

# 지능형 소프트웨어 에이전트에 기반한 피크 기간에서의 여행 계획 추천 시스템

임홍순\* · 안형준\*\* · 김종우\*\*\* · 박성주\*\*

## Intelligent Agent-based Travel Planning Recommendation System in Peak Seasons

Hong Soon Yim\* · Hyung Jun Ahn\*\* · Jong Woo Kim\*\*\* · Sung Joo Park\*\*

### ■ Abstract ■

This paper presents a multi-agent system for intelligent recommendation of travel plans to users. The goal of the system is to provide alternative and preferable travel plans to users when the availability of tickets is low such as in vacations, holidays, weekends, or peak seasons. The multiple agents in the system search for available alternatives for a target travel in collaboration with other agents and recommend best alternatives by analyzing them using a multi-criteria decision-making model. A prototype online travel support system was constructed and a simulation experiment was performed for evaluation and comparison with different travel planning strategies.

Keyword : Multi-Agent System, Multi-Criteria Decision-Making(MCDM), Travel Planning

## 1. 서 론

여행 산업에서의 e-비즈니스(e-business)의 발  
달은 인터넷을 통해 광범위한 여행 정보를 얻을 수

있도록 만들고 있다. 인터넷을 통해 여행자들은 여  
행 수단에 대한 시간표, 경로, 예약 가능한 좌석,  
숙박, 렌터카, 식당 등의 여행 계획을 위한 정보를  
얻을 수 있다[20]. 또한, 지능형 소프트웨어 에이전

논문접수일 : 2004년 7월 30일      논문게재확정일 : 2004년 10월 15일

\* 핸디소프트

\*\* 한국과학기술원 테크노경영대학원

\*\*\* 한양대학교 경영대학 경영학부(교신저자)

트 기술은 단순한 이메일 필터링에서 항공 통제 시스템과 같은 복잡한 시스템까지 적용 범위가 확장되고 있다[15]. 에이전트 기술을 여행 산업에 적용하는 연구들도 자연스럽게 진행되고 있으며, 이러한 연구들을 통해 다양한 연구 목적의 시범(prototype) 시스템들이 개발되었다[7, 12, 20, 21, 22, 24]. 이러한 연구 목적의 시범 시스템들 외에도 익스피디아(expedia.com), 트래블로시티(travelocity.com) 등과 같은 상용 목적의 여행 지원 서비스 사이트들도 존재한다[22].

현재 에이전트에 기반 한 여행 계획 지원 시스템들에서, 에이전트들은 여행자의 요구사항과 제약 조건을 만족시키는 대안들을 검색하여 제공하는 수준에 머무르고 있다. 하지만, 방학, 휴가, 주말 등과 같은 여행 수요가 정점에 다다른 기간에는 이를 만족시키는 대안이 존재하지 않는 경우가 발생한다. 여행자가 여행 계획을 하는 시점에 원하는 여행 계획을 얻지 못하는 경우, 여행자는 자신이 원했던 계획은 아니지만 에이전트가 제시해 준 차선의 대안들 중에서 선택하거나, 여행자가 원했던 계획이 타인의 취소에 의해서 가용 하게 될 때까지 기다리는 수 밖에 없다. 그러나 원하는 여행 계획을 기다리는 것만으로는 그 여행 계획을 얻을 수 있다는 것을 보장할 수 없기 때문에, 에이전트가 제시해준 대안이 만족스럽지 않다면 여행자는 자신이 스스로 다른 대안을 찾아야 한다. 하지만, 이것은 검색 시간의 소비, 동일 사이트에 대한 반복적인 방문, 대안 경로에 대한 검색 등의 추가적인 노력을 요구한다.

본 논문에서는 에이전트 기술에 기반 한 여행 계획 추천 시스템을 제시한다. 기존 예약의 취소 또는 임시 운행 수단의 추가 등으로 인해 가능한 여행 계획이 계속적으로 변화하는 환경에서, 에이전트 기반 여행 지원 시스템의 성공여부는 여행자의 간섭 없이 에이전트가 지능성과 자율성[15]을 통해 얼마나 여행자를 만족시키는가의 의해서 결정된다. 본 논문에서 제시된 시스템에서 여행 계획의 추천은 계속해서 임시 계획을 생성하고, 해당 임시 계획이 최선의 대안이 될 때, 해당 계획을 여

행자에게 제공한다. 임시 계획으로 결정된 여행 계획은 현재 계획보다 더 나은 새로운 임시 계획이 발견될 때까지 예약을 통해 보호된다. 만약 이 임시 계획을 보호하지 않는다면 해당 임시 계획이 다시 유효하지 않게 될 수도 있기 때문이다. 임시 계획은 완화된 제약 조건들과 해당 임시 계획을 발견할 때까지 소요된 비용들을 나타내는 속성들로 구성되며, 이러한 속성들은 최선의 대안을 선택하기 위한 판단 근거가 된다. 따라서, 임시 계획을 만드는 문제를 다중 기준 의사 결정(MCDM; Multi-Criteria Decision Making) 문제[23]로 구성하여 다루고 있다.

본 논문에서 제시되는 시스템은 대안 선택을 위한 의사결정이 이루어질 때마다 동적으로 구성된 의사 결정 문제를 해결함으로써, 해당 시점에서 최선의 대안으로 선택된 여행 계획을 추천한다. 이러한 추천 메커니즘을 실현시키기 위해서 다중 에이전트 시스템이 설계되었다. 에이전트들은 자신의 역할에 따라 문제를 구성하고 해결하기 위한 지식을 가지며, 문제가 의사결정 때마다 변화하는 환경에 적응하기 위한 협동 메커니즘에 따라 주어진 문제를 해결하기 위해 서로 협동한다. 본 추천 메커니즘을 통해 여행자들은 여행 계획을 얻지 못하는 위험을 줄이고, 처음에 가용한 대안보다 더 나은 여행 계획을 제공 받는 것이 가능하다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구들을 검토하고, 3장에서는 에이전트에 기반 한 여행 계획 추천 모형을 제시한다. 4장에서는 구현된 시스템과 여행 계획 예제를 제시한다. 5장에서는 서로 다른 예약 방식들을 비교하기 위한 시뮬레이션 분석 결과를 소개한다. 마지막으로 6장에서 결론을 제시한다.

## 2. 관련 연구

### 2.1 에이전트 기반 여행 지원 시스템

현재까지 에이전트 기반 여행 지원 시스템과 관

련해서 다양한 연구들이 수행되어 왔다[1, 7, 12, 13, 20, 21, 22, 24]. 이러한 연구 주제들에는 여행 계획 지원을 위한 에이전트 시스템의 요구 사항과 구조 설계[12, 20], 지능형 여행 계획 방법[7], 에이전트들의 협상 메커니즘[24], 에이전트의 지식 표현[22], 에이전트 기반 교통 시뮬레이션[1] 등이 포함된다. 이러한 연구들의 주요 목적은 여행 산업에 있어서 에이전트 기술을 적용하기 위한 기술적 요구 사항을 도출해내고, 에이전트 시스템의 상용적 가치를 증명하는 것이다.

여행 계획 지원을 위한 에이전트 시스템의 기술적 요구 사항으로는 여행 정보의 수집, 에이전트들 간의 통신과 협동, 온톨로지(ontology), 기존 시스템과의 통합, 사후 모니터링 등이 있다[20]. 한편 에이전트 시스템의 표준 구조의 하나로 제시되고 있는 FIPA(Foundation for Intelligent Physical Agents)[12]를 사용하여 기반으로 에이전트 기반 여행 지원 시스템의 상용적 가치에 대한 증명을 시도한 연구도 있다[21]. 일부의 연구들은 여행자가 에이전트의 지능성과 자율성을 통해 편리한 방법으로 여행 계획을 얻을 수 있다는 것을 보여주었다[13, 20, 24]. 연구 목적의 시험적 시스템 외에도, 익스피디아(expedia.com), 트래블로시티(travelocity.com), 오비츠(orbitz.com) 등과 같은 사이트들에서도 상용 여행 서비스를 제공하지만, 이러한 시스템들에서 소프트웨어 에이전트들은 여행 정보에 대한 검색과 예약과 같이 그 역할이 제한되어 있다[22].

현재까지의 상용 시스템과 연구 목적의 시험 시스템들은 여행자가 여행 계획을 하는 시점에 원하는 여행 계획이 없을 경우 여행자가 만족할 만한 여행 계획을 제시해주는 데에 한계를 가지고 있다. 이러한 상황은 방학, 휴가, 주말 등과 같은 여행 수요가 정점에 다다른 기간에 자주 발생한다. 현재의 시스템들은 여행자가 에이전트가 제시해준 대안에 대해 만족하지 않고 더 나은 대안을 원함에도 불구하고, 여행 계획을 처음 했던 시점 보다 나중에 좀 더 나은 여행 계획을 얻을 수 있는 기회를 암묵적으로 포기하고 있다. 이러한 한계를 극복하기 위해

서는 다음과 같은 추가적인 요구조건의 만족이 필요하다. 첫째, 에이전트들은 대안을 찾고, 최선의 대안을 선택하기 위해 필요한 지식을 가져야 한다. 둘째, 가능한 여행 계획이 계속 변화하는 환경에 대해 적용할 수 있는 에이전트들의 협동 메커니즘이 필요하다. 한편, 다중 기준 의사 결정(MCDM; Multi-Criteria Decision Making) 방법들은 에이전트들이 대안을 선택하기 위한 지식을 표현할 수 있는 이론적인 배경을 제공한다. 다음 절에서는 MCDM에 대한 연구들에 대한 간략한 검토가 기술된다.

## 2.2 MCDM 방법

MCDM 문제는 다수의 의사 결정 기준들의 존재하는 상황에서 대안들에 대한 평가를 다룬다[25]. 현재까지 다양한 형태의 MCDM 방법들이 제시되어왔는데, 가중합 모형(WSM; Weighted Sum Model), 가중곱 모형(Weighted Product Model), AHP(Analytic Hierarchy Process)[23], ELECTRE (Elimination and Choice Translating Reality)[4], TOPSIS(Technique for Order Preference by Similarity to Ideal Solution) 등이 이에 포함된다[14]. 각 방법들은 주어진 대안들 중에서 최선의 대안을 선택하는 방법에 있어서는 차이가 있지만, 선택들을 표현하는 대안들의 사용, 의사 결정 기준으로 사용되는 다중 속성들, 서로 다른 차원에서의 속성 표현, 속성에 대한 가중치 사용이라는 공통적인 측면을 가지고 있다[25].

MCDM 기법은 특히 에이전트 기술과 결합되어 에이전트의 의사결정에 도움을 주도록 이미 사용된 바 있다. 의료분야에서 장기 이식과 관련된 의사결정의 문제[19], e-러닝에서 학습 환경 구성을 위한 브로커 에이전트의 활용[17] 등 응용 연구 외에도 MCDM에 적합한 프로토콜에 관한 연구[18], 일반적 의사결정 모델 및 메커니즘에 관한 연구[27] 등이 존재한다.

여행 계획 추천에서 여행 계획 대안들은 다수의

성들에 의해서 비교될 수 있기 때문에 MCDM 방법의 적용이 가능하다. 특히, Saaty의 AHP[23]와 Belton[3]등의 연구는 다음과 같은 이유로 여행 계획 추천 모형을 위해 사용하기 적합하다. 첫째, MCDM 분야에서 가장 중요한 문제 중의 하나는 정성적인 자료를 정량화하는 데에 있다. Saaty의 연구는 각 기준의 관점에서 각 대안의 상대적인 중요도를 결정하기 위해, 쌍대비교(pairwise comparison)에 근거한 접근 방법을 제시하고 있다. 이러한 접근 방법은 여행 계획의 대안에서 각 기준의 상대적인 가중치를 제공하는 데에 적용될 수 있다. 둘째, 의사결정 행렬의 처리과정에서, 새로운 대안이 추가될 때 대안의 우선순위에 대한 불일치가 발생할 수 있다. Belton 등의 연구는 의사결정 행렬의 정규화 과정에서 각 대안들을 해당 기준의 합으로 나누는 과정에서 우선순위에 대한 불일치가 발생하며, 이를 피하기 위해 행렬의 정규화를 해당 기준에서 최대값으로 각 대안들을 정규화할 것을 제안했다. 이러한 Belton 등이 제시한 개념은 가능한 여행 계획이 변화하고 새로운 대안이 계속적으로 발생하기 때문에 여행 계획 추천 모형에 적합하다.

그러나 여행 계획 추천 모형을 위해서는 다음과 같은 추가적인 고려가 필요하다. 첫째, 의사결정 행렬은 새로운 대안이 생성될 때 마다 새롭게 구성될 수 있어야 한다. 즉, 전체 의사 결정 과정은 새로운 대안이 발생할 때마다 이루어지는 의사결정들의 연속으로 이루어진다. 둘째, 여행자들은 자신들의 선호도에 대한 명확한 표현을 하지 않을 수 있기 때문에, 각 기준에 대한 상대적인 중요도는 그들의 과거 여행 자료로부터 추론될 필요가 있다.

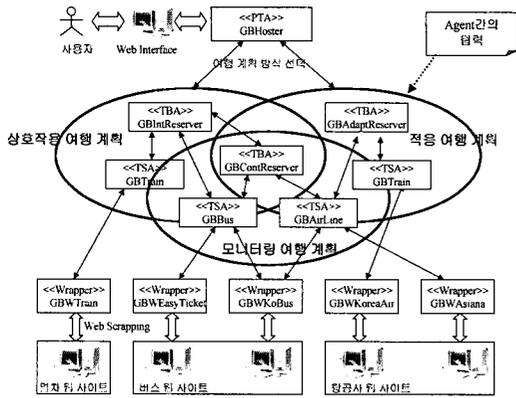
### 3. 에이전트를 통한 지능형 여행 계획

#### 3.1 시스템 개요

본 연구에서 구현된 에이전트 시스템은 길벗(GilBot)으로 명명되었다. 길벗에는 FIPA에서 정의된 표준

구조[12]를 따른 에이전트들로 구성된다. 에이전트들은 그 역할에 따라 개인 에이전트(PTAs ; Personal Travel Agents), 브로커 에이전트 (TBAs ; Travel Broker Agents), 서비스 제공 에이전트 (TSAs ; Travel Service Agents)로 구분된다. PTA는 여행자로부터 여행 계획에 대한 요구사항을 받고, 이에 대한 답변을 보여준다. TBA는 주어진 여행 계획 문제를 해결할 에이전트들에 대한 증개 서비스와 주어진 문제를 해결할 때까지 여행 계획의 상태에 따라 작업 참여 에이전트들을 통제한다. TSA는 주어진 여행 계획 문제에 대한 개별 대안을 제공한다. TSA는 TBA의 통제에 따라 주어진 작업에 대한 참여와 탈퇴가 결정된다. 특히, 길벗에서 TSA는 운송 수단에 대한 서비스를 제공하는 사이트에 따라 구성된 래퍼(Wrapper) 에이전트와 통신하며 부분적인 대안을 제공하게 된다. <그림 1>은 길벗 시스템의 전체적인 구조를 표현한다.

길벗 시스템은 사용자의 요구에 따라 크게 세 가지의 여행 계획 방법을 제공한다. 첫째, 상호작용 여행 계획(IP ; Interactive Planning)은 여행자가 수작업을 통해 여행 계획을 검색 및 수립하는 것으로서 기존의 에이전트 기반 여행 계획 시스템이 제공하는 서비스와 동일하다. 둘째, 모니터링 여행 계획(CP ; Continuous Planning)은 여행자가 원하는 여행 계획을 얻을 수 없을 경우, 원하는 여행 계획에 대한 상황을 계속 모니터링 함으로써 여행자가 해당 여행 계획을 얻을 수 있도록 지원해준다. 사용자 에이전트는 주어진 기간 동안 사용자가 원하는 여행 계획과 동일한 티켓을 예약하기 위해 반복적 주기적으로 검색 및 예약을 시도하게 된다. 셋째, 적응 여행 계획(AP ; Adaptive Planning)은 CP와 마찬가지로 여행자가 원하는 여행계획을 얻을 수 없는 경우에 적용되며, 본 논문에서 제시하는 여행 계획 추천 모형에 의해 여행자의 여행 계획을 지원한다. 실제로 IP와 CP의 작동 메커니즘은 간단하고 명확하므로, 본 논문의 3장에서는 주로 AP에서 사용되는 계획 검색 및 의사결정 지원 메커니즘에 대해 설명한다.



〈그림 1〉 길벗 시스템의 구조

### 3.2 여행 계획 문제의 정의

본 절은 여행 계획 문제를 명확히 하기 위하여 여행계획에 대한 공식적 정의를 기술한다. 여행자에 의해 주어진 여행 문제는 출발 도시와 시간, 도착 도시와 시간, 버스, 기차, 비행기 등의 여행 수단에 의해 기술된다. 여행  $T$ 는 한 장소에서 다른 장소로의 일련의 경로들의 집합으로 정의된다.

$$T = \{ (n, p_d, t_d, p_a, t_a, m) | n = 1, 2, 3, \dots, N, \\ p_d, p_a \in P, t_d, t_a \in Time, m \in M, p_d \neq p_a, t_d < t_a \}$$

- 단,  $n$ 은 여행 구간의 순서를 표시하는 일련번호,
- $P$ 는 장소들의 집합,
- $p_d$ 는 출발지,  $p_a$ 는 도착지,
- $Time$ 는 가능한 출발 시간과 도착 시간의 집합,
- $t_d$ 는 출발 시간,  $t_a$ 는 도착시간,
- $M$ 은 여행 방법의 집합이다.

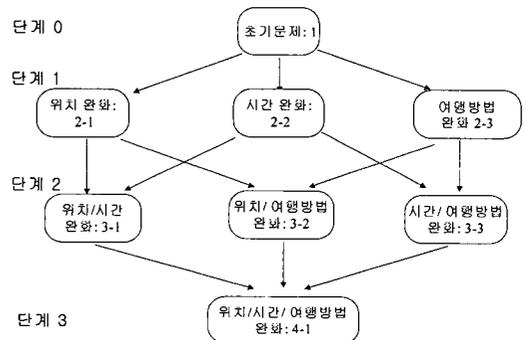
즉, 여행 계획을 수립하는 것은 일련의 가능한 여행 집합을 찾고 그 순서를 정하는 것으로 정의된다. 그러나, 여행 계획의 대안은 그 대안을 찾기 위해 여행자가 기다리는 시간, 이전 여행 계획의 취소 등과 같은 비용을 발생시킨다. 따라서 여행 계획의 대안  $A$ 는 정의된 여행에서 비용 요소를 추가함으로써 정의 된다.

$$A = \{ (t', c) | t' \in T, c \in C \},$$

- 단,  $t'$ 은 조건이 완화된 여행,
- $T$ 는 여행 집합,
- $c$ 는 해당 대안을 찾는 데 소요된 비용.

### 3.3 여행 계획 대안의 생성

서비스 제공 에이전트 (TSA)는 주어진 여행 계획 문제에 대한 조건을 완화시켜 여행 계획의 대안을 생성한다. 이러한 대안들은 조건 완화를 사용하여 생성된 질의어들에 의해 찾아진다. TSA가 현재의 질의어들을 가지고 대안을 찾는 데 실패하면, 추가적인 조건의 완화가 이루어진다. 이때, 기존의 질의어들과 새롭게 생성된 질의어들은 동시에 실행되면서 대안을 찾는다. 〈그림 2〉는 시간, 장소, 여행 방법 등의 세가지 변수들을 가진 여행 계획 문제에 대한 조건 완화 단계를 보여준다. 이때 위치 완화에서 중간 경유지를 통과하는 상황을 고려하면 더 일반적인 완화 방식을 제시할 수 있으나 한국적인 상황에서 주요 도시간의 이동의 경우 경유지에서 교통수단을 갈아타는 경우가 흔하지 않으므로 본 논문에서는 배제하고 동일 도시 혹은 인접 지역으로의 위치 완화만을 고려하였다. 추후 응용 분야를 국제 여행이나 타 국가의 상황까지 확장할 경우 이에 대한 연구가 추가적으로 필요할 수 있다. 조건 완화 단계는 조건이 완화된 변수들의 수를 의미하며 이는 대안 선택을 위한 의사결정에서 판단 기준으로 사용된다.



〈그림 2〉 세가지 변수에 대한 조건 완화

완화된 여행 변수들 외에도 대안들은 판단 기준이 되는 비용 속성을 갖는다.

### 3.4 의사결정 행렬

에이전트가 여행 계획의 대안들 중에서 사용자에게 보다 적합한 대안을 찾아 예약하기 위해서 앞서 소개된 대안들을 바탕으로 의사결정 행렬이 구성된다. 의사결정 행렬은 대안들과 여행자가 처음 제시했던 여행과의 속성별 차이들을 정규화된 형태로 표현하고 있다. 즉, 의사결정 행렬의 한 행에 속한 원소들은 한 대안에 대한 속성별 정규화된 거리들이며, 이 정규화 거리는 다음에 기술될 정의들을 통해서 결정된다.  $D_1, D_t, D_r, D_c$ 는 각각 장소, 시간, 여행 방법, 취소 비용의 관점에서 측정된 정규화 된 거리를 표시하며 다음과 같이 정의된다.  $D_1$ 은 장소들 사이의 지리적 거리에 기초하여 정의된다. 여행자가 초기에 제시한 여행  $i = (n, p_d, t_d, p_a, t_a, m)$ 와 장소 변수가 완화된 여행  $i' = (n, p_d', t_d, p_a', t_a, m)$ 에 대하여  $D_1 = (dist(p_d, p_d') / D_d + dist(p_a, p_a') / D_a) / 2$ 로 정의된다.

단,  $dist()$ 는 두 장소 사이의 지리적 거리 함수,  
 $D_d = \max \{ dist(p_d, p') | p' \text{은 } p_d \text{와 같은 도시에 소속} \}$ ,  
 $D_a = \max \{ dist(p_a, p') | p' \text{은 } p_a \text{와 같은 도시에 소속} \}$   
 이다.

$D_d$ 와  $D_a$ 는  $D_1$ 의 값을 0과 1사이로 정규화시키기 위해 정의된다. 이 값이 이와 같은 형태로 정의된 것은 장소의 완화는 처음에 주어진 장소가 속해 있는 동일한 도시의 범위 내에서 이루어지기 때문이다. 시간에 대한 거리를 표시하는  $D_t$ 도 비슷한 방식으로 정의된다. 여행자가 제시한 여행  $i = (n, p_d, t_d, p_a, t_a, m)$ 와 시간이 완화된 여행  $i' = (n, p_d, t_d', p_a, t_a', m)$ 에 대하여,  $D_t = (diff(t_d, t_d') + diff(t_a, t_a')) / T$ 로 정의된다.

단,  $diff()$ 는 두 시간 사이의 시간 차이 함수,  $T$ 는 시간 조건 완화의 허용 가능한 최대값이다. 여행방법에 대한 정규화된 거리인  $D_r$ 은 다음과 같이

정의된다. 여행자가 제시한 여행  $i = (n, p_d, t_d, p_a, t_a, m)$ 와 여행 방법 조건이 완화된 여행  $i' = (n, p_d, t_d, p_a, t_a, m')$

에 대하여,  $D_r = \begin{cases} 0 & \text{if } m = m' \\ 1 & \text{if } m \neq m' \end{cases}$ 로 정의된다.

취소비용을 의미한  $D_c$ 를 정의하기 위하여,  $c_b$ 를  $m$ 개의 대안과  $n$ 개의 의사결정 기준을 가진 ( $m \times n$ ) 행렬에서 기준에 예약된 대안  $A_b$ 의 취소비용으로 놓는다. 그러면, 대안  $A_i$ 의 취소비용 기준 값  $c_i$ 는 다음과 같이 정의된다. (이때  $i = 1, 2, \dots, m$ , 이고  $i \neq b$ 이다)

$$c_i = \gamma K,$$

단,  $\gamma$ 는 취소에 대한 청구율(charge rate),  $K$ 는 대안  $A_b$ 의 가격이다. 따라서 정규화된  $D_c$ 는  $c_i / C_{\max}$ 로 정의된다. (이 때  $C_{\max}$ 는 취소 비용의 최대값)

### 3.5 기준의 가중치 값에 대한 추론

의사결정 행렬에 의해 주어진 대안 중에서 사용자에게 적합한 대안을 추천하기 위해서는 정규화된 각 속성에 대해 사용자의 가중치를 알 수 있어야 한다. 본 논문에서는 MCDM 문제에 널리 사용되는 AHP를 이용한다. AHP를 위한 쌍대비교 행렬은 다음과 같이 정의된다. 쌍대비교 행렬의 원소  $p_{xy}$ 는  $y$ 기준의 관점에서  $x$ 기준의 상대적 선호도 값을 표시한다. 여행자가 과거에  $k$ 번의 여행 경력을 가지고 있다고 가정하면 여행 경력은 최종적으로 선택되었던 대안들의 집합으로 정의된다. 따라서,  $p_{xy}$ 는 다음과 같이 정의된다.

$$p_{xy} = \frac{1}{\sum_{i=1}^k a_{ix}} + \frac{1}{\sum_{i=1}^k a_{iy}} = \frac{\sum_{i=1}^k a_{iy}}{\sum_{i=1}^k a_{ix} \sum_{i=1}^k a_{iy}}$$

단,  $a_{ix}, a_{iy} \neq 0$ 이고  $a_{ix}, a_{iy}$ 는 최종적으로 선택되었던 대안  $i$ 의 기준  $x, y$ 에 해당하는 의사결정 행렬의 원소값들이다.

이때 과거에 선택된 대안들의 속성 값들은 여행자의 해당 속성에 대한 허용가능한 값으로 해석되

어 질 수 있기 때문에, 작은 값일수록 더 중요한 기준이 되는 것으로 해석할 수 있다. 또한,  $p_{xy}$ 가 상대적인 값을 계산하기 때문에, 두 값이 완화된 조건 (즉,  $a_{ix}, a_{iy} \neq 0$ )에 대해서만 계산이 이루어진다.

쌍대비교 행렬은 계산된  $p_{xy}$ 값( $x < y, x, y = 1, 2, \dots, n$ 에 대하여)을 바탕으로 대칭적으로 구성된다. 따라서  $p_{xy}$ 를 모두 구하고 난 이후 나머지 쌍대비교 행렬의 원소  $b_{ij}$ 는 다음과 같이 정의된다.

$$b_{ij} = \begin{cases} p_{ij} & \text{if } i > j, \\ 1 & \text{if } i = j, \\ 1/p_{ji} & \text{if } i < j, \end{cases}$$

그러면, 최종적으로 각 기준에 대한 상대적 가중치 값은 기하 평균 방식의 AHP 해법을 사용하여 계산된다[2]. 의사결정 기준  $i=1, 2, \dots, n$ 에 대하여, 기준  $i$ 에 대한 상대적인 가중치 값  $w_i$ 는 다음과 같이 계산된다. 이때 여행경력을 가지고 있지 않는 여행자의 경우에는 초기 가중치 값은 각 기준에 대해 동일하게 주어질 수 있다.

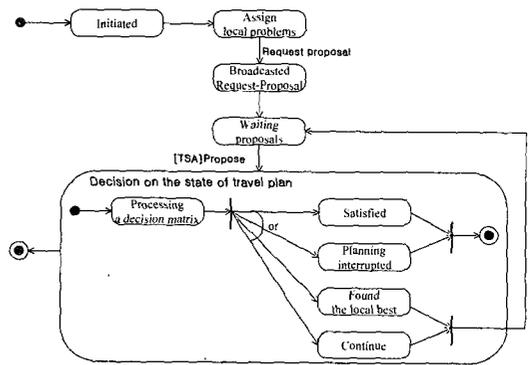
$$w_i = \left( \prod_{j=1}^n b_{ij} \right)^{\frac{1}{n}}$$

### 3.6 에이전트들의 협력

에이전트들은 자신을 역할을 수행하기 위한 지식을 가지고, 여행자의 과거 여행 경력을 관리하거나, 대안을 찾거나, 찾아진 대안 중에서 최선의 대안을 선택한다. 에이전트들의 협력 메커니즘을 위해 Controlled contract-net 프로토콜이 설계되었다. 여행 계획을 위해 설계된 에이전트 시스템의 경우, 하나의 주어진 여행 계획 문제에 대해서 여러 에이전트들이 자신들의 대안들을 동시에 제안하고 그 중에서 최선의 대안들이 선택된다는 점에서 Contract-net 프로토콜[5]이 적합하다. 그러나, 여행 계획 문제에서는 가능한 여행 계획이 계속해서 변화하기 때문에, 기존의 Contract-net 프로토콜에서와 같이 일련의 프로토콜의 흐름을 통한 협

동과정을 통해서 문제를 해결하는 데에는 한계가 있다. 예를 들면, 각 TSA가 제시한 대안에 의해서 다른 TSA들은 새로운 대안을 찾아야 하는 등의 영향을 미칠 수 있다.

<그림 3>은 여행 계획 추천을 위한 TBA의 협동 상태 모형을 표현한다. 협동 상태 모형은 TBA가 TSA가 제안한 대안들에 의해서 어떻게 TSA를 통제하는가를 보여준다.



<그림 3> TBA의 협동 상태 모형

주어진 여행 계획 문제를 풀기 위해 에이전트들 간의 협동 과정이 시작되면, TBA는 현재 문제 해결에 참여할 수 있는 모든 TSA들에게 'request-proposal' 메시지를 보내고, TSA들로부터 'propose' 메시지를 받는다. 그러면, TBA의 상태는 TSA들로부터 받은 메시지에 포함되어 있는 대안들을 평가하기 위한 'Decision on the state of travel plan' 상태가 된다. 이 상태에서는 먼저, 의사결정 행렬을 구성하고 각 대안들에 대한 결과 값을 해석한다. 해석된 결과 값과 현재 최선으로 선택된 대안에 따라서, TBA는 해당 메시지를 보내온 TSA에게 통제 메시지를 전해주기 위한 네 가지의 하위 상태 중의 하나로 이동한다. 네 가지의 하위 상태로는 'satisfied', 'planning interrupted', 'found the local best', 'continue' 상태 등이 있다.

여행 계획 추천 프로세스는 'satisfied' 또는 'planning interrupted' 상태에 도달하게 되면 종료가 된다. 'satisfied' 상태는 의사결정 행렬의 결과

값이 종료 되는 조건을 만족하고, 이는 최종적인 추천 대안이 생성되었음을 의미한다. ‘planning interrupted’ 상태는 최종적인 대안에 도달하지 않았지만, 여행자에 의해서 프로세스가 종료되었음을 의미한다. ‘found the local best’ 상태는 TSA 중의 하나가 자신이 제시할 수 있는 최선의 대안을 제시했음을 의미한다. ‘continue’ 상태는 프로세스가 계속해서 진행됨을 의미한다.

각 상태에서 TBA는 대안을 제시한 TSA에게 해당 대안이 선택되었는지 여부를 알려주는 ‘accept’ 또는 ‘reject’ 메시지를 각 TSA에게 보낸다. 또한, ‘continue’ 상태를 제외한 나머지 세 가지 상태에서는 ‘inform’ 메시지를 추가적으로 보낸다. ‘inform’ 메시지는 전체 프로세스가 종료된다거나 또는 TSA가 해당 프로세스에 참여할 필요가 없다는 것을 알려준다. 예를 들면, ‘found the local best’ 상태에서, TBA는 자신의 제시할 수 있는 최선의 대안을 제시한 TSA에게 대안의 의사결정 행렬 처리 값이 현재 선택된 대안보다 좋으면 ‘accept’ 메시지를 보내고, 나쁘면 ‘reject’ 메시지를 보낸다. 또한, 해당 TSA는 더 이상 좋은 대안을 제시할 수 없기 때문에 프로세스 참여를 종료시키는 내용을 담은 ‘inform’ 메시지를 보낸다.

반면, ‘continue’ 상태에서는, TSA들은 자신들이 제시한 대안에 대하여 ‘accept’ 메시지나 ‘reject’ 메시지를 받지만, ‘inform’ 메시지를 받지 않는다. 즉, 대안 추천 프로세스는 계속 진행 중이기 때문에 TBA는 TSA들로부터 추가적인 대안을 포함한 ‘propose’ 메시지를 기다린다. 이때, ‘accept’와 ‘reject’ 메시지에는 현재 선택된 대안의 상태를 표현하는 내용을 담고 있고, TSA들은 이 상태에 따라 후속적인 대안을 찾기 위한 질의어 생성을 한다. 예를 들면, 현재 선택된 대안이 두 변수에 대해서 조건이 완화된 것이라면, TSA들은 세 변수 이상에 대한 조건을 완화하지 않는다. 왜냐하면 추가적인 변수의 완화는 대안을 찾는다 하더라도, 현재 선택된 대안보다 결과가 좋지 않을 확률이 높기 때문이다.

### 3.9 에이전트의 추론 규칙 및 구현

길벗 시스템은 웹 상에서 여행자가 자유롭게 접속할 수 있도록 구현되었다. 이를 위해서 에이전트 엔진은 자바로 구현되었으며, 웹 인터페이스를 위해서는 JSP(Java Server Page)로 구현되었다. 특히, 에이전트들의 추론을 위해서 JESS(Java Expert System Shell)가 사용되었다.

에이전트들은 자신의 역할에 따른 추론 엔진을 가지고 작업을 수행한다. 예를 들면, TBA는 의사결정 행렬을 구성하고, 최선의 대안을 선택하며, 프로세스에 참여한 TSA를 통제하기 위한 메시지를 생산한다. <그림 4>는 TBA인 GBAdaptReserver가 의사결정 행렬의 처리 값에 따라 메시지를 생산하는 규칙을 표현한다.

```
(defrule test_score)
  (solution_state "incomplete")
  (TSA(name ?name)
   (tickets(ticket_no ?ticket_no))
   (test(=?score Minimum))
   →
  (assert(best-provider ?name)
   (solution_state "incomplete")
   (solution(tickets(ticket_no ?ticket_no))))
   (assert(message(performatives "accept")
   (receiver?name)))

(defrule other_message_satisfied)
  (solution_state "incomplete")
  (working_TSA(name ?name)
   (best_provider ?provider))
  (test(/= ?name ?provider))
  →
  (assert(message(performatives "reject")
   (receiver ?name)))
```

<그림 4> GBAdaptReserver의 메시지 생산 규칙

또한, TSA들은 주어진 여행 계획 문제에 대한 부분적인 대안을 제시하기 위하여 대안을 생성을 위한 질의어를 생산하고, 이에 따른 대안을 찾는다. <그림 5>는 TSA에서 장소 변수를 완화시키는 규칙을 표현한다.

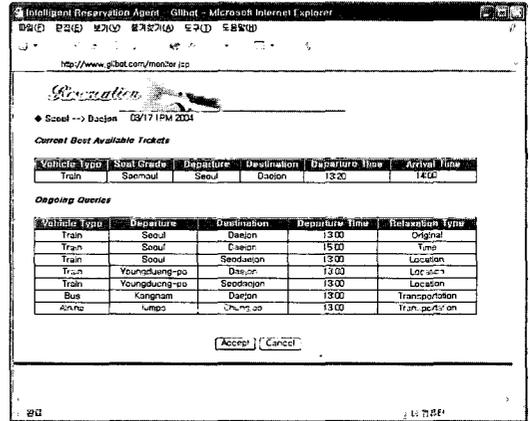
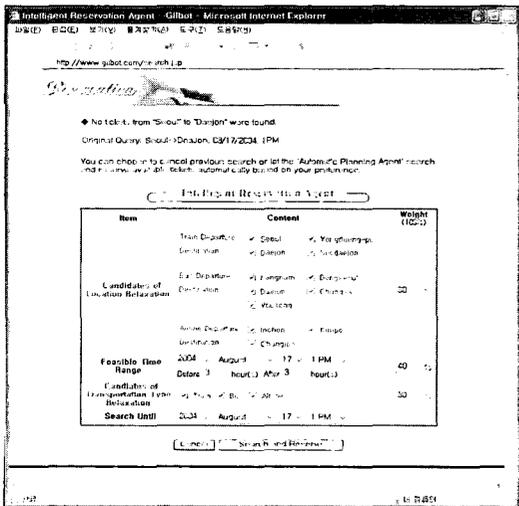
```
(defrule produce-location-relaxed-query)
(journey-leg(start-date ?s-date)
 (start-time ?s-time)
 (start-station ?s-station)
 (destination-station ?d-station))
(belong (city ?s-city) (station ?s-station))
(belong (city ?d-city) (station ?d-station))
(belong (city ?sa-city) (station ?sa-station))
(belong (city ?da-city) (station ?da-station)))

→

(assert (queries (start-date ?s-date)
 (start-time ?s-time)
 (start-station ?sa-station)
 (destination-station ?da-station)))
```

<그림 5> TSA에서 장소 변수 완화를 위한 규칙

<그림 6>은 서울시에서 대전시로의 여행 예제에 대한 길벗 시스템 화면들이다. 화면은 여행자가 여행 계획을 추천 받기 위해 적응 여행 계획(Adaptive Travel Planning)을 사용했을 때의 화면들이다. 첫 화면은 여행자가 적응 여행 계획을 실행하기 위한 화면이고, 두 번째 화면은 추천 상태를 모니터링 할 수 있는 화면이다. 즉, 특정 일자에 여행자가 서울로부터 대전으로 가는 여행 대안을 요청했을 경우, 여행자의 요구를 정확하게 만족시키는 대안이 없는 경우 <그림 6>의 첫번째와 같은 화면을 제시하여, 장소, 시간, 여행 방법 등에 대한



<그림 6> 길벗 시스템의 화면 예제

완화 범위를 입력하도록 요청한다. 두 번째 화면은 적응 여행 계획에 의해서, 현재 확보된 티켓과 지속적으로 검색을 수행하고 있는 대안들의 리스트를 보여주고 있다.

## 4. 사례 연구

### 4.1 사례 기술 및 가정

본 절에서는 길벗 시스템을 사용한 여행 계획의 구체적인 사례를 제시하도록 한다. 본 사례의 여행 계획은 특정 출발지에서 특정 도착지까지의 가능한 경로에서 운행되는 여행 티켓을 구하고자 한다. 즉, 어떤 여행자가 서울에서 대전으로, 토요일 오후 1시에 출발해서, 다음날 오후 5시에 돌아오자 한다고 가정하자. 이 경우 두 개의 단위 여행 구간 (1, 서울, 07/17/2004, 오후 1시, 대전, 기차), (2, 대전, 07/18/2004, 오후 5시, 서울, 기차)이 도출된다. 여행 계획을 위해서는 모든 여행 구간에 대해서 필요한 티켓을 구해야 하지만, 각 티켓은 같은 방식으로 구해지므로 하나의 티켓을 구하는 것에 대해서만 기술하도록 한다.

본 사례 연구를 위하여 다음과 같은 내용들이 가정된다.

첫째, 여행자가 여행 계획을 하는 시점에 원하는 초기 티켓은 구할 수가 없다.



posal'에 대한 답신으로 GBAdaptReserver에게 보낸다. 여기서 GBTrain과 GBBus가 자신들이 현재 상태에서 찾은 최선의 대안이 <표 2>와 <표 3>과 같다고 가정하자. GBAirline은 SA<sub>1</sub>과 DA<sub>1</sub> 사이의 가능한 경로가 없으므로 'sorry' 메시지를 보낸다. 두 TSA, GBTrain과 GBBus는 자신들이 보낸 메시지에 대한 답신을 기다린다.

<표 2> GBTrain의 대안

출발지	출발 시간	도착지	도착 시간	티켓 수
ST <sub>1</sub>	15:00	DT1	16:50	12

<표 3> GBBus의 대안

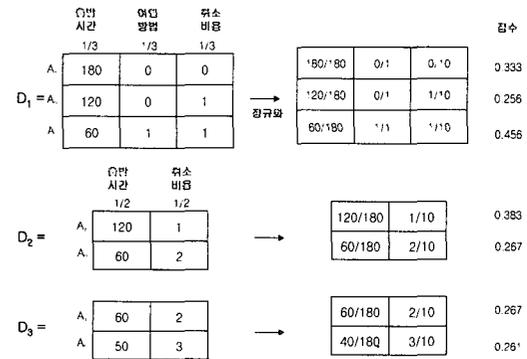
출발지	출발 시간	도착지	도착 시간	티켓 수
SB <sub>1</sub>	14:00	DB1	16:00	10

각 TSA들로부터 받은 메시지에 포함되어 있는 대안을 통하여, GBAdaptReserver는 현재 선택되어 있는 대안까지 포함하여 의사결정 행렬을 구성하고, 최선의 대안을 선택한다. <그림 8>은 이러한 프로세스 동안 생성되는 의사결정 행렬들과 이를 처리하는 내용을 보여준다.

<그림 8>과 같이, 첫 번째 의사결정 행렬 D<sub>1</sub>은 새롭게 생성된 두 개의 대안과 현재 선택되어있는 대안으로 구성되어 있는 세 개의 대안을 가진다. 또한 현재 대안들은 시간, 여행 방법 등의 여행 변수에 대한 조건 완화가 이루어졌고, 취소 비용이 고려되었기 때문에 세 가지의 의사결정 기준을 가지고 있다. 따라서 의사결정 행렬 D<sub>1</sub>은 (3×3) 행렬로 구성된다. 각 행렬의 원소 값들은 제3장에서 정의된 내용을 따른다. 특히, 새롭게 생성된 대안의 취소 비용은 현재 선택된 대안을 취소하면 발생하는 비용으로 값이 설정된다.

첫 번째 의사결정 행렬에서는 두 번째 대안, 즉, GBTrain에 의해 새롭게 생성된 대안이 현재 상태에서 최선의 대안이 된다. 첫 번째 의사결정 행렬에 대한 처리 결과로 GBAdaptReserver는 GBTrain에게는 'accept' 메시지를, GBBus에게는 'reject' 메

시지를 보낸다. 또한 GBBus에 의해 생산될 수 있는 최선의 대안은 여행 방법 변수만이 완화된 티켓이라 하더라도, 현재 선택된 대안이 더 높은 우선순위를 갖고 있다. 따라서, GBAdaptReserver는 GBBus에게 더 이상 해당 프로세스에 참여할 필요가 없다는 내용을 포함한 'inform' 메시지를 보낸다.



<그림 8> 의사결정 행렬의 처리

<그림 8>과 같이, 첫 번째 의사결정 행렬 D<sub>1</sub>은 새롭게 생성된 두 개의 대안과 현재 선택되어있는 대안으로 구성되어 있는 세 개의 대안을 가진다. 또한 현재 대안들은 시간, 여행 방법 등의 여행 변수에 대한 조건 완화가 이루어졌고, 취소 비용이 고려되었기 때문에 세 가지의 의사결정 기준을 가지고 있다. 따라서 의사결정 행렬 D<sub>1</sub>은 (3×3) 행렬로 구성된다. 각 행렬의 원소 값들은 제3장에서 정의된 내용을 따른다. 특히, 새롭게 생성된 대안의 취소 비용은 현재 선택된 대안을 취소하면 발생하는 비용으로 값이 설정된다.

첫 번째 의사결정 행렬에서는 두 번째 대안, 즉, GBTrain에 의해 새롭게 생성된 대안이 현재 상태에서 최선의 대안이 된다. 첫 번째 의사결정 행렬에 대한 처리 결과로 GBAdaptReserver는 GBTrain에게는 'accept' 메시지를, GBBus에게는 'reject' 메시지를 보낸다. 또한 GBBus에 의해 생산될 수 있는 최선의 대안은 여행 방법 변수만이 완화된 티켓이라 하더라도, 현재 선택된 대안이 더 높은 우선순위를 갖고 있다. 따라서, GBAdaptReserver는

GBBus에게 더 이상 해당 프로세스에 참여할 필요가 없다는 내용을 포함한 'inform' 메시지를 보낸다.

첫 번째 의사결정 결과로 'accept' 메시지를 받은 GBTrain은 계속해서 더 나은 대안을 찾고, 대안이 찾아지면 대안을 포함한 'propose' 메시지를 GBAdaptReserver에게 보낸다. <그림 8>에서는 GBTrain이 추가적으로 두 번의 'propose' 메시지를 보냈다고 가정하고 있다. 예를 들면, 두 번째 의사결정 행렬에서는 시간 변수의 값이 초기 여행자가 원했던 시간에서 한 시간 늦은 대안을 보냈기 때문에 생성되었다. 이 행렬의 결과로 새롭게 보낸 대안이 선택된다. 마지막으로, 세 번째 행렬에서는 새롭게 보내진 대안이 이전에 선택된 대안보다 우선순위가 더 낮음을 알 수 있다. 따라서, GBAdaptReserver는 의사결정 과정에서 더 나은 대안을 찾지 못했기 때문에 전체 프로세스를 종료시킨다. 이는 전체 프로세스를 종료 시키는 조건을 만족시키기 때문에, 최종적인 대안으로써 세 번째 행렬에서 선택된 대안을 여행자에게 추천하게 된다. 또한, 'reject' 메시지와 'inform' 메시지를 GBTrain에게 보냄으로써, GBTrain이 보낸 대안은 선택되지 않았고, 더 이상 프로세스에 참여할 필요가 없다는 것을 알려준다.

## 5. 시뮬레이션

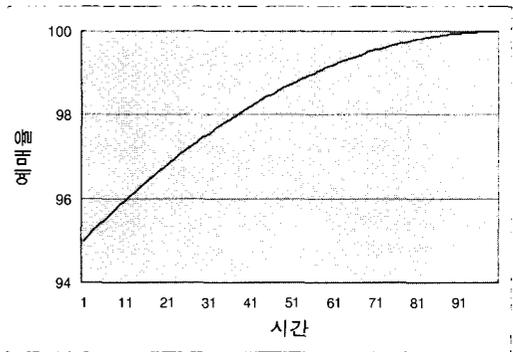
### 5.1 시뮬레이션 개요

본 논문에서 제시된 세 가지 방식(IP, CP, AP)의 비교를 위해 시뮬레이션 실험을 수행하였다. 피크 시즌의 수요 행태를 모형화 하기 위하여 피크 시즌의 특정 시점에 추가적인 예매 및 특정 시점에 기존 예매의 취소가 모두 비동질적(non-homogeneous) 포아송 분포를 따른다고 가정하였고 각각의 분포의 평균이 시간에 따라 선형적으로 변화한다고 가정하였다. 즉, 취소와 관련된 포아송 분포의 평균은  $\lambda_1(t) = a_1t + b_1$ 로 정의되고 추가적인 예매는  $\lambda_2(t) = a_2t + b_2$ 로 주어진다고 할 때  $t$ 시

점의 총 예매율은 다음과 같은 평균을 갖게 된다.

$$\begin{aligned}\bar{\lambda}_t &= \int_0^t (a_2s + b_2) ds - \int_0^t (a_1s + b_1) ds \\ &= \left( \frac{a_2 - a_1}{2} \right) t^2 + (b_2 - b_1)t = at^2 + bt\end{aligned}$$

이 식을 바탕으로 시뮬레이션 실험의 타임 프레임 동안 예매율은 다음 그림의 커브처럼 주어진다 고 가정하고 이를 이용하여 각각의 방식에 대해 실험을 수행하였다.



<그림 9> 시뮬레이션에 사용된 예매율 변화 곡선 (초기 예매율 95%,  $a = -0.02$ 를  $b = -2$  적용한 경우)

실험과 관련해서 한 가지 언급할 점은 AP와 CP 방식의 에이전트를 이용하는 사용자의 행태는 엄밀하게 정의될 수 있지만 IP를 사용하는 사용자에 대해서는 그 행태에 대한 가정이 필요하다. 따라서 본 논문에서는 IP의 경우 한 사용자가 주어진 시뮬레이션 타임 프레임 속에서 4회 예약을 시도한다고 가정하였다. 이때 첫 시도에서는 본래의 티켓 예매를 시도하고 그 이후의 시도에서는 탐색의 범위를 조건을 매 시도 횟수마다 한 단계(depth)씩 넓힌다고 완화한다고 가정하였다.

### 6.2 세 방식의 비교

비교를 위한 성능 평가 지수는 세 가지 하위 지수, 즉 예약된 티켓과 최초 희망 티켓의 유사도, 취소 비용, 그리고 예약에 실패하는 위험도 등을 바

탕으로 만들어졌다. 시뮬레이션 실험의 간편성을 위해서 티켓들의 유사도는 조건 완화 단계를 이용하였다. 취소 비용은 시스템이 더 좋은 조건의 새로운 티켓을 예약하기 위해 기 예약된 티켓을 취소하는 경우 발생할 수 있다. 위험도는 특정한 시점에 티켓의 예매를 완전히 실패할 확률을 의미하며 이는 완화 단계가 낮은 티켓일수록 위험도가 높다고 볼 수 있다. 각 세가지 하위 지수는 그 값의 범위가 0에서 1 사이에 존재하도록 정규화 되었다.

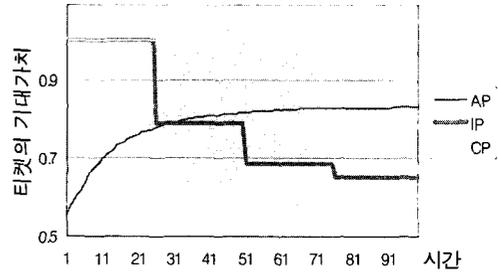
하위 메트릭을 통합하는 종합 성능 평가 지수는 사용자의 의사 결정 선호도 혹은 정책을 반영하여 세 가지 하위 지수를 선형결합하여 정의하였다. 이때  $w_{dist}$ ,  $w_{cost}$ ,  $w_{risk}$  등이 각 하위 지수에 대한 가중치로 사용된다.

$t$ 시점의 성능 평가 지수

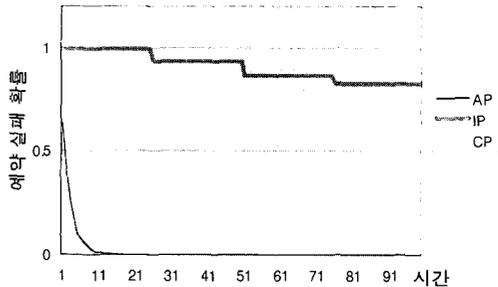
$$\begin{aligned}
 &= (t\text{시점에서 예약된 티켓의 유사도 기대값}) \cdot w_{dist} \\
 &\quad - (t\text{시점에서 총 예상 취소 비용}) \cdot w_{cost} \\
 &\quad - (t\text{시점에서 예약에 실패할 기대 위험}) \cdot w_{risk} \\
 &= E\left(1 - \frac{d(t)}{d_{max}} \mid reservation\_successful\right) \cdot w_{dist} \\
 &\quad - E\left(n_c(t) \frac{c_{cancel}}{c_{max}}\right) \cdot w_{cost} - P_{no\_ticket}(t) \cdot w_{risk}
 \end{aligned}$$

이때  $d(t)$ 는  $t$ 시점의 완화 단계이며,  $d_{max}$ 는 정규화를 위한 상수이다.  $n_c(t)$ 는  $t$ 시점까지 취소의 횟수이며  $c_{cancel}/c_{max}$ 는 정규화된 취소 비용이다.  $P_{no\_ticket}(t)$ 는  $t$ 시점에서 예약에 실패할 확률이다. CP 방식에서는 사용자가 입력한 티켓과 동일한 티켓만 예약되므로  $d(t)$ 와  $n_c(t)$ 가 항상 0의 값을 갖는다. 시뮬레이션 실험은 앞서 제시된 피크 시점의 예약율을 사용하여 0에서 100까지 시점에 대해 1000회 반복 수행되었다. <그림 10>, <그림 11>, <그림 12>에는 각 세가지 방식의 비교가 티켓의 가치, 예매에 실패하는 위험도, 그리고 취소 비용 관점에서 제시되고 있다. 티켓의 가치 측면에서 CP는 항상 1을 유지하며 IP는 계단형으로 감소하고 AP는 단조 증가 하는 것을 확인할

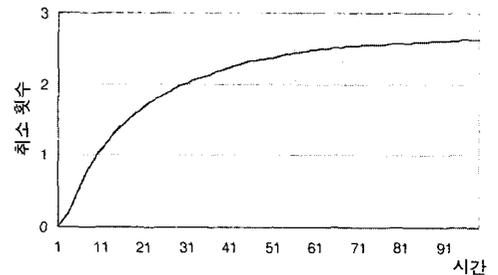
수 있다. <그림 11>에서는 AP가 예매 실패 위험도 측면에서 다른 두 방식보다 훨씬 나음을 보여주고 있으며 <그림 12>에서는 AP방식에서  $t=100$  시점에서 취소의 횟수가 약 2에서 3 사이임을 보여주고 있다.



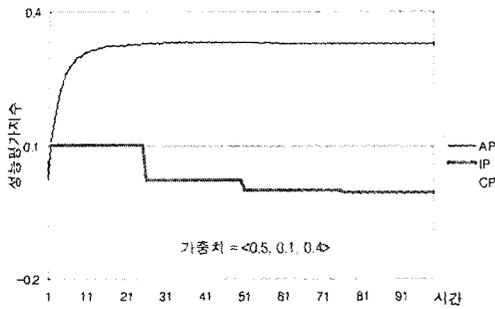
<그림 10> 예매된 티켓의 가치 측면에서 세 가지 방식의 비교



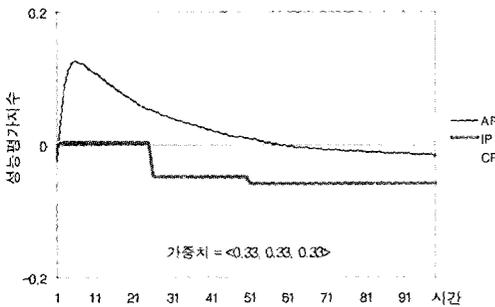
<그림 11> 예약 실패 확률 관점에서 세 가지 방식의 비교



<그림 12> AP에서 각 시각별 취소 횟수



<그림 13> 취소 비용이 작을 때의 세 방식의 비교



<그림 14> 취소 비용이 상대적으로 높을 때의 세 방식의 비교

다음으로 앞서 제시된 종합 성능 평가 지수를 사용하여 다른 형태의 가중치가 세 하위 지수에 주어질 때 세가지 방식의 성능을 비교해 보았다. 첫째, <그림 13>은 비교적 작은 취소 비용 가중치를 사용한 것으로 AP 방식이 다른 두 방식보다 뛰어난 결과를 보여주고 있다(적은 취소 비용을 이용해도 유사한 결과를 얻는다). 이는 AP가 여러 차례 예약을 시도할 수 있고 또한 취소에 따른 비용이 낮기 때문이다. 반면, 두 번째인 <그림 14>의 예에서는 취소 비용이 비교적 높다고 가정되고 있다. 여기서는 취소가 발생하지 않는 CP가 더 좋은 성능을 나타낼 수 있음을 보여준다(취소 비용을 높여도 유사한 결과를 얻는다). 이 예들에서 IP는 계속 좋지 못한 결과를 보여주고 있는데 이는 본 시뮬레이션 실험에서 사용자가 수동적으로 예약을 시도하는 데는 횟수의 한계가 있다고 보았기 때문이다.

위 시뮬레이션 결과들로부터 알 수 있는 사항들을 요약하면 다음과 같다. 첫째, AP와 CP 방식이 IP보다 전반적으로 좋은 결과를 가져오는데, 이는 에이전트 시스템이 반복적이고 지능적인 역할을 사용자를 대신하여 효과적으로 수행할 수 있기 때문이다. 두 번째, 각 방식의 성능은 각각 세 가지 하위 지수의 값이나 가중치의 변화에 의해 크게 의존적일 수 있다. 예를 들어 AP 방식은 위험을 기피하는 성향의 사용자들에게 더욱 효과적일 수 있고 그러한 사용자들은 해당 지수에 높은 가중치를 갖게 된다. 세 번째, <그림 14>에서 보여지듯이 취소 비용이 높을 때에 AP의 성능은 크게 떨어지지만 만약 사용자에게 짧은 타임프레임만 주어진다면 CP보다 더 좋은 선택일 수 있다. 마지막으로, 시스템이 세 가지 방식의 예약을 지원하므로 사용자는 각각 사용자의 선호도나 상황 및 시점에 맞게 세 가지 방식을 혼합해서 사용하거나 그 중 하나를 선택해서 사용할 수 있다.

## 6. 결 언

본 논문에서는 에이전트에 기반한 피크 기간에서의 여행 계획 추천 시스템을 제시하였다. 본 시스템의 목적은 피크 시즌 등에서 여행자가 여행 계획을 하는 시점에 원했던 여행 계획을 얻기 어려울 때에 원래 여행 계획과 유사한 대안 중에서 보다 만족할만한 여행 계획을 여행자에게 제공하는 것이다. 제시된 시스템에서는 에이전트들이 제약 조건 완화를 통해 가능한 대안을 찾고 다중 기준 의사결정(MCDM; Multi-Criteria Decision Making) 모형을 이용해서 현재 상태에서 최선의 대안을 선택할 수 있는 지식을 갖도록 설계되었다. 또한, 가능한 여행 계획이 계속적으로 변화하는 환경에 적응할 수 있는 협동 메커니즘을 제시함으로써, 에이전트들이 전체 여행 계획의 상태에 따라 서로 협동하면서 대안을 찾을 수 있도록 하였다. 제시된 시스템을 통해 여행자는 여행 계획을 얻지 못하는 위험을 최소로 하면서, 처음보다 나은 계획을 얻을

수 있다. 또한, 설계된 에이전트 협동 메커니즘은 에이전트들의 동시 작업을 통해 시스템의 효율을 높이고, 문제 해결의 전체 상태에 따라 문제 해결에 대한 에이전트의 참여와 탈퇴를 결정하고, 불필요한 작업 수행을 최소화 하도록 한다. 제시된 메커니즘을 바탕으로 JSP, JESS를 사용한 프로토타입 시스템이 구축되었으며, 세 가지 다른 방식의 예약 방법들의 특성 관찰과 비교를 위해서 시뮬레이션 실험이 수행되었다.

현재의 연구를 발전시키기 위한 추후 연구과제는 다음과 같다. 첫째, 제시된 모형은 여행자의 도움 없이 과거의 여행 경력에 기반하여 여행자의 선호도를 추측하기 때문에 급한 용무로 인한 여행과 같은 특정한 상황에 맞는 여행자의 선호도를 반영하는 데에는 한계가 있다. 따라서 다양한 상황에 따른 선호도 반영 방식과 의사결정 지원이 요구된다. 둘째, 여행 수단의 중간 변경, 호텔 예약 등과 같은 추가적인 여행 요소를 반영한 좀 더 포괄적인 여행 계획 모형에 대한 추가적인 연구가 필요하다. 마지막으로 여행자의 추가 입력 사항이나, 대안을 찾기 위한 시간 간격의 결정 등의 같은 상용화와 관련된 추가적인 고려가 필요하다.

## 참 고 문 헌

- [1] Balmer, M., B. Raney and K. Nagel, "Agent-Based Activities Planning for an Iterative Traffic Simulation of Switzerland," STRC 2004, Swiss, 2004.
- [2] Barzilai J., "Deriving Weights from Pairwise Comparison Matrices," *Journal of the Operational Research Society*, Vol.48(1997), pp.1226-1240.
- [3] Belton, V. and T. Gear, "On a short-coming of Saaty's method of analytic hierarchies," *Omega*, Vol.11(1983), pp.228-230.
- [4] Benayoun, R., B. Roy and N. Sussman, "Manual de reference du programme electre, Note de Synthese et Formation," (1996), *Direction Scientifique SEMA*, Paris, French.
- [5] Brenner, W., R. Zarnekow, and H. Wittig, *Intelligent Software Agents : Foundations and Applications*, Springer-Verlag, 1998.
- [6] Bridgman, P.W., *Dimensional Analysis*, Yale University Press, New Haven, CN, 1992.
- [7] Camacho, D., D. Borrajo and J.M. Molina, "Intelligent Travel Planning : A Multi-Agent Planning System to Solve Web Problems in the e-Tourism Domain," *Autonomous Agents and Multi-Agent Systems*, Vol.4(2001), pp.387-392.
- [8] Chen, S.J. and C.L. Hwang, "Fuzzy Multiple Attribute Decision Making : Methods and Applications," *Lecture Notes in Economics and Mathematical Systems*, No.375(1992), Springer-Verlag, Berlin, Germany.
- [9] Decker, K., K. Sycara and M. Williamson, "Matchmaking and Brokering," *Proceedings of the Second International Conference on Multi-Agent Systems(ICMAS-96)*, December, 1996.
- [10] Finin, T. and G. Wiederhold, "An overview of KQML : A Knowledge Query and Manipulation Language," Dept. of Computer Science, Stanford Univ., 1991.
- [11] Foundation for Intelligent Physical Agents, "FIPA Architectural Overview," *FIPA '99 Specification*, July, 1999, Available from <http://www.fipa.org>.
- [12] Foundation for Intelligent Physical Agents, "FIPA Personal Travel Assistance Specification," Aug., 2001, Available from <http://www.fipa.org>.
- [13] Hsu, M., P.H. Chang, Y. Wang and V. Soo, "Multi-Agent Travel Planning through Coalition and Negotiation in an Auction," PRIMA2003,

- Seoul, Korea, 2003
- [14] Hwang, D.L. and K. Yoon, *Multiple Attribute Decision Making : Methods and Applications*, Springer-Verlag, New York, NY, 1981.
- [15] Jennings, N.R., K. Sycara and M. Wooldrige, "A Roadmap of Agent Research and Development," *Autonomous Agents and Multi-Agent Systems*, Vol.1(1998), pp.275-306.
- [16] Lootsma, F.A., "Numerical scaling of human judgment in pairwise-comparison methods for fuzzy multi-criteria decision analysis," *Mathematical Models for Decision Support*, NATO ASI Series F, Computer and System Sciences, Springer-Verlag, Berlin, Germany, 1988, pp.57-88.
- [17] Manouselis, N. and D. Sampson, "Multi- Criteria Decision Making for Broker Agents in eLearning Environments," *Operational Research : an International Journal*, Vol.2 (2002), pp.347-361.
- [18] Matsatsinis, N.F. and P. Delias, "A Multi-criteria Protocol for Multi-agent Negotiations," SETN 2004, Samos, Greece, pp.103-111.
- [19] Moreno, A., A. Valls and D. Sanchez, "A multi-criteria decision aid agent applied to the selection of the best receiver in a transplant," 4th International Conference on Enterprise Information Systems, Ciudad Real, April, 2002.
- [20] Ndumu, D.T., J.C. Collis and H.S. Nwana, "Towards Desktop Personal Travel Agents," *BT Technology Journal*, Vol.16, No.3(1998), July, pp.69-78.
- [21] O'Sullivan, D. et al., "Experiences in the use of FIPA agent technologies for the development of a Personal Travel Application," *International Conference on Autonomous Agents*, Barcelona, Spain, 2000.
- [22] Paprzycki, M. et al., "Knowledge Representation in the Agent-Based Travel Support System," *LNCS 2457*, Springer-Verlag, 2002, pp.232-241.
- [23] Saaty, T.L., *The Analytic Hierarchy Process*, McGraw-Hill International, New York, NY, 1980.
- [24] Soo, V. and S. Liang, "Recomending a Trip Plan by Negotiation with a Software Travel Agent," *M. Klush and F. Zambonelli Eds : CIA 2001, LNAI 2182*, 2001, pp.32-37.
- [25] Triantaphyllou, E. and S.H. Mann, "An examination of the effectiveness of multi-dimensional decision-making methods : A decision-making paradox," *International Journal of Decision Support Systems*, Vol.5 (1989), pp.303-312.
- [26] Triantaphyllou, E., B. Shu, S. Nieto Sanchez and T. Ray, "Multi-Criteria Decision Making : An Operations Research Approach," *Encyclopedia of Electrical and Electronics Engineering*, Vol.15(1998), pp.175-186.
- [27] Wu, C.W. and B.H. Far, "An Implemen-tation of Decision Making Models and Mechanisms for Multi-Agent Systems in Competitive Environments," *IEEE Canadian Conference on Electrical and Computer Engineering(CCECE 2003)*, May, 2003, pp.1227-1230.