안한다. 본 연구에서 제안하는 서비스 선택의 기본 아이디어는 확률적으로 실행 가능성이 높은 조건분기의 경로에 대한 최적 해를 우선적으로 탐색하되, 전체 경로를 대상으로 하는 해의 탐색 횟수를 가능한 줄이며 실행 확률이 낮

은 조건분기 경로들에 대해서도 제약조건을 만족하는 해를 생성하는 것이다. 따라서 본 연구의 결과는 IT서비스에서 SOA 기반의 정보시스템 구축시에 조건 분기를 갖는 복합 서비스 프로세스의 효과적인 구성과 서비스 선택을 위한 기초 자료로 활용될 수 있을 것이다.

본 논문은 다음과 같이 구성된다. 제 2장에서는 예를 통하여 서비스 선택 문제를 살펴보고 다차원 다중선택 Knapsack에 기초한 최적 선택 모형을 설명한다. 제 3장에서는 조건 분기를 고려한 서비스 선택 절차를 제시하고, 제 4장에서는 서비스 타입 수와 후보 수가 탐색 시간에 미치는 영향을 실험을 통하여 알아본다. 그리고 제 5장에서 논문을 맺는다.



(a) 복합 웹 서비스 프로세스

기능	서비스 타입	후보 서비스	Utility	실행 시간	가격	가용성
F ₁	S ₁	S ₁₁	212	100	50	0.95
		S ₁₂	219	180	60	0.92
F ₂	S ₂	S ₂₁	195	200	50	0.98
		S ₂₂	123	160	100	0.95
		S ₂₃	150	180	80	0.97
F ₃	S ₃	S ₃₁	216	150	100	0.94
		S ₃₂	160	120	80	0.99
F ₄	S ₄	S ₄₁	150	130	60	0.93
		S ₄₂	200	140	40	0.99
F ₅	S ₅	S ₅₁	231	200	150	0.96
F ₆	S ₆	S ₆₁	162	200	100	0.97
		S ₆₂	123	180	130	0.99

(b) QoS 값

[그림 1] 복합 서비스 프로세스의 예[17]

2. 웹 서비스 선택 문제

2.1 서비스 선택 문제의 예

복합 웹 서비스의 선택 문제란 각각의 서비스 타입에 해당되는 후보 서비스들 가운데 하나씩의 후보 서비스를 결정하되, 웹 서비스에 대한 총 이득이 가능한 최대화되며 이와 동시에 가격 총액, 전체 응답시간 등에 대한 제약 조건이 주어질 경우 이를 만족하도록 하는 문제이다. [그림 1]은 복합 웹 서비스 프로세스와 후보 서비스의 QoS 값을 보여주고 있다.

[그림 1-a]에는 두 개의 분기가 포함되어 있는데, 이들을 모두 조건 분기로 가정하면 모두 4개의 실행 경로, 즉, {F₁, F₂, F₃, F₄}, {F₁, F₂, F₆}, {F₅, F₂, F₃, F₄}, {F₅, F₂, F₆}가 있을 수 있다. 각 함수 F_i는 해당되는 서비스 타입 S_i에 속한 임의의 후보 서비스를 실행함으로써 처리될 수 있다. 전체 복합 서비스의 실행에 대한 QoS 제약조건이 총 실행시간은 600이하, 총 비용은 250이하, 그리고 전체 가용성이 85% 이상으로 주어진다면, 서비스 선택 문제는 각 서비스 타입 S_i에 대한 후보 서비스 s_{ij}를 선택하되 선택된 서비스들에 대한 실행 경로가 실행시간의 합, 가격의 합, 그리고 가용성의 곱이 각각 주어진 제약조건을 만족시키면서 Utility, 즉, 각 서비스의 선택으로 인한 이득의 합이 크도록 하는 것을 말한다.

[그림 1-a]에서 선택 가능한 후보 서비스의 조합은 모두 48개이지만, 실행 가능한 경로 별 후보 서비스의 조합을 따져보면 모두 54가지가 있다. 즉, 앞에서 설명한 경로를 각각 F₁₂₃₄, F₁₂₆, F₅₂₃₄, F₅₂₆이라고 하면 F₁₂₃₄에는 24가지, F₁₂₆과 F₅₂₃₄에는 각각 12가지, F₅₂₆에는 6가지의 방법으로 후보를 선택할 수 있다. 그러나 이 중 전체 제약조건을 만족하는 수는 F₁₂₃₄에는 {S₁₁, S₂₁, S₃₁, S₄₂}, {S₁₁, S₂₁, S₃₂, S₄₂}, {S₁₁, S₂₃, S₃₂, S₄₂}, {S₁₁, S₂₁, S₃₂, S₄₁}의 4가지이고, F₁₂₆에는 {S₁₂, S₂₁, S₆₁}, {S₁₁, S₂₁, S₆₁} 등을 포함한 7가지이나, 경로 F₅₂₃₄, F₅₂₆에서

는 제약 조건을 만족하는 서비스 선택이 하나도 없다. 따라서 모든 실행 가능한 경로에 대해서 제약 조건을 만족하기 위해서는 주어진 QoS 제약 값의 일부 또는 모두가 조정되어야 한다. 즉, 실행 시간과 가격의 상향 조정되어야 하고 가용성은 하향 조정되어야 할 것이다. 또 다른 쟁점은 F_{1234} 의 4가지 가능한 선택 조합 가운데 이득함계가 가장 큰 것은 $\{S_{11}, S_{21}, S_{31}, S_{42}\}$ 이고 F_{126} 경로에서 이득함계가 가장 큰 것은 $\{S_{12}, S_{21}, S_{61}\}$ 인데 이럴 경우 S_1 의 S_{11} 과 S_{12} 가운데 어느 후보를 선택할 것인지에 대해서다. 특히 S_{12} 를 선택하면 F_{1234} 경로에서는 제약조건을 만족하는 해가 없게 되는데, 다행히 이 예제의 경우에는 S_{11} 을 택하는 것이 F_{126} 에 대해서 최적은 아니지만 제약조건을 만족할 수 있으므로 가장 적절한 선택일 것이다.

후보 서비스의 수가 많고 복합 웹 프로세스의 서비스 타입 수가 많아지면 보다 체계적인 선택 방안이 요구된다. 또한 위 예에서처럼 제약조건을 만족하는 모든 후보 선택 조합 가운데 경로의 이득 값이 가장 큰 $\{S_{11}, S_{21}, S_{31}, S_{42}\}$ 만 결정하는 것으로는 불충분하다. 실행 가능한 모든 경로에 대하여 제약 조건을 만족하는 후보 서비스를 결정하는 절차가 필요한 것이다.

2.2 복합 웹 서비스의 QoS 모형

본 절에서는 최적 서비스 탐색 모형의 설정에 관련된 몇 가지 가정을 설명한다. 각 웹 서비스에 대한 이득 값(또는 Utility 값)은 여러 QoS 값을 종합적으로 고려하여 계산되는 값으로 가정한다 [17, 18]. 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 이득 값으로 간주한다.

제약 조건은 가격, 응답시간 등의 경우와 같이 주어진 값보다 같거나 작은 부등식으로 표현된다고 가정한다. 2.1절의 예에서의 가용성과 같은

QoS 항목은 주어진 값 이상의 제약조건 형태인데, 이런 경우에는 $(1 - \text{가용성})$ 으로 변환하여 같거나 작은 부등식으로 표현할 수 있다.

복합 웹 서비스는 BPEL(Business Process Execution Language) 등과 같이 하나 이상의 단순 또는 복합 웹 서비스들이 순차, 반복, 분기 등의 실행 흐름에 따라 구성된다[8]. 본 논문에서 임의의 경로에 대한 순차적 실행 흐름에 대한 QoS 값은 QoS 값의 합계로 정의한다. 예를 들면, [그림 1]에서 선택된 경로 $\{S_{11}, S_{21}, S_{31}, S_{42}\}$ 에 대한 QoS 가격 항목의 값은 $590(=100 + 200 + 150 + 140)$ 이다. 가용성과 같은 QoS 값은 앞서 설명한 바와 같이 $(1 - \text{가용성})$ 으로 변환한 후 로그함수를 적용하면 곱을 합으로 나타낼 수 있다. 병행분기는 각 분기가 각각 실행되기 때문에 가격의 경우 분기에 포함되는 모든 웹 서비스의 가격의 합으로 정의되고, 응답시간은 각 분기 경로 별로 포함된 서비스의 응답시간 합계 가운데 최대인 경로의 값을 취한다. 조건 분기의 경우에는 실행 시점의 분기 기준 값에 따라 각 분기가 선택적으로 실행되는데, 이전의 실행 이력이나 응용의 특성에 따라 분기별로 실행 가능 확률 값이 주어진다고 가정한다. 만약 확률 값이 주어지지 않으면 각 분기가 동일한 확률 값을 갖는 것으로 가정할 수 있다. 조건 분기에 대한 QoS 값은 각 실행 가능한 경로 별로 QoS 항목의 합계를 구하고, 가장 큰 값이 분기를 대표하는 QoS 값이다. [그림 1]의 예에서 F_{34} 분기 경로와 F_6 의 분기 경로에 대하여 후보 $\{S_{32}, S_{42}\}$ 과 $\{S_{62}\}$ 를 각각 선택하였다면, 응답시간은 $260(=120 + 140)$ 과 180 가운데 260 을, 가격은 $120(=80 + 40)$ 과 130 가운데 130 을 분기 구간을 대표하는 QoS 값으로 간주한다.

반복 흐름은 최대 반복 횟수, 예상 반복 횟수, 또는 과거의 실행에서 평균 반복횟수만큼 반복 블록내의 웹 서비스 프로세스가 중복되어 있다고 간주함으로써 QoS 값을 정의할 수 있다. 본 연구에서는 조건 분기에 대한 서비스 선택을 대상으로 하기 때문에 반복 흐름과 병행 분기는 별도로 고

려하지 않기로 한다.

2.3 서비스 선택의 최적화 모형

복합 웹 서비스의 최적 구성 문제는 후보 서비스를 선택한다는 점에서 Knapsack 문제와 본질적으로 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 것도 다중선택 Knapsack과 일치한다[13]. 그리고 웹 서비스의 가격, 응답시간, 가용성 등의 QoS 속성들과 복합 웹 서비스의 선택에 대한 제약조건은 다차원 Knapsack의 제약 속성 및 제약 조건으로 간주할 수 있다[13]. 본 논문에서 서비스 선택 최적화 수식에 사용되는 기호와 의미는 아래 <표 1>과 같다.

추가적으로 조건분기에 대한 QoS 제약을 반영하기 위해서는 실행 가능한 경로 집합에 대하여 제약조건을 표현해야 한다. 실행 가능한 경로 집

합이란 복합 웹 서비스의 시작부터 종료까지 2.1절의 가정을 전제로 분기에 의하여 나뉘어지는 모든 서로 다른 경로들을 뜻한다. 특히 조건 분기의 후보 서비스들 가운데 어떤 서비스가 선택되더라도 전체 QoS 제약조건이 만족되어야 하므로 모든 실행 가능한 경로 집합에 대하여 각 제약조건을 적용해야 한다. 복합 웹 서비스의 실행 가능한 경로가 모두 R 개 있다고 하고 각각을 $path_r (1 \leq r \leq R)$ 이라고 하자. 또, $path_r$ 에 속한 서비스 타입의 집합을 T_r 이라고 하자. 2.2절의 QoS 모형에 대한 가정을 바탕으로, 다차원 다중선택 Knapsack 모형을 이용하여 복합 웹 서비스 최적 설계 문제를 아래와 같이 수식화할 수 있다.

$$\text{Maximize } \sum_{i=1}^M \sum_{j=1}^{N_i} util_{ij} x_{ij} \quad (1)$$

$$\text{Subject to } \sum_{t_i \in T_r} \sum_{j=1}^{N_i} q_{ijk} x_{ij} \leq W_k, \quad \text{for each } k \text{ and } path_r, \quad (1 \leq r \leq R, 1 \leq k \leq K). \quad (2)$$

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

목표함수 식 (1)은 복합 웹 서비스의 이득이 최대가 되도록 모든 후보 서비스들의 이득을 합산하는데, 제약 조건 식 (3)에 의하여 하나의 서비스 타입에서 하나의 후보 서비스가 결정되도록 제한한다. QoS 제약 조건 식 (2)은 각 실행 가능 경로에 대하여 서비스 타입 별로 하나씩 결정되는 후보 서비스들의 각 QoS 값의 합계가 해당 제약조건 값을 만족하도록 한정한다. 위 최적화 수식은 목표함수와 함께 다중선택 요건에 대한 M 개의 제약조건 부등식과 각 경로 별로 다차원 제약에 대한 QoS 제약조건 수 K 개의 부등식으로 구성되며 $\sum_{i=1}^M N_i$ 개의 이진 결정변수를 포함한다.

<표 1> 수식에 이용되는 기호와 정의

표기	의 미
M	복합 웹 서비스에 포함된 서비스 타입의 수
t_i	복합 웹 서비스에 포함된 서비스 타입 ($1 \leq i \leq M$)
N_i	서비스 타입 t_i 에 속한 후보 서비스 집합 ($1 \leq i \leq M$)
K	QoS 제약 항목의 수
s_{ij}	서비스 타입 t_i 의 후보 서비스 ($1 \leq i \leq M, 1 \leq j \leq N_i$)
x_{ij}	후보 서비스 s_{ij} 에 대한 이진 결정변수
$util_{ij}$	후보 서비스 s_{ij} 의 이득 (utility)
Q_k	웹 서비스의 k 번째 QoS 항목 ($1 \leq k \leq K$)
q_{ijk}	후보 서비스 s_{ij} 의 서비스 항목 Q_k 의 값 ($1 \leq i \leq M, 1 \leq j \leq N_i, 1 \leq k \leq K$)
W_k	QoS 항목 Q_k 의 전체 제약 값 ($1 \leq k \leq K$)

3. 조건 분기와 서비스 선택

3.1 접근 방법

복합 웹 서비스의 모든 서비스들이 결정되어 실행하게 될 경우, 조건 분기에 도달하면 직전 서비스의 실행 결과에 따라 여러 분기 중의 한 분기로 진행된다. 그러나 웹 서비스의 선택 시점에는 어떤 경로로 실행될지를 미리 알 수 없다. 따라서 모든 가능한 실행 경로에 대하여 주어진 전체 제약 조건을 만족하면서 가능한 전체 이득이 크도록 서비스를 선택해야 한다. 본 연구의 기본 아이디어는 전체 경로와 제약조건을 고려하여 각 조건 분기의 실행 확률이 높은 순으로 분기 경로의 서비스를 선택하되, 가능한 해의 탐색 횟수와 시간을 줄이는 방향으로 탐색하는 것이다.

임의의 경로 또는 구간을 대표하는 QoS 값은 각 QoS 항목별로 구성되기 때문에 배열의 형태가 될 것이다. 배열과 관련된 설명을 간단히 하기 위하여 <표 1>에 추가적으로 다음 <표 2>의 정의를 사용한다.

조건 분기의 수, 조건 분기 내의 경로 수, 조건 분기의 중첩 등에 따라 복합 웹 서비스의 형태는 복잡할 수 있고 이를 모두 고려한 서비스 선택 절

차를 한번에 설명하는 것은 매우 복잡할 뿐 아니라 이해하기도 용이하지 않을 것이다. 따라서 본 논문에서는 조건 분기의 간단한 형태별로 서비스 선택 절차를 기술하기로 한다. 3.2절에서는 하나의 조건 분기가 두 개의 경로를 갖는 기본 형태를 대상으로 설명하고, 3.3절은 하나의 조건 분기에 여러 개의 경로가 있는 경우를 설명하고 3.4절은 조건 분기가 순차적으로 여러 개 있는 경우, 3.5절은 조건 분기가 중첩된 경우로 나눠서 설명한다.

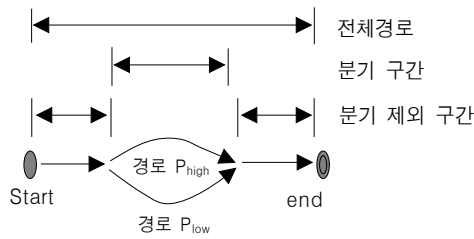
3.2 하나의 조건 분기의 기본 형태

다음 [그림 2]와 같이 복합 웹 서비스의 시작부터 종료까지를 전체경로라고 하고, 조건분기가 시작되어 종료되는 구간을 분기구간이라고 하며, 전체경로에서 분기구간을 제외한 부분을 분기제외 구간이라고 하자. 또 조건분기에서 분기 확률이 큰 경로를 p_{high} 라고 하고 확률 값이 작은 경로를 p_{low} 라고 부르기로 한다. 확률 값이 동일한 경우에는 임의의 한 경로를 p_{high} 로 나머지를 p_{low} 로 한다. 하나의 조건 분기에 여러 개의 p_{low} 가 있는 경우와 여러 개의 조건 분기가 있는 경우는 다음 절에서 설명한다.

서비스 선택 절차는 먼저 실행 확률이 높은 조

<표 2> QoS 배열에 대한 기호와 연산 정의

표기	의 미
W	복합 웹 서비스의 전체 제약조건 값을 갖는 배열로서, QoS 항목 수 만큼의 크기이고, W_k 는 QoS 항목 Q_k 에 대한 복합 웹 서비스의 전체 제약조건 값이다($1 \leq k \leq K$).
$\theta(p)$	경로 또는 구간 p 를 대표하는 QoS로서, QoS 항목 수(K) 만큼의 배열이다
$\theta(p)_k$	배열 $\theta(p)$ 의 k 번째 값으로서 경로 또는 구간 p 에서 선택된 후보 서비스들의 QoS 항목 Q_k 값의 합계이다($1 \leq k \leq K$).
$\theta(p_1) \oplus \theta(p_2)$	\oplus 는 배열의 항목별 최대값을 선택하는 연산자로서, 연산의 결과는 배열이며, 배열의 k 번째 값은 $\theta(p_1)_k$ 과 $\theta(p_2)_k$ 중 큰 값이다($1 \leq k \leq K$).
$\theta(p_1) < \theta(p_2)$	<는 배열의 비교 연산자로서, 모든 k ($1 \leq k \leq K$)에 대하여 $\theta(p_1)_k \leq \theta(p_2)_k$ 을 만족할 때 참이고, 그렇지 않으면 거짓이다.
$\theta(p_1) + \theta(p_2)$	배열에 대한 덧셈으로서 연산의 결과는 배열이며, k 번째 값은 $\theta(p_1)_k + \theta(p_2)_k$ 이다($1 \leq k \leq K$).
$\theta(p_1) - \theta(p_2)$	배열에 대한 뺄셈으로서 연산의 결과는 배열이며, k 번째 값은 $\theta(p_1)_k - \theta(p_2)_k$ 이다($1 \leq k \leq K$).



[그림 2] 분기 경로의 기본 형태

건 분기 경로, 즉, p_{high} 를 포함하는 전체 경로의 해에 대한 QoS 값의 배열 $\Theta(p_{high})$ 을 구하고, $\Theta(p_{high})$ 를 제약조건으로 하여 나머지 분기 경로 p_{low} 의 해를 탐색한다. 탐색된 해는 $\Theta(p_{high})$ 의 각 QoS를 만족하므로 전체적인 제약조건 W 도 만족하게 된다. 만일 해를 구하지 못할 경우, 이는 실행 확률이 낮지만 실제로 실행되었을 경우 전체적인 QoS 제약 조건 W 을 만족하지 못하는 결과를 초래할 수 있다는 의미이고, $\Theta(p_{low})$ 가 $\Theta(p_{high})$ 보다 더 큰 제약 값을 요구한다는 것이다. 실행 시점에서 조건 분기 이전에 이미 실행한 서비스들의 실행을 취소하고 새로운 해를 탐색한다는 것은 현실적으로 타당하지 않기 때문에 서비스 선택 시점에서 이러한 문제가 발생하지 않도록 해야 할 것이다.

이를 위해서 실행 확률은 낮지만 QoS 제약이 큰 조건 분기 p_{low} 를 포함하는 전체 경로에 대하여 최적 해를 탐색한다. 이미 탐색했던 p_{high} 분기의 서비스들과 p_{low} 분기 구간에서 새로 선택된 서비스들에 대하여, $\Theta(p_{high}) < \Theta(p_{low})$ 가 성립한다면, 즉, 각 QoS 항목에 Q_k 에 대하여 $\Theta(p_{high})_k \leq \Theta(p_{low})_k$ 라면 p_{high} 분기의 서비스들은 여전히 유효한 해라고 할 수 있다. 왜냐하면 $(\Theta(\text{분기제외구간}) + \Theta(p_{high})) < (\Theta(\text{분기제외구간}) + \Theta(p_{low}))$ 로서 전체 제약 조건을 여전히 만족하기 때문이다. 이때 p_{high} 분기를 포함하는 경로가 실행될 때의 실행 이득, 즉, 결정함수의 값은 낮아질 수 있다. 이는 p_{low} 분기가 전체적인 제약조건을 만족하는 실행 계획을 생성하기 위해서는 불가피하다. 전체

실행이득은 확률 값과 각 분기에서 결정되는 서비스의 이득 값의 합계에 따라 결정될 것이다.

만약 위의 QoS 값의 비교에서 $\Theta(p_{high}) < \Theta(p_{low})$ 가 거짓이면, 즉, 하나 이상의 QoS 항목 Q_k 에서 $\Theta(p_{low})_k \leq \Theta(p_{high})_k$ 이면, p_{high} 분기에서 선택된 서비스들은 더 이상 유효한 해라고 보장할 수 없다. p_{high} 를 포함하는 전체 경로가 여전히 제약조건을 만족할 수도 있지만 위배할 가능성도 있기 때문이다. 만약 위배할 경우에는 $\Theta(p_{high}) \oplus \Theta(p_{low})$ 를 분기 구간의 QoS 값으로 하여 분기제외 구간에 대한 최적 해를 탐색한다. 각 QoS 항목 값 중에 큰 값을 취하기 때문에 두 경로 모두 만족할 수 있게 된다. 이상의 설명을 바탕으로, 하나의 조건분기에 두 개의 분기 경로가 있을 때의 서비스 선택 절차를 다음과 같이 정리할 수 있다.

- Step 1: 전체경로에서 p_{high} 를 포함하는 경로 P 를 선택한다.
- Step 2: 경로 P 에 대하여 최적 해를 구한다. 만약 해의 탐색을 실패했을 경우에는 제약 값을 상향 조정하여 다시 탐색한다.
- Step 3: 구해진 최적 해에서 $\Theta(p_{high})$ 를 계산한다.
- Step 4: $\Theta(p_{high})$ 를 제약 조건 값으로 하여 경로 p_{low} 에 대한 최적 해를 구한다. 해가 구해지면 서비스 선택은 완료되고, 만약 해가 구해지지 않았을 경우에는 Step 5로 진행한다.
- Step 5: 전체 경로에서 p_{low} 만 포함하는 경로 P' 을 선택하고 최적 해를 탐색한다. 만약 해가 구해지지 않았을 경우에는 제약 조건을 재설정하여 해를 구한다.
- Step 6: 구해진 최적 해에서 $\Theta(p_{low})$ 를 계산한다.
- Step 7: $\Theta(p_{high}) < \Theta(p_{low})$ 가 참이면 서비스 선택은 완료된다. 그렇지 않으면 Step 8로 진행한다.
- Step 8: $W - (\Theta(p_{high}) \oplus \Theta(p_{low}))$ 을 제약조건으로 하여 분기제외 구간에 대한 최적 해를 구한다. 만약 최적 해를 구하지 못하면 전체 제약조건을 상향 조정해야 한다. □

Step 2에서 주어진 제약조건에서 해를 탐색하지 못했을 때에는 조건 분기 여부와 관계없이 전체 제약 조건을 만족할 수 있는 적절한 후보 서비스들이 없다는 뜻이므로 제약 조건 값의 재 설정이 필요한 것은 당연하다. Step 3에서 경로 p_{high} 에 선택된 서비스들의 QoS 항목별 합계를 계산한다. Step 4는 경로 p_{low} 만을 대상으로 최적 해를 구하되, 이 때 제약 조건은 Step 3에서 구한 $\Theta(p_{high})$ 를 이용한다. 만약 해가 구해진다면 서비스 선택의 최종 결과로서 p_{high} 를 포함하는 전체 경로에 대해 선택된 서비스 집합과 p_{low} 경로에 대해 선택된 서비스들로 결정되고, 실행 시점에 어느 분기로 진행되더라도 전체 QoS 제약 범위 이내의 실행이 가능하다. 물론 실제 실행에서 통신의 장애 등의 이유로 예정된 QoS를 위배할 수도 있지만 서비스 선택 단계에서는 최소한 QoS 제약을 위배하지 않는 실행 계획을 수립하는 것이 필수적이다.

한 가지 고려할 사항은 Step 4의 결과 해에 대한 전체 이득에 대한 해석이다. p_{high} 구간의 서비스들에 대한 실행 이득의 합이 p_{low} 구간의 서비스들에 대한 이득의 합보다 작다면 확률적으로 전체 실행 이득이 작은 방향으로 서비스들이 선택될 가능성이 있다. 그러나 실행 이득이란 하나의 서비스 타입에 속한 후보 서비스 가운데 하나를 선택할 때에는 QoS 값과 함께 중요한 기준이지만 조건 분기구간에 대해서는 우선 고려 대상일 수가 없다. 즉, 조건 분기는 실행시의 어떤 비교 값에 의하여 결정되므로, 이득의 합이 높다고 해서 실행시에 그 분기가 결정될 확률이 높은 것이 아니기 때문이다. 그렇기 때문에 본 연구에서는 조건 분기의 경로들에 대해서는 실행 이득의 기대 값이 아닌 실행 확률을 우선적으로 고려하여 서비스 선택을 하는 것이다.

만약 Step 4에서 p_{low} 구간만의 해가 구해지지 않을 경우에는 Step 5에서 p_{low} 를 포함하는 전체 경로에 대한 최적 해를 탐색함으로써 실제 실행에서 p_{low} 구간이 선택될 때 전체 실행 이득을 최대

화시킬 수 있다. 만약 제약조건인 QoS 항목 수가 하나 (즉, $K = 1$) 라면 Step 6-8의 과정은 불필요할 것이다. 왜냐하면 Step 4에서 $\Theta(p_{high})$ 를 이용한 p_{low} 의 해를 구하지 못했다는 사실은 Step 5에서 새로 구한 p_{low} 의 $\Theta(p_{low})$ 가 $\Theta(p_{high}) < \Theta(p_{low})$ 를 만족한다는 뜻이고, 따라서 p_{high} 구간의 해는 새로 구해진 분기제외 구간의 해와 결합될 때 전체 제약조건을 만족한다는 것을 의미하기 때문이다 (즉, $\Theta(p_{high}) + \Theta(\text{분기제외구간}) < W$). 그러나 QoS 항목 수가 두 개 이상이면 p_{high} 구간의 해가 전체 제약조건을 만족하지 못할 수도 있다. 어떤 QoS 항목 Q_i 에서는 $\Theta(p_{low})_i < \Theta(p_{high})_i$ 이고 다른 항목 Q_j 에서는 $\Theta(p_{low})_j > \Theta(p_{high})_j$ 일 수 있기 때문이다 ($i \neq j$).

Step 8에서는 각 QoS 항목 Q_k 별로 $\Theta(p_{high})_k$ 와 $\Theta(p_{low})_k$ 중 큰 값을 $\Theta(\text{분기구간})_k$ 의 값으로 설정하고, 전체 제약 조건 W_k 에서 $\Theta(\text{분기구간})_k$ 을 뺀 값을 제약조건으로 하여 분기제외 구간에 대한 해를 구한 것이다. Step 8까지 마친 결과는 전체 경로를 고려하여 p_{high} 구간, p_{low} 구간, 그리고 분기제외 구간에 대한 해를 차례로 구한 것이다. 전체 경로를 고려하지 않고 각 분기와 구간 별로 최적 해를 구하는 접근방법도 생각해 볼 수 있으나, 다음과 같은 문제점이 있다. 즉, 각 조건 분기 경로에 대한 최적 해를 구하기 위해서는 제약 조건 값이 필요한데, 전체 경로에 대한 조건을 대상으로 하면 분기제외 구간에 대한 해가 제약 조건을 위배할 가능성이 매우 높으며 그럴 경우 불필요하게 제약조건을 상향 조정해야 하는 일이 발생할 것이다. 그렇지 않고 적정 조건 값을 결정하기 위해서는 임의의 제약 조건 값으로 여러 번의 탐색을 해야 할 것이다. 제안된 방법에서는 많은 경우에 Step 4 또는 Step 7에서 서비스 선택이 완료될 가능성이 높다. 조건 분기의 각 경로의 서비스 타입의 수가 다르거나, 서비스 타입 별로 후보 서비스 수가 많을수록 $\Theta(p_{high})$ 를 만족하는 p_{low} 의 서비스 또는 $\Theta(p_{low})$ 를 만족하는 p_{high} 의 서비스를 선택할 수 있는 기회가 높기 때문이다.

[예제 1]

[그림 1]의 예제 데이터를 중심으로 위의 절차에 따라 서비스 선택을 실행한다. 하나의 조건 분기에 대한 경우이므로 [그림 1-a]에서 F_6 은 존재하지 않고 F_1 의 실행 확률이 F_5 보다 높다고 가정한다. 가용성은 (1-가용성)의 로그 값을 취하여 가격 및 응답시간과 동일한 방식으로 나타낼 수 있으나 [그림 1-b]의 수치를 그대로 보여주기 위해서는 배열 연산이 복잡해질 뿐 아니라 이 항목이 서비스 선택 절차에 영향을 주지 않기 때문에 앞으로의 예에서 가용성은 서비스 선택에 고려하지 않기로 한다.

Step 1, 2에서 F_{1234} 경로에 대한 최적 해를 구한다. 제약조건을 만족하며 이득이 가장 큰 최적 해는 $\{s_{11}, s_{21}, s_{31}, s_{42}\}$ 이고 $\Theta(p_{high}) = \Theta(F_1) = \langle 100, 50 \rangle$ 이다.

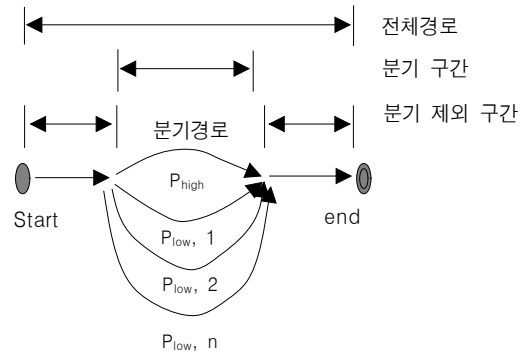
Step 4에서 $\Theta(p_{high})$, 즉, $\langle 100, 50 \rangle$ 를 제약 조건으로 하여 p_{low} , 즉, F_5 의 서비스를 선택하는데 유일한 서비스 s_{51} 의 QoS값 $\langle 200, 150 \rangle$ 은 제약 조건을 만족하지 못한다. 따라서 Step 5에서 경로 P' , 즉, F_{5234} 의 최적 해를 탐색하면 전체 제약 조건 $\langle 600, 250 \rangle$ 을 만족하는 해가 존재하지 않는다. 따라서 제약 조건의 상향 조정이 필요하고, 제약 조건을 $\langle 650, 340 \rangle$ 으로 조정하면 $\{s_{51}, s_{21}, s_{32}, s_{41}\}$ 해를 선택할 수 있다.

Step 6에서 F_5 에 대한 $\Theta(F_5)$ 는 $\langle 200, 150 \rangle$ 이고, 이 값은 Step 7의 $\Theta(p_{high}) < \Theta(p_{low})$, 즉, $\langle 100, 50 \rangle < \langle 200, 150 \rangle$ 를 만족하므로 서비스 선택이 완료된다.

최종 선택 결과는 $\{s_{11}, s_{21}, s_{32}, s_{41}, s_{51}\}$ 이고 제약 조건은 $\langle 650, 340 \rangle$ 으로 조정되었다. F_{1234} 경로를 위해서는 제약조건의 조정이 불필요하지만 F_{5234} 경로로 실행되기 위해서는 제약조건의 상향 조정이 불가피하다. □

3.3 여러 경로의 단일 조건 분기

3.2절에서 분기 경로 p_{low} 가 하나일 때의 서비



[그림 3] 하나의 조건분기에 여러 경로가 있는 복합 웹 서비스

스 선택 절차를 설명하였는데, 본 절에서는 아래 [그림 3]과 같이 p_{low} 가 $p_{low, 1}, p_{low, 2}, \dots, p_{low, n}$ 으로 여러 개 있는 경우에 대하여 선택 절차를 확장한다. 분기가 실행될 확률은 $p_{high} \geq p_{low, 1} \geq p_{low, 2} \geq \dots \geq p_{low, n}$ 이라고 가정한다. Step 3까지 절차는 3.2절과 동일하다.

- Step 1: 전체경로에서 p_{high} 를 포함하는 경로 P 를 선택한다.
- Step 2: 경로 P 에 대하여 최적 해를 구한다. 만약 해의 탐색을 실패했을 경우에는 제약 값을 조정하여 다시 탐색한다.
- Step 3: 구해진 최적 해에서 $\Theta(p_{high})$ 를 계산하고 이를 $\Theta(\text{분기구간})$ 으로 한다.
- Step 4: 각 경로 $p_{low, i}$ 에 대하여 다음을 실행한다 ($i = 1, \dots, n$).
 - Step 4-1: $\Theta(\text{분기구간})$ 을 제약 조건으로 하여 분기 경로 $p_{low, i}$ 에 대한 최적 해를 탐색한다. $p_{low, i}$ 의 해가 구해지면 $p_{low, i+1}$ 을 시작으로 하여 Step 4를 계속 진행한다. $p_{low, i}$ 의 해가 구해지지 않았을 경우에는 Step 4-2로 진행한다.
 - Step 4-2: 전체 경로에서 $p_{low, i}$ 만 포함하는 전체 경로 P' 을 선택하고 최적 해를 탐색한다. 만약 해가 구해지지 않았

을 경우에는 제약 조건을 재설정하여 해를 구한다.

Step 4-3: 구해진 해에서 $p_{low,i}$ 구간에서 선택된 서비스들에 대한 $\Theta(p_{low,i})$ 를 계산한다.

Step 4-4: $\Theta(\text{분기구간}) < \Theta(p_{low,i})$ 가 참이면 $\Theta(p_{low,i})$ 를 새로운 $\Theta(\text{분기구간})$ 로 지정하고 $p_{low,i+1}$ 을 시작으로 하여 Step 4를 계속 진행한다.

Step 4-5: $\Theta(\text{분기구간})$ 을 $\Theta(p_{low,1}) \oplus \dots \oplus \Theta(p_{low,i})$ 로 지정하고, $W - \Theta(\text{분기구간})$ 을 제약 조건 값으로 하여 분기제외 구간에 대한 최적 해를 탐색한다. 만약 최적 해를 구하지 못하면 전체 제약조건을 상향 조정 후 다시 분기제외 구간에 대한 해를 구한다.

Step 4-6: $p_{low,i+1}$ 을 시작으로 하여 Step 4를 계속 진행한다. □

Step 3에서 p_{low} 분기가 여러 개이므로 조건 분기구간을 대표하는 QoS 값이 여러 번 변경될 수 있기 때문에 분기구간에 대한 QoS 배열 $\Theta(\text{분기구간})$ 를 도입하였다. Step 4-1에서 $p_{low,i}$ 의 해가 구해진다는 것은 $\Theta(p_{low,i}) < \Theta(\text{분기구간})$ 이 참이며 실제 실행에서 $p_{low,i}$ 로 분기되었을 경우 전체 제약 조건을 만족할 수 있다는 것이다. 해를 구하지 못했을 때에는 $p_{low,i}$ 를 포함하는 전체 경로의 최적 해를 구하게 된다(Step 4-2).

$p_{low,i}$ 경로의 새로운 해에 대하여 $\Theta(p_{low,i})$ 의 각 QoS 값이 $\Theta(\text{분기구간})$ 의 해당 값보다 모두 크면 전 단계에서 구한 p_{high} 와 $p_{low,1}, \dots, p_{low,i-1}$ 구간의 서비스 해는 여전히 전체 제약 조건을 만족시키는 유효한 해이므로 다음 분기 경로로 계속 진행한다(Step 4-4). 그렇지 않을 경우 $p_{low,i}$ 이전에 구한 분기 경로들의 QoS 값이 새로 구해진 분기제외 구간의 해와 결합될 때, 전체 제약 조건을 여전히 만족할 수도 있지만 위배할 가능성도 있다. p_{high} 와 $p_{low,1}, \dots, p_{low,i-1}$ 각 구간의 서비스 해에 대하

여 전체 제약 구간의 위배 여부를 검사하여 위배하는 경로에 대해서만 Step 4-5의 방식을 적용하는 방법도 생각해 볼 수 있지만, 본 연구에서는 p_{high} 와 $p_{low,1}, \dots, p_{low,i}$ 모두를 대상으로 $\Theta(\)$ 를 구하고 각 QoS 합계의 항목별 최대값을 취하여 이를 $\Theta(\text{분기구간})$ 으로 새로 지정한다. 이렇게 함으로써 $p_{low,i+1}$ 부터 나머지 경로를 고려할 때 $\Theta(\text{분기구간})$ 에 대한 한번의 비교만으로 전체 제약조건 위배여부를 검사하여 비교 작업을 간단하고 효율적으로 할 수 있다.

최종적으로 전체 제약 조건에서 QoS 항목별로 $\Theta(\text{분기구간})$ 의 해당 값을 제한 결과를 제약조건으로 하여 분기제외 구간에 대한 해를 탐색한다. 이상의 절차에서 구해진 해는 분기 구간 p_{high} 와 $p_{low,1}, \dots, p_{low,n}$ 에 대한 서비스 집합과 분기제외 구간에 대한 서비스 집합의 합으로 구성되고, 전체 제약 조건을 만족할 뿐 아니라 실행 확률이 높은 순서부터 분기 구간의 한 경로를 포함하는 전체 경로가 최적화되거나 적어도 분기제외 구간에 대한 최적화를 얻게 된다.

이상의 절차를 통하여 $n+1$ 개의 조건 분기 경로들에 대하여 최선의 경우 단 한번의 전체 경로에 대한 최적해 탐색과 나머지 n 번의 분기구간에 대해서만 최적해 탐색을 하게 된다. 경로 길이가 길어질수록 최적해 탐색 시간은 기하급수적으로 증가하기 때문에 전체 경로에 대한 최적해 탐색 회수를 줄이는 것은 전체 웹 서비스 선택 시간을 크게 감소시키는 효과가 있다. 최악의 경우에는 각각의 p_{high} 와 각 $p_{low,i}$ 마다 전체 경로에 대하여 최적 해를 탐색해야 할 수도 있는데, 그렇다고 하더라도 이는 대부분 이전 연구에서 제시한 모든 가능한 전체 경로에 대한 최적 해를 탐색하는 것과 같은 것이다[16, 18]. 실행 확률이 높은 순서대로 고려하기 때문에 물론 그 경로에 대한 최적해가 우선적으로 고려되는 장점은 여전히 유지된다. 각 분기 구간들의 QoS 비교 검사 비용이 필요하지만 분기 수와 QoS 항목수가 경로 수보다는 대체로 작기 때문에 실행 시간에는 거의 영향을 주지

않을 정도로 생각된다.

[예제 2]

하나의 조건 분기에 세 개 이상의 분기 경로가 있어야 하므로, [그림 1-a]의 프로세스에서 F_6 가 V_5 에서 분기되고 실행 후 F_2 로 진행되는 것으로 가정한다. 분기의 실행 확률은 F_1, F_5, F_6 라고 가정한다.

분기 경로 F_5 의 해를 구하기까지는 예제 1과 동일하다. 즉, Step 1, 2에서 F_{1234} 경로에 대한 최적 해 $\{s_{11}, s_{21}, s_{31}, s_{42}\}$ 를 구하고 $\Theta(\text{분기구간}) = \langle 100, 50 \rangle$ 이다 Step 4에서 먼저 F_5 에 대하여 진행하면 Step 4-2에서 제약조건을 $\langle 650, 340 \rangle$ 으로 조정하여 F_{5234} 에 대한 $\{s_{51}, s_{21}, s_{32}, s_{41}\}$ 을 구할 수 있고 Step 4-4에서 $\Theta(F_5) = \Theta(\text{분기구간}) = \langle 200, 150 \rangle$ 이다.

다음으로 분기 경로 F_6 의 해를 구하기 위하여 Step 4-1로 진행하면, $\Theta(\text{분기구간}) = \langle 200, 150 \rangle$ 를 만족하는 경로 F_6 의 해는 $\{s_{61}\}$ 과 $\{s_{62}\}$ 가 있으나 $\{s_{61}\}$ 의 이득이 크기 때문에 $\{s_{61}\}$ 을 최적해로 선택한다. 모든 분기 경로에 대한 해를 탐색하였기 때문에 서비스 선택은 완료되고, 이 때 결과는 $\{s_{11}, s_{21}, s_{32}, s_{41}, s_{51}, s_{61}\}$ 이고 조정된 제약 조건은 $\langle 650, 340 \rangle$ 이다. □

3.4 여러 개의 조건분기

본 절에서는 아래 [그림 4]와 같이 복합 웹 서

비스에 조건 분기가 두 개 이상 순차적으로 포함되어 있는 경우를 고려한다. 이 때 분기 i 에서는 분기 경로 $p^{i_{high}}$ 의 실행확률이 $p^{i_{low}}$ 보다 높다고 가정한다 ($i = 1, \dots, m$). 편의상 [그림 4]를 기준으로 절차를 설명하고, 하나의 조건 분기에 3개 이상의 분기 경로가 있을 경우에는 3.3절의 방안을 적용하는 것으로 한다.

Step 1. 전체경로에서 $p^{1_{high}}, p^{2_{high}}, \dots, p^{m_{high}}$ 를 포함하는 경로 P 를 선택한다.

Step 2. 경로 P 에 대하여 최적 해를 구한다. 해의 탐색을 실패했을 경우에는 제약 값의 재설정이 필요하다.

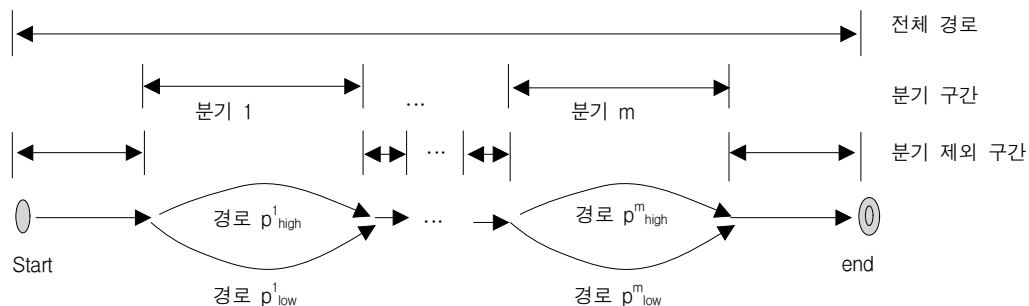
Step 3. 구해진 최적 해에서, 각 경로 $p^{i_{high}}$ 를 p^i 로 지정하고 p^i 에 대한 $\Theta(p^i)$ 를 계산한 후 이를 $\Theta(\text{분기}_i)$ 라고 각각 지정한다.

Step 4. 각각의 분기 i 에 대하여 다음을 실행한다 ($i = 1, \dots, m$).

Step 4-1. $\Theta(\text{분기}_i)$ 를 제약 조건으로 하여 경로 $p^{i_{low}}$ 에 대한 최적 해를 구한다. 해가 구해지면 Step 4에서 다음 분기 $i+1$ 에 대하여 절차를 계속하고, 모든 분기에 대하여 실행이 완료되면 서비스 선택은 완료된다. 만일 해가 구해지지 않으면 Step 4-2로 진행한다.

Step 4-2. 분기제외 경로와 $p^{i_{low}}$ 로 구성된 경로 P' 을 선택하고 최적 해를 탐색하되,

$$\text{제약조건은 } W - \sum_{j=1}^m \Theta(\text{분기}_j) + \Theta(\text{분기}_i)$$



[그림 4] 두 개 이상의 순차적 조건분기의 예

기_i)으로 한다. 해가 구해지지 않으면 전체 제약 조건을 재설정하여 해를 탐색한다.

Step 4-3. p_{low}^i 의 탐색된 서비스들에 대한 QoS 배열 $\Theta(p_{low}^i)$ 을 구한다.

Step 4-4. $\Theta(\text{분기}_i) < \Theta(p_{low}^i)$ 가 참이면, $\Theta(p_{low}^i)$ 를 새로운 $\Theta(\text{분기}_i)$ 로, p_{low}^i 를 새로운 p^i 로 각각 지정한 후 분기 $i+1$ 을 시작으로 하여 Step 4를 계속 진행한다.

Step 4-5. $\Theta(\text{분기}_i)$ 를 $\Theta(\text{분기}_i) \oplus \Theta(p_{low}^i)$ 로 지정하고 $W - \sum_{j=1}^m \Theta(\text{분기}_j)$ 를 새로운 제약 조건으로 하여 분기 제외 구간에 대한 최적 해를 탐색한다. 해가 구해지지 않을 경우에는 전체 제약 조건을 조정하여 새로운 제약 조건에서 분기 제외 구간에 대한 해를 구한다.

Step 4-6. 분기 $i+1$ 을 시작으로 Step 4를 계속한다. □

이상의 절차에서 보았듯이 조건 분기가 순차적으로 두 개 이상 포함되어 있을 경우에는 앞선 3.2절과 3.3절의 방안을 원칙적으로 적용하되 각각의 실행 조합에 대한 고려가 추가적으로 필요하다. Step 1, 2에서 각 조건 분기별로 실행 확률이 높은 분기 경로를 전체 경로에 포함시켜 최적 해를 먼저 구한다. Step 4는 각 조건 분기_i에 대하여 분기 경로 p_{low}^i 의 해를 구하기 위한 반복 구문이다. 각 조건 분기에서 p_{low}^i 가 두 개 이상일 때에는 3.3절에서와 유사한 과정이 Step 4에 중첩 반복 구문으로 적용되어야 할 것이다.

Step 4-1에서 $\Theta(\text{분기}_i)$ 를 제약조건으로 하여 p_{low}^i 구간의 최적해가 구해지지 않으면 p_{low}^i 를 포함하는 전체 경로에 대한 최적 해를 구하는데, 이때 고려되어야 하는 것은 분기_i 이외의 다른 분기에 대해서 어떤 경로를 포함하여 최적 해를 탐색

할 것이냐 하는 문제이다. 어떤 조건 분기_k($k < i$)에서는 p_{high}^k 대신 p_{low}^k 를 기준으로 하는 최적해가 탐색되었을 수도 있고 분기 k 를 대표하는 QoS는 $\Theta(\text{분기}_k) \oplus \Theta(p_{low}^k)$ 로 지정되었을 수도 있다. p_{low}^i 를 포함하는 최적 해를 구하기 위하여 p_{high}^k 또는 p_{low}^k 중 어느 하나를 포함하는 전체 경로를 대상으로 한다면, 새로 구해진 경로 p_{low}^i 의 $\Theta(p_{low}^i)$ 에 대하여 분기_k의 다른 경로의 해가 만족되지 않을 수가 있게 된다. 즉, 조건분기_i의 경로의 해를 구하기 위하여 분기_k의 어느 한 경로만 포함해서는 분기_k의 나머지 경로를 동시에 항상 만족시킬 수는 없다.

따라서 본 연구에서는 각 분기_i에서 최종적으로 계산된 $\Theta(\text{분기}_i)$ 를 그 분기를 대표하는 QoS로 하고(Step 3, Step 4-4, Step 4-5), 임의의 p_{low}^i 경로를 포함하는 해를 전체 제약조건을 대상으로 구하고자 할 때, 분기_i를 제외한 나머지 분기의 해가 결정된 것으로 간주하여, 전체 제약 (W)에서 $\Theta(\text{분기}_1), \dots, \Theta(\text{분기}_{i-1}), \Theta(\text{분기}_{i+1}), \dots, \Theta(\text{분기}_n)$ 을 제한 것을 제약 조건으로 하여 분기제외 구간과 경로 p_{low}^i 에 대한 해를 구하도록 한다(Step 4-2). 이를 통하여 이전에 결정한 각 분기 경로의 해를 최대한 유지하도록 하여 다른 조건 분기의 경로에 대한 최적 해를 재탐색할 때 이용하도록 하였다. Step 4-5에서는 분기 제외 구간에 대한 해를 구한다. 만약 한 조건 분기_i에 분기수가 여러 개 있을 경우에는 p^i 와 $\Theta(\text{분기}_i)$ 는 여러 번 재지정될 수 있다.

이상의 절차에 따라 서비스 선택을 하게 되면 m 개의 조건 분기와 n_1, \dots, n_m 만큼의 각 분기 수에 대하여, 최선의 경우 각 p_{high}^i 를 포함하는 전체 경로에 대한 한번의 최적해 탐색과 $\sum_{i=1}^m (n_i - 1)$ 만큼의 부분 최적화 탐색이 필요하다. 이 때 부분 최적화 탐색의 대상은 각 조건 분기의 p_{low}^i 경로들이다. 최악의 경우, 각 p_{low}^i 분기마다 추가적으로 p_{low}^i 와 분기제외 구간이 결합된 경로에 대한

최적 해를 구해야 하므로 $\sum_{i=1}^m (n_i - 1)$ 만큼의 부분 경로에 대한 최적해 탐색이 추가적으로 필요할 것이다. 그러나 이런 경우에도, 이전의 연구에서 요구되었던 모든 가능한 경로의 조합에 대한 전체 경로에 대한 탐색 수 $\prod_{i=1}^m n_i$ 보다는 월등히 작은 수치일 뿐 아니라 최적해 탐색 경로의 길이도 부분 경로로서 짧기 때문에 서비스 선택 해의 탐색 시간이 매우 단축될 수 있을 것이다.

[예제 3]

본 절의 선택 절차에 대한 예제는 [그림 1]의 예제 데이터를 그대로 이용할 수 있다. 두 개의 조건 분기가 있고 각각 두 개의 분기 경로가 있다. 분기 경로의 실행 확률은 앞선 예제에서와 마찬가지로 F_1 의 실행 확률이 F_5 보다 높고, F_3 의 실행확률이 F_6 보다 높다고 가정한다.

Step 1, Step 2에서 F_{1234} 경로에 대한 최적 해 $\{s_{11}, s_{21}, s_{31}, s_{42}\}$ 를 구한다.

Step 3에서 $p^1 = F_1$ 이고 $\Theta(p^1) = \Theta(\text{분기}_1) = \langle 100, 50 \rangle$ 이며, $p^2 = F_{34}$ 이고 $\Theta(p^2) = \Theta(\text{분기}_2) = \langle 290, 140 \rangle$ 이다. 여기서 $\Theta(p^2)$ 의 값은 s_{31} 의 QoS 배열 $\langle 150, 100 \rangle$ 과 s_{42} 의 QoS 배열 $\langle 140, 40 \rangle$ 를 항목별로 합산한 값이다.

Step 4에서 분기₁의 F_5 경로에 대하여 진행한다. $\Theta(p^1) = \langle 100, 50 \rangle$ 에 대하여 해를 구하지 못하였으므로 Step 4-2에서 분기 제외 경로와 F_5 , 즉, F_{52} 에 대한 최적 해를 탐색하되 제약조건은 $\langle 600, 250 \rangle - \langle 290, 140 \rangle$, 즉, $\langle 310, 110 \rangle$ 으로 한다. 경로 F_{52} 에 세 가지 가능한 선택이 있으나 모두 제약조건을 만족하지 못한다. 실행 시간과 가격 제약 값을 각각 90만큼 증가시키면 $\langle 400, 200 \rangle$ 을 제약조건으로 하는 해 $\{s_{51}, s_{21}\}$ 를 얻을 수 있다.

Step 4-3에서 $\Theta(p^1_{low}) = \Theta(F_5) = \langle 200, 150 \rangle$ 이고, Step 4-4에서 $\Theta(\text{분기}_1) < \Theta(p^1_{low})$, 즉, $\langle 100,$

$50 \rangle < \langle 200, 150 \rangle$ 가 만족되므로 $\langle 200, 150 \rangle$ 을 $\Theta(\text{분기}_1)$ 의 값으로 새로 지정하고 분기₂에 대하여 Step 4를 계속 진행한다.

분기₂의 p^2_{low} , 즉, F_6 에 대하여 $\Theta(\text{분기}_2) = \langle 290, 140 \rangle$ 을 제약 조건으로 하여 해를 찾아보면 $\{s_{61}\}$ 과 $\{s_{62}\}$ 두 개의 해가 가능한데 $\{s_{61}\}$ 의 이득이 크기 때문에 $\{s_{61}\}$ 을 선택한다. 모든 분기의 각 경로에 대한 해를 결정하였으므로 서비스 선택이 완료되며, 결과 해는 $\{s_{11}, s_{21}, s_{31}, s_{42}, s_{51}, s_{61}\}$ 로 구성되고 제약조건은 90씩 증가한 $\langle 690, 340 \rangle$ 으로 조정되었다. □

5.5 중첩된 조건분기

이번 절에서는 아래 [그림 5]와 같이 복합 웹 서비스에 조건 분기가 중첩되어 있는 경우에 대한 서비스 선택 절차를 설명한다. 그림에서 분기 경로 p^{out}_{high} 와 p^{in}_{high} 가 p^{out}_{low} 와 p^{in}_{low} 보다 각각 실행 확률이 높은 것으로 가정한다. 설명을 간단하게 하기 위하여 그림과 같이 하나의 조건분기가 p^{out}_{high} 경로에 중첩되고 각 분기의 수가 2개씩 있는 것으로 한다.

Step 1. 외부 분기의 경로 p^{out}_{high} 와 내부 분기의 경로 p^{in}_{high} 를 포함하는 전체경로 P 를 선택한다.

Step 2. 경로 P 에 대하여 최적 해를 구한다. 만약 해의 탐색을 실패했을 경우에는 제약 값의 재설정이 필요하다.

Step 3. 구해진 최적 해에서 $\Theta(p^{out}_{high})$ 를 계산하고 이를 $\Theta(\text{분기}_{out})$ 으로 지정한다.

Step 3-1. $\Theta(p^{in}_{high})$ 를 구하고 이를 $\Theta(\text{분기}_{in})$ 으로 지정한다

Step 3-2. 내부 분기 경로 p^{in}_{low} 에 대하여 $\Theta(\text{분기}_{in})$ 을 제약조건으로 하는 최적 해를 탐색한다. 해가 구해질 경우에는 Step 4로 진행한다.

Step 3-3. 전체 경로 중, 분기 제외 구간과 p^{in}_{low} 경로만 포함하는 부분 경로에 대

하여 최적 해를 탐색하되 제약 조건은 $W - \Theta(\text{분기}_{out})$ 으로 한다. 해가 구해지지 않으면 전체 제약조건의 재설정이 필요하다.

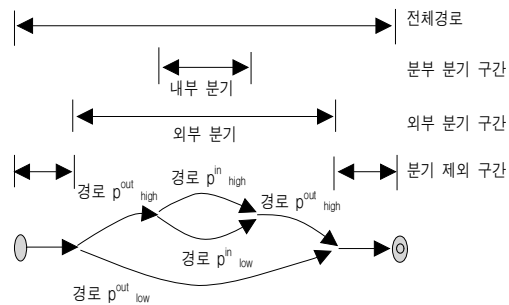
- Step 3-4. 구해진 해에서 p^{in}_{low} 에 포함된 서비스들에 대한 $\Theta(p^{in}_{low})$ 를 계산한다.
- Step 3-5. $\Theta(\text{분기}_{in}) < \Theta(p^{in}_{low})$ 가 참이면 $\Theta(p^{in}_{low})$ 를 새로운 $\Theta(\text{분기}_{in})$ 으로 지정하고 Step 4로 진행한다.
- Step 3-6. $\Theta(\text{분기}_{in})$ 는 $\Theta(\text{분기}_{in}) \oplus \Theta(p^{in}_{low})$ 으로 지정하고 분기 제외 구간에 대한 최적 해를 구한다. 이 때 제약 조건은 $W - \Theta(\text{분기}_{in}) - \Theta(\text{분기}_{out})$ 으로 한다. 해를 구하지 못하면 전체 제약 조건을 상향 조정하여 다시 해를 탐색한다.
- Step 4. $\Theta(\text{분기}_{out})$ 은 $\Theta(\text{분기}_{out}) + \Theta(\text{분기}_{in})$ 로 지정하고 이를 제약 조건으로 하여 경로 p^{out}_{low} 에 대한 최적 해를 탐색한다. 해가 구해지면 서비스 선택이 완료된다.
- Step 5. 외부 분기 경로 p^{out}_{low} 를 포함하는 전체 경로를 P' 라고 하고, P' 에 대한 최적 해를 탐색한다. 해가 구해지지 않으면 전체 제약조건 값의 재설정이 필요하다.
- Step 6. $\Theta(\text{분기}_{out}) < \Theta(p^{out}_{low})$ 가 참이면 서비스 선택이 완료된다.
- Step 7. 분기 제외 구간에 대한 최적 해를 탐색하되, 제약 조건은 $W - (\Theta(\text{분기}_{out}) \oplus \Theta(p^{out}_{low}))$ 로 한다. 만약 최적 해를 구하지 못하면 전체 제약조건을 조정 한 후 다시 분기 제외 구간에 대한 해를 구한다. □

조건 분기가 중첩된 서비스 선택 절차의 중요한 차이점은 중첩된 내부 분기에서 해를 찾고자 할 때이다. Step 3-2에서 $\Theta(\text{분기}_{in})$ 을 제약조건으로 하여 p^{in}_{low} 경로에 대한 해를 구하지 못하면 전체 경로를 고려한 해를 구해줘만 한다. 이 때 내부 분기를 포함하는 경로 p^{out}_{high} 를 제외하고 분기 제

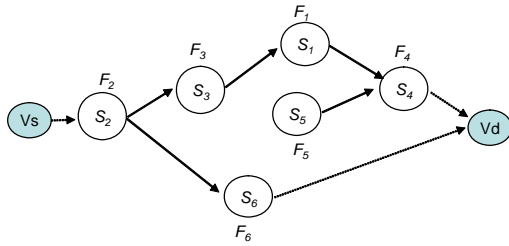
외 구간과 p^{in}_{low} 경로에 대해서만 새로운 해를 구하도록 하였다(Step 3-3). 이는 여러 단계로 중첩될 경우와 중첩된 내부 분기의 경로수가 많을 경우 탐색횟수가 많아지는 것을 제한하고 또한 최적화 대상 경로가 길어짐으로 인한 탐색시간의 급격한 증가를 방지하기 위한 것이다. 또한 p^{out}_{high} 에 대하여 이전 단계에서 구한 해는 어느 정도 최적 해에 가까운 해이기 때문에 이를 유지하는 것이 실행 이득 측면에서도 나쁘지 않을 것이다.

Step 3-4부터 Step 3-6의 과정은 3.4절에서의 절차가 유사하게 적용된 것이다. 내부 분기 내에 p^{in}_{low} 경로가 두 개 이상 있는 경우에는 Step 3-2부터 Step 3-6이 각 내부 경로 수만큼 반복되도록 하면 된다. Step 4부터는 그림에서 p^{out}_{low} 에 대한 해를 구하기 위한 절차이다. p^{out}_{low} 에 대응되는 분기 경로는 p^{out}_{high} 와 내부분기 부분이므로 $\Theta(\text{분기}_{out}) + \Theta(\text{분기}_{out})$ 를 제약 조건으로 하여 p^{out}_{low} 경로의 해를 탐색하고, 나머지 절차는 이전 절의 절차와 유사하게 진행되었다.

[그림 5]와 달리 만약 내부 분기가 p^{out}_{low} 에 포함되어 있다면, 위의 절차의 순서가 다소 조정되어야 하지만 기본 원리는 동일하다. Step 3까지의 절차에 의하여 외부 분기 p^{out}_{high} 의 서비스와 $\Theta(\text{분기}_{out})$ 이 구해질 것이고, $\Theta(\text{분기}_{out})$ 을 제약 조건으로 하여 p^{out}_{low} 와 p^{in}_{high} 에 대한 해를 탐색하는 것으로 진행된다. p^{in}_{high} 에 대한 해가 결정되면 $\Theta(p^{in}_{high})$ 를 제약조건으로 하여 p^{in}_{low} 에 대한 해를 구한다. 여러 개의 조건분기가 중첩되었을 경우에



[그림 5] 중첩된 조건 분기



[그림 6] 중첩 분기의 예제 프로세스

는 Step 1에서 각 내부 분기에서 확률이 높은 분기를 우선 선택하고 중첩의 수만큼 Step 3 블록의 과정이 되풀이되어야 할 것이다. 중첩된 조건 분기가 순차적으로 여러 개 있을 경우에는 본 절과 3.3절의 과정을 적용하여 설명할 수 있다.

[예제 4]

분기가 중첩된 예제에 대한 위의 절차를 살펴보기 위하여 [그림 1-a]의 프로세스를 [그림 6]과 같이 조정한다. 즉, [그림 1-a]의 분기이 F_3 과 F_4 사이에 중첩시키고 예제 데이터는 그대로 이용한다. 분기 경로의 실행 확률은 앞선 예제에서와 마찬가지로 F_3 의 실행확률이 F_6 보다 높고, F_1 의 실행 확률이 F_5 보다 높다고 가정한다.

Step 1, 2에서 경로 F_{2314} 를 선택하고 최적해 $\{S_{21}, S_{31}, S_{11}, S_{42}\}$ 를 구한다.

Step 3에서 $\Theta(p_{high}^{out}) = \Theta(\text{분기}_{out}) = \Theta(F_{34}) = \langle 290, 140 \rangle$ 을 계산한다.

Step 3-1에서 $\Theta(p_{high}^{in}) = \Theta(\text{분기}_{in}) = \Theta(F_1) = \langle 100, 50 \rangle$ 으로 지정한다.

Step 3-2에서 내부 분기 경로 p_{low}^{in} , 즉, F_5 에 대하여 $\langle 100, 50 \rangle$ 을 제약조건으로 해를 구하지 못하므로, Step 3-3에서 분기제외구간인 F_2 와 F_5 만 포함된 F_{25} 에 대하여 제약 조건은 $W - \Theta(\text{분기}_{out}) = \langle 600, 250 \rangle - \langle 290, 140 \rangle = \langle 310, 110 \rangle$ 으로 하여 해를 검색한다. 세 가지 가능한 서비스 조합이 모두 해를 만족시키지 못하므로 제약조건을 조정한다. 전체 제약 조건 W 를 $\langle 690, 340 \rangle$ 으로 조정하면 $W - \Theta(\text{분기}_{out}) = \langle 400, 200 \rangle$ 에 대

하여 해 $\{S_{51}, S_{21}\}$ 를 구할 수 있다.

Step 3-4에서 $\Theta(p_{low}^{in}) = \Theta(F_5) = \langle 200, 150 \rangle$ 을 구하고, Step 3-5에서 $\Theta(\text{분기}_{in}) < \Theta(p_{low}^{in})$, 즉, $\langle 100, 50 \rangle < \langle 200, 150 \rangle$ 이 만족하므로 $\langle 200, 150 \rangle$ 를 $\Theta(\text{분기}_{in})$ 의 새로운 값으로 지정하고 Step 4로 진행한다.

Step 4에서 $\Theta(\text{분기}_{out})$ 는 $\Theta(\text{분기}_{out}) + \Theta(\text{분기}_{in}) = \langle 290, 140 \rangle + \langle 200, 150 \rangle = \langle 490, 290 \rangle$ 으로 지정하고, 이를 제약조건으로 하여 p_{low}^{out} , 즉, F_6 의 해를 탐색하면, $\{S_{61}\}$ 과 $\{S_{62}\}$ 가운데 이득이 큰 $\{S_{61}\}$ 을 선택할 수 있다.

이로써 최종해는 $\{S_{11}, S_{21}, S_{31}, S_{42}, S_{51}, S_{61}\}$ 이고 제약 조건은 $\langle 690, 340 \rangle$ 이다. 앞선 예제 3과 동일한 결과가 구해졌지만 서비스 선택 절차를 예로써 보여주기에 적합하다고 생각된다. □

4. 실험

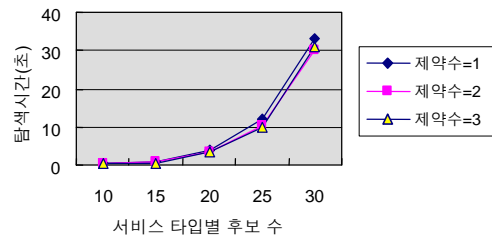
4.1 실험 환경

본 연구에서 제안한 서비스 선택 절차에 대해서는 예를 통하여 살펴보았다. 본 장에서는 순차적 경로에 대하여 서비스 타입의 개수와 서비스 타입별 후보 서비스 수가 최적해 탐색에 미치는 영향을 살펴보고자 한다. 제 3장의 서비스 선택 절차를 임의의 형태의 조건분기 구성에 대해 적용할 수 있는 일반화된 알고리즘을 개발하고 그 성능을 분석하는 일은 향후 과제로 미루기로 한다. 본 연구에서는 전체 경로에 대한 최적화 탐색을 가능한 줄이고 부분 경로, 즉, 전체 경로보다 길이가 짧은 경로들에 대한 해를 구하는 방향으로 접근하였는데, 본 실험에서 경로 길이와 후보 서비스의 수가 탐색 시간에 미치는 영향을 알아보는 것이 우선적으로 의미가 있다고 생각하기 때문이다. 또한 제약 조건의 수가 최적해 탐색 시간에 미치는 영향도 함께 살펴보았다. 따라서 실험 데이터는 단순한 순차적 프로세스를 모형으로 하고 문제의 크기, 즉, 복잡도는 서비스 타입의 개수, 각 서비스

타입에 속하는 후보 서비스의 수, QoS 제약 항목의 수를 이용하여 조절하였다.

최적 해를 구하는 프로그램으로 OPBDP를 이용하였다[2]. OPBDP는 Davis-Putnam 최적해 기법에 기초하고 있으며 정수 계수의 목표함수와 주어진 제약조건 부등식에 대하여 이진 결정변수의 해를 구하는 프로그램이다. OPBDP는 Pentium 4 PC(Windows XP Professional) 환경에서 실행되었다. 최적 해의 탐색 시간은 실험 컴퓨터의 성능과 탐색 프로그램에 따라 많은 차이가 있지만, 본 실험의 목적은 짧은 탐색시간을 구하는 것이 아니라 서비스 수와 프로세스 경로의 길이가 해의 탐색에 미치는 정도를 살펴보는 것이기에 때문에 실험의 전산환경은 측정값을 이해하는데 참고하면 충분할 것으로 생각한다.

각 서비스 타입에서 선택될 후보 서비스의 개수는 10~30 범위에서 5씩 증가시키며 실험하였다. 더 이상의 후보 서비스의 수에서는 실행시간이 급격히 증가하여 탐색 시간을 측정하는데 지나치게 오랜 시간이 소요될 뿐 아니라 측정 시간 자체의 의미가 없을 정도여서 적정 수준의 개수로 제한하였다. 경로의 길이, 즉, 서비스 타입의 개수도 4~7까지 1씩 증가시키며 측정이 용이한 범위로 제한하였다. 제약조건 식은 3개를 생성하여 한 개, 두 개, 세 개일 때 분리하여 탐색 프로그램을 각각 실행하였다. 실험에서 각 후보 서비스의 이득 값은 10~30의 임의 값으로 생성되도록 하였다. 각 후보 서비스에 대한 QoS 값은 5~15사이의 임의의 값으로 생성하였다. 세 개의 QoS 항목에 대한 전체 제약 조건 값을 각각 다르게 설정할 경우 탐색시간에 불규칙적인 영향을 미칠 수 있기 때문에, 전체 제약조건 값은(10×경로길이) 값으로 동일하게 지정하였다. 여기서 10은 5~15의 중간 값을 택한 것이다. 경로 길이와 후보 서비스 수가 주어졌을 때, 문제 인스턴스를 2개 생성하여 프로그램을 각각 실행한 후, 실행 시간의 평균값을 취하였다.

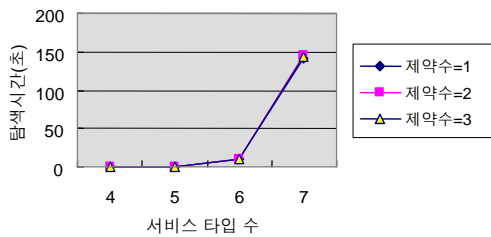


[그림 7] 후보 수의 변화에 대한 최적해 탐색 시간(서비스타입 수 = 5)

4.2 실험 결과

[그림 7]은 서비스 타입의 수가 5일 때, 후보 서비스 수를 10부터 30까지 5씩 증가시키며 최적해 탐색 시간을 측정하였다. 그림에서 보는 바와 같이 후보 수가 15일 때에는 1초 내외에 해를 찾았고, 20일 때에 3~4초, 25일 때 10~12초, 30일 때에 30여 초로 가파르게 상승하였다. 이 경우, 후보 수가 10개 증가할 때 대략 10배씩 탐색 시간이 증가하는 수준으로 보인다. 서비스 타입의 수가 많아지면 그 증가 속도는 더욱 빠를 것이다. 제약 조건의 수는 해의 탐색 시간에 크게 영향을 미치지 않았다. 대체로 제약수가 많아질수록 탐색시간이 약간 줄어드는 경향이 많았으나 모든 경우에 그렇지는 않았다.

[그림 8]은 후보 서비스 수를 15로 고정하고 서비스 타입의 수를 증가시키며 해의 탐색 시간을 측정하였다. 서비스 타입 수가 6일 때까지는 10초 이내로 천천히 증가하였으나 서비스 타입 수가 6에서 7로 하나 증가함에 따라 탐색시간이 142초 정도로 15배 이상 급격히 증가하였다. [그림 6]과 비교해 볼 때 서비스 타입의 개수, 즉, 탐색 대상 경로의 길이가 후보의 수보다 해의 탐색 시간에 훨씬 결정적인 영향을 미친다는 것을 알 수 있다. 서비스 타입의 수가 6이고 후보의 수가 20일 때 해의 탐색시간은 53초 정도였는데, 후보 수가 15에서 20으로 증가한 비율(33%)에 대한 탐색 시간의 증가율(5.5배)을 서비스 타입 수가 6에서 7로 증가한 비율(대략 17%)에 대한 탐색 시간



[그림 8] 서비스 타입 수의 변화에 대한 최적해 탐색 시간 (후보 수=15)

의 증가율 (15배)을 비교해 봐도 그 차이를 알 수 있다.

[그림 8]에서도 제약수의 변화에 따른 실행시간의 차이는 거의 미약했는데, 한가지 흥미로운 사실은 각 서비스 타입 수에 대한 탐색시간의 차이가 [그림 7]의 실험에서보다 더 작았다는 것이다. 즉, [그림 7]의 관측에서는 후보 수가 15일 때 0.6~0.8초, 20일 때 3.3~3.9, 25일 때 9.7~12초, 30일 때 30~33초의 범위에서 차이가 있는 반면, [그림 8]의 실험에서는 서비스 타입의 수가 5일 때 0.6~0.8초, 6일 때 9.1~9.7초, 그리고 7일 때 142~145초의 범위로 그 차이가 작게 관찰되었다. 그 이유는 [그림 7]의 경우 후보 수가 늘어남에 따라 후보들의 QoS 값이 다양해지는 반면, [그림 8]의 경우에는 후보 수가 동일하기 때문에 제약 조건에 미치는 변동의 폭이 작기 때문인 것으로 여겨진다.

5. 결 론

본 논문에서는 조건 분기를 포함하는 복합 웹 서비스의 서비스 선택 방안을 제안하였다. 모든 실행 가능한 분기 경로에 대하여 최적 해를 탐색하는 연구가 많이 있었지만 전체 QoS 제약조건을 만족시키는 방안에 대한 연구는 부족하였다. 본 연구에서 제안하는 서비스 선택의 기본 아이디어는 확률적으로 실행 가능성이 높은 조건분기의 경로에 대한 최적 해를 우선적으로 탐색하되, 전체 경로를 대상으로 하는 해의 탐색 횟수를 가능

한 줄이며 실행 확률이 낮은 조건분기 경로들에 대해서도 제약조건을 만족하는 해를 생성하는 것이다. 하나의 조건분기에 여러 경로가 있는 형태, 여러 조건 분기가 순차적으로 나열된 형태, 그리고 조건 분기가 중첩된 기본적인 형태들에 대하여 서비스 선택 절차를 소개하였고, 예를 통하여 설명하였다. 실험을 통하여 후보 서비스의 수보다 최적해 탐색 경로, 즉, 서비스 타입의 수가 탐색시간에 보다 크게 영향을 미치는 것을 관찰하였다.

복합 웹 서비스에 대한 서비스 선택은 간단한 문제가 아니며 많은 추가 연구를 필요로 한다. 본 논문에서는 기본적인 형태의 조건 분기에 대한 서비스 선택 절차를 소개하였는데, 이들 기본 형태가 복합적으로 혼합된 서비스 프로세스에 대한 통합되고 일반화된 알고리즘의 연구가 필요하다. 또한 최적해 탐색 시간을 줄일 수 있는 방안과 제약 조건을 가장 적절하게 조정할 수 있는 방안에 대한 연구도 필요할 것으로 생각된다.

참 고 문 헌

- [1] 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.
- [2] Barth, P., "Davis-Putnam Based Enumeration Algorithm for Linear Pseudo-Boolean Optimization", MPI-I-95-2-003, MPI Informatik, (<http://www.mpi-inf.mpg.de/departments/d2/software/opbdp>). (1995).
- [3] Benatallah, B., *et al*, "On automating Web Services discovery", VLDB Journal, Vol. 14(2005).
- [4] Canfora, G., M. D. Penta, R. Esposito, and L. V., Maria, "An Approach for QoS-aware Service Composition based

- on Genetic Algorithm”, Proceedings of the International Conference on Genetic and Evolutionary Computation (2005).
- [5] Erl, T., *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services*, Prentice Hall, 2004.
- [6] Fan, J. and Kambhampati, S., “A Snapshot of Public Web Services”, ACM SIGMOD Record, Vol.34, No.1(2005).
- [7] Freitas, A. A., “A Critical Review of Multi-Objective Optimization in Data Mining : a position paper”, ACM SIGKDD Explorations Newsletter, Vol.6, No.2(2004).
- [8] Juric, M. B., B. Mathew, and P. Sarang, *Business Process Execution Language for Web Services*, Packt Publishing, 2004.
- [9] Kerrigan, M., “Web Service Selection Mechanisms in the Web Service Execution Environment”, Proceedings of ACM SAC '06 Conference (2006).
- [10] Kuster, U. *et al*, “DIANE - An Integrated Approach to Automated Service Discovery, Matchmaking and Composition”, Proceedings of WWW2007 Conference (2007).
- [11] Manikrao, U. S. and T. V. Prabhakar, “Dynamic Selection of Web Services with Recommendation System”, Proceedings of IEEE International Conference on Next Generation Web Services Practices (2005).
- [12] Maximilien, E. M. and M. P. Singh, “A Framework and Ontology for Dynamic Web Service Selection”, IEEE Internet Computing, September–October(2004).
- [13] Pisinger, D., “A Minimal algorithm for the Multiple-choice Knapsack Problem,” *European Journal of Operational Research*, Vol.83, (1995).
- [14] Ran, S., “A Model for Web Services Discovery with QoS”, ACM SIGecom Exchange, Vol.4, No.1(2003).
- [15] Weerawarana, S., *et al*, *Web Services Platform Architecture*, Prentice Hall, 2005.
- [16] 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)
- [17] Yu, T., Yue Zhang, and K.-J. Lin, “Efficient Algorithms for Web Services Selection with End-to-End QoS Constraints”, *ACM Transactions on the Web*, Vol.1, No.1(2007).
- [18] Zeng, L., *et al*, “QoS-Aware Middleware for Web Services Composition”, *IEEE Transactions on Software Engineering*, Vol.30, No.5(2004).

◆ 저 자 소 개 ◆

**서 상 구 (skseo@kw.ac.kr)**

서울대학교에서 컴퓨터공학과 학사, 한국과학기술원(KAIST) 전산학과에서 전산학 석사 및 박사학위를 취득하였다. 현대전자 (현, 하이닉스), 현대정보기술에서 소프트웨어 개발 및 기술지원본부에서 근무하였고, 현재 광운대학교 경영대학 경영정보학과 교수로 재직 중이다. 연구분야는 데이터베이스 최적화, 웹 데이터베이스, 웹 서비스, 데이터웨어하우스 등이다.

