동적 스케줄링 문제에서 사용자 상호작용을 이용한 제약조건 완화

Constraint Relaxation using User Interaction in Reactive Scheduling Environment

이 훈*, 정종진**, 조근식**

Hoon Lee*, Jong Jin Jung**, and Geun Sik Jo**

요 약

스케줄링 문제는 응용 영역에 따라 각기 다른 작업과 자원을 가지며, 이들간에 여러 가지 제약 조건이 존재한다. 이러한 문제에 인공지능 기법을 이용하여 문제를 효율적으로 해결할 수 있다. 그러나, 동적으로 변하는 상황에서 모든 제약조건을 만족하는 최적해를 구할 수 없는 경우, 부분적으로 만족하는 해를 구하기 위해 제약 조건들은 수정 또는 추가되어야 하며, 때로는 고려 대상에서 제외되어야 한다. 부분적으로 만족하는 해를 구하기 위해, 수정되어야 할 적절한 제약 조건을 스케줄링 시스템이 자동으로 선택하기에는 많은 문제가 있다. 이러한 문제를 해결하기 위해, 본 논문에서는 부분적으로 만족하는 해를 구해야 하는 경우, 제약 조건 완화를 위해 사용자와 상호 작용하며 사용자의 의견을 제약 조건에 반영하는 지능형 사용자 인터페이스(Intelligent User Interface)의 모델을 설계하고 구현하였다. 김포공항의 운항 스케줄 데이터를 사용한 실험에서 제약 조건을 모두 적용하여 이를 만족하는 해를 구하지 못한 경우, 사용자의 의견이 반영된 제약 조건완화를 통해 검색 도메인의 증가와 제약 조건을 부분적으로 만족하는 해를 구할 수 있었다.

Abstract

In optical scanning holography, 3-D holographic information of an object is generated by 2-D active optical scanning. The optical scanning beam can be a time-dependent Gaussian apodized Fresnel zone plate. In this technique, the holographic information manifests itself as an electrical signal which can be sent to an electron-beam-addressed spatial light modulator for coherent image reconstruction. This technique can be applied to 3-D optical remote sensing especially for identifying flying objects. In this paper, we first briefly review optical scanning holography and analyze the resolution achievable with the system. We then present mathematical expression of real and virtual image which are responsible for holographic image reconstruction by using Gaussian beam profile.

*LG-EDS시스템 공공사업부 사업추진부(LG-EDS System, Public Div. New Business Dept.)

**인하대학교 전자계산공학과(Dept. Computer Science and Eng., Inha Univ.)

· 논문번호 : 98-2-7

· 접수일자 : 1998년 12월 30일

이 논문은 1997년 한국학술진흥재단의 공모과제 연구비에 의하여 연구되었음.

Ⅰ. 서 론

스케줄링은 다양한 응용 영역을 갖는 문제로 많은 작업과 자원들간의 제약조건으로 인해 해결하기 어려운 문제이다[1],[2]. 스케줄링 문제에 대한 많은 연구 중 전문가 시스템은 제약조건을 나타내는 복잡하고 많은 양의 규칙들로 구성되어 있으며, 동적으로 변하는 상황에서 해를 구하기 위해 제약조건들의 변경 및 추가가 불가피한 경우가 있다. 이를 위해 사용자와 효과적으로 상호 작용하여 제약조건을 완화하는 지능형 사용자 인터페이스에 대한 연구가 요구된다. 지능형 사용자 인터페이스는 작업에 대한 도메인 지식을 갖고 있어야 한다. 이러한 지식을 이용해 스케줄링작업에서 사용자가 고려해야 할 문제를 자동으로 처리해줌으로써 효과적인 작업 진행이 이루어지도록 해야 한다. 또한, 사용자가 해당 문제에 대한 전문가인 경우, 사용자의 요구를 제약조건에 반영해 적절히 동작할 수 있어야 한다.

본 논문에서는 사용자의 의견을 제약조건에 반영하기 위한 지능형 사용자 인터페이스를 설계하고 구현하였다. 구현된 사용자 인터페이스는 본 연구실에서 개발된 주기장 관리 전문가 시스템(Ramp Activity Coordination Expert System : RACES)에서 미배정 항공편을 사용자가 수작업으로 주기장에 배정하는 작업을 초점으로 구현되었으며, 사용자와 상호 작용하여 미배정 항공편을 위한 후보 주기장을 제안해 주는 기능을 갖고 있다. 또한, 제안된 해에 대한 적합성 검증을 위해 주기장 관리 전문가 시스템을 에이전트 기반으로 설계하여, 동적 재스케줄링 에이전트와 상호 작용을 통해 미배정 항공편을 위한 후보 주기장의 적합성 여부를 검증하도록 하였다.

Ⅱ. 관련 연구

사용자와 상호 작용하는 시스템은 사용자가 컴퓨터 시스템의 기능을 최대한으로 활용할 수 있도록 한다. 이러한 목표를 위해 시스템이 사용자와 마치 대화하듯 상호 작용하는 사용자 인터페이스에 대한 연구가 이루어져 왔다. 관련한 연구의 예로 사용자의 음성과 같은 자연어를 이용한 입력방법, 메뉴나 아이콘을 이용한 직접 제어방식 등이 있다. 그러나, 사용자의 요구를 적절히 반영하기 위해서는 보다 지능적인 사용자 인터페이스에 대한 연구가 요구되어진다.

2-1 지능형 사용자 인터페이스를 위한 기법들

지능형 사용자 인터페이스에 대한 연구는 컴퓨터 시스템과 이것을 사용하는 사람사이의 상호작용에 초점이 맞추어져 있다. 많은 연구자들은 사람이 사용하는 자연어를 이용한 연구와 각각 다른 여러 사용자에게 적절히 상호 작용할 수 있도록 사용자에 대한 적응성(adaptation)을 향상시키기 위한 연구를 수행해왔다. 이러한 연구를 통해 얻어진 지능형 사용자 인터페이스를 위한 기법에는 사용자 모델링(User Modeling), 자연어 처리 기법(Natural Language Technology), 설명 생성 기법(Explanation Generation) 등이 있다[3].

- 사용자 모델링

다양한 사용자에 따라 시스템이 적절히 반응하여 상호작용이 이루어질 수 있도록 시스템이 사용자에 대한 지식을 관리/유지하는 기법이다.

- 자연어 처리 기법

사람이 이해하기 쉬운 텍스트나 음성과 같은 자연어를 시스템이 이해하거나 생성할 수 있도록 하는 기법이다.

- 설명 생성 기법

시스템으로부터 얻은 해를 분석하여 이에 대한 설명을 사용자에게 제공하는 기법이다.

2-2 지능형 사용자 인터페이스의 응용분야

앞서 언급한 기법들을 이용한 응용 분야에는 지

능형 교수 시스템(Intelligent Tutoring System), 지능형 도움말 시스템(Intelligent Help System) 그리고 정보 여과 시스템(Information Filtering System)등이 있다[4],[5].

▪ 지능형 교수 시스템

특정 영역에 대한 지식을 사용자에게 교육시키기 위한 시스템으로 사용자에게 해당 영역에 대한 질의와 응답을 통해 사용자의 이해 정도를 평가한다. 사용자에 대한 평가결과를 바탕으로 사용자에게 질의에 대한 자세한 해를 제시하거나, 사용자의 응답이 잘못된 경우 사용자의 잘못된 지식에 대한 문제점을 지적한다. 이러한 지능형 교수 시스템에서 사용자에 대한 정확한 평가를 위해 사용자 모델을 응용한 학습자 모델(Student Model)을 구성하여 사용자에 관한 정보를 유지한다.

▪ 지능형 도움말 시스템

도움말 시스템은 사용자가 특정 작업을 수행할 때 작업에 대한 도움말을 제공한다. 도움말 시스템은 특정 작업에서 사용자의 질의를 평가하여 사용자에게 도움말을 제공한다는 점에서 지능형 교수 시스템과 유사하나, 지능형 교수 시스템은 도움말 제공 및 사용자의 잘못된 지식 진단에 의한 사용자 교육에 있는 반면, 지능형 도움말 시스템은 도움말을 이용해서 사용자가 작업을 원활히 진행할 수 있도록 하는데 목적이 있다. 사용자의 작업에 대한 적절한 도움말을 제공하기 위해서는 사용자의 질의를 분석하여 사용자의 목적을 파악해야 한다. 이러한 사용자의 목적을 파악하기 위해 플랜인식(Plan Recognition)이 필요하다. 사용자가 특정 작업을 수행할 때 작업의 플랜을 서브플랜(Sub Plan)으로 분류하고, 이에 대한 도움말을 제공한다[6][7]. 이러한 도움말은 각각의 서브플랜을 메타룰(Meta Rule)에 적용하여 룰로부터 도움말을 제공한다. 예를 들어, 전자우편 시스템의 경우 사용자가 도착된 편지에 대한 작업으로 read and save 명령을 선택했다면 사용자의 플랜은 read와 save 두 가지 서브플랜으로 구분되며, read에 대한 도움말과 save에 대한 도움말이 메타 룰에 의해 선택되어지고 두 가지 도움말이 결합된 도움말이 사용자에게 제공된다.

▪ 정보 여과 시스템

인터넷 상의 방대한 정보 중에서 사용자가 원하는 정보는 매우 부분적이다. 정보 여과 시스템은 이러한 부분적인 정보를 많은 정보들 가운데서 검색할 때 사용자가 원하는 정보만을 여과하여 주는 시스템으로 사용자가 인터넷 상에서 검색한 자료들의 패턴을 생성하여 사용별로 그 패턴을 유지한다. 사용자가 검색을 시작하면 사용자별 패턴을 이용하여 사용자에게 인터넷 상의 정보를 여과하여 보여 준다. 또한, 패턴에 해당하는 정보들이 인터넷 상에서 변경되었을 때 이를 알려주는 기능도 포함하고 있다. 이러한 정보 여과 시스템 역시 정확한 사용자의 요구를 파악하기 위해 사용자에 대한 정확한 정보를 유지하고 있어야 하며, 이를 위해 정확한 사용자 모델이 요구되어진다.

2-3 스케줄링 문제

스케줄링 문제는 다양한 영역의 문제에서 많은 작업과 자원들간의 제약 조건이 존재하며, 이러한 제약 조건들로 인해 해결하기 어렵다. 이러한 스케줄링 문제를 해결하기 위해 인공지능 기법을 많이 이용하고 있다. 인공지능에서는 주로 기호적 관계에 기반을 두고 경험이나 직관에 의한 문제 해결을 위해 추론에 기반을 둔 접근방법을 많이 사용한다. 따라서, 기호적 제약조건이 복잡하게 연관되어 있고, 전문가의 경험에 의한 지식을 필요로 하는 많은 스케줄링의 경우 이러한 방법을 적용하면, 문제를 효율적으로 해결할 수 있다[2],[8].

스케줄링 문제의 성격에 따라서 스케줄링 결과가 한번 구해지고 결과가 변하지 않는 예측 스케줄링(predictive scheduling)과 외부 요인에 의한 상황의 변동이나, 사용자의 요구 사항이 증가함에 따라 새로운 제약조건이 추가되거나 삭제되어 기존의 스케줄링 결과를 재조정해 주어야 하는 동적

재스케줄링(reactive scheduling)으로 나눌 수 있다. 대부분의 스케줄링 문제는 동적 재스케줄링을 요구하게 되고, 따라서 해결하기가 용이하지 않다[9]. 동적 재스케줄링에서 제약조건을 모두 만족하는 최적해를 구할 수 없을 때, 제약 조건을 완화하여 부분적으로 만족하는 해를 구할 수 있다. 제약조건 완화에서 스케줄링 문제의 전문가인 사용자와 시스템이 상호 작용한다면 보다 효과적으로 제약조건을 완화할 수 있을 것이다. 본 논문에서는 이러한 제약조건 완화를 위해 지능형 사용자 인터페이스를 제시한다.

Ⅲ. 지능형 사용자 인터페이스의 구조 및 처리과정

3-1 지능형 사용자 인터페이스의 구조

제약조건 완화를 위한 사용자 인터페이스는 스케줄링 제약조건, 자원(resource)들에 대한 정보, 완화된 제약조건, 제약조건을 이용한 스케줄링 엔진, 제약조건 완화기로 구성되어 있다. 스케줄링 엔진은 스케줄링 제약조건을 이용하여 스케줄링 작업을 수행한다. 만일 스케줄링 제약조건을 모두 만족하는 해를 구하지 못하였을 때는 제약조건 완화가 필요하다. 사용자는 전체 제약조건 리스트를 참조하여 제약조건을 완화한다. 사용자에 의한 제약조건 완화가 발생하면, 제약조건 완화기는 제약조건 완화를 위한 사용자 플랜(User Plan)을 생성한다. 생성된 사용자 플랜을 인지하여 제약조건 사이에 존재하는 상위, 하위 관계를 고려하며 사용자가 완화한 제약조건을 생성한다. 이렇게 완화된 제약조건을 이용하여 스케줄링 엔진은 부분적으로 만족하는 해를 구한다.

이와 같은 사용자 상호작용을 통한 제약조건 완화를 실험하기 위해 본 연구실에서 개발한 주기장 관리 전문가 시스템(RACES : Ramp Activity Coordination Expert System)을 대상으로 에이전트 기반의 스케줄링 시스템을 설계하였다. 설계한 에이전트 기반 스케줄링 시스템은 일일 스케줄에 의해 자원을 할당하는 예측 스케줄링 에이전트(Predictive Scheduling Agent), 스케줄의 변경에 따라 스케줄링 결과를 재스케줄하는 동적 재스케줄링 에이전트(Reactive Scheduling Agent), 사용자 인터페이스 에이전트(User Interface Agent), 그리고 각각의 에이전트들에 의해 데이터 베이스내의 정보가 변경되었을 경우 데이터들의 일관성을 유지하기 위한 데이터 핸들링 에이전트(Data Handling Agent)로 구성된다.

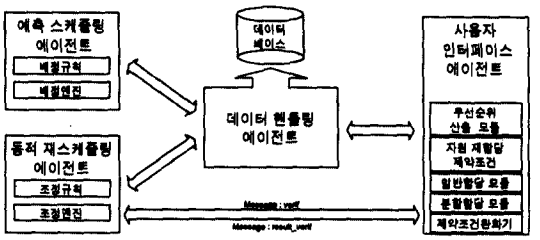


그림 1. 에이전트 기반의 스케줄링 시스템의 구조
Fig. 1. Architecture of agent based scheduling system.

본 논문에서는 지능형 사용자 인터페이스에 해당하는 사용자 인터페이스 에이전트를 설계하고 구현하였다. 구현된 사용자 인터페이스 에이전트는 스케줄링에 실패한 작업에 대한 자원의 재할당을 위해 사용자에게 후보 자원을 제안한다. 만일 제약조건을 모두 만족하는 해를 구할 수 없는 경우, 제약조건을 부분적으로 만족하는 해를 구하기 위해 사용자와 상호 작용하며 제약조건을 완화한다. 위와 같은 기능을 갖춘 지능형 사용자 인터페이스는 그림 1과 같이 스케줄링 결과를 바탕으로 자원 사용 현황에 대한 수치 데이터를 산출하여 자원의 우선 순위를 결정하는 우선 순위 산출 모듈, 자원 재할당 제약조건, 자원 재할당 제약조건에 의해 자원을 할당하는 일반할당 모듈과 작업을 분할하여 자원에 할당하는 분할할당 모듈로 구성되어 있다.

3-2 우선순위 결정을 위한 수치자료 산출 과정

스케줄링 작업이 종료되었을 때, 얻어진 스케줄링 결과를 분석하여 자원들의 사용현황에 대한 수치자료를 산출한다.

1) 자원별 수치 자료

- 자원에 할당된 작업의 수

스케줄링 작업의 결과 각 자원별로 자원에 할당되었던 작업들의 수를 합산하여 산출한다.

- 작업의 자원 사용시간 , 작업의 자원 사용율

각 자원별로 자원에 할당된 작업들의 자원 점유시간을 합산하여 자원 사용시간을 산출하고, 이것을 다시 일일 자원 사용시간을 이용하여 자원 사용율을 산출한다.

2) 작업별 수치 자료

- 자원 점유 시간

각 작업마다 당일 할당되었던 자원들의 점유시간을 산출한다.

수치자료 산출로부터 얻어진 데이터들은 사용자가 스케줄링에 실패한 작업에 할당하기 위한 작업을 수행할 때 사용자에게 제공되는 자원들의 우선순위 결정에 반영된다. 예를 들어, 사용자가 자원별로 작업의 자원 사용율을 최대화할 수 있는 해를 원한다면 자원별 작업의 자원 사용율이 순위 결정에 반영된다.

3-3 주기장 문제에서의 자원 재할당 과정

미 스케줄링 작업에 대한 처리과정은 미 스케줄링 작업에 대한 자원의 대한 분할할당 검사 과정과 분할할당에 실패할 경우, 이미 다른 작업이 할당되어 있는 자원에서 할당된 작업을 다른 자원으로 분할할당하고, 미 스케줄링 작업을 그 자원에 할당하는 과정으로 구분된다. 이것을 주기장 문제에 적용시켰을 때 자원은 주기장에 해당하며 미 스케줄링 작업은 배정에 실패한 항공기이다. 배정에 실패한 항공기를 주기장에 할당하는 과정은 다음과 같다.

1) 항공기의 크기에 의한 배정 가능한 후보 주기장의 선정

각 항공편마다 운항하도록 정해진 항공기들은 종류별로 크기가 다르며 크기에 따라 배정할 수 있는 주기장이 각각 다르다. 따라서 입력된 미배정 항공편을 배정할 주기장은 우선 크기에 따라 후보 주기장들이 주기장 리스트들 중에서 검색된다.

2) 항공편에 따른 후보 주기장의 여과

크기에 따라 검색된 후보 주기장들로부터 항공편에 따른 여과 작업이 진행되는데 다음과 같은 기준을 갖는다.

① 야간주기 항공기 여부 검사

야간 주기 항공기는 당일 공항에 도착하여 다음날 출발하는 항공기로 야간 주기가 가능한 주기장에 배정되어야 한다. 이것을 위해 후보 주기장으로부터 야간 주기가 가능한 주기장들만 여과시킨다.

② 여객기와 화물기 구분

항공편 운항 시 이용되는 항공기가 여객기인지 화물기인지 구분한다. 화물기의 경우, 화물기를 주기할 수 있는 주기장들만 여과시킨다. 여객기의 경우 다음의 규칙을 따른다.

③ 여객기 청사 구분

여객기는 국내선과 국제선으로 구분하고, 국내선 항공편은 국내선 청사에 국제선 항공편은 국제선 1청사, 국제선 2청사에 소속된 주기장들만 여과시킨다. 특정 A 항공사 소속이 아닌 타 항공사 소속의 항공기는 항공사별로 지정된 주기장을 이용한다. A항공사 소속의 항공기인 경우에는 항공편 유형을 고려하여 다음과 같이 청사를 결정한다.

- 국제선 도착편 , 국제선 출발편 : 국제선 2청사
- 국제선 도착편 , 국내선 출발편 : 국제선 2청사
- 국내선 도착편 , 국제선 출발편 : 국제선 2청사
- 국내선 도착편 , 국내선 출발편 : 국내선 청사

3) 항공편 시간대에 따른 후보 주기장의 여과

위의 과정들에서 여과된 후보 주기장들에서 배정하고자 하는 항공편의 도착 시간과 출발 시간을 고려하여 사용 가능한 후보 주기장들을 여과한다. 이 과정에서는 후보 주기장에 미배정 항공편을 배정하기 위하여 이미 배정되어 있는 항공기들을 다른 주기장으로 견인한다. 이를 위해 후보 주기장에

이미 배정된 항공편의 도착시간이 미배정 항공편의 도착시간보다 빨라야 하며, 반대로 출발시간은 미배정 항공편의 출발시간보다 늦어야 한다. 이를 바탕으로 후보 주기장에 이미 배정된 항공기의 점유시간과 미배정 항공편의 점유시간 차를 구하며 이것은 후보 주기장들의 우선 순위 결정에 이용된다. 이러한 과정을 통해 미배정 항공편을 배정할 수 있는 1차 후보 주기장들을 선정한다.

4) 후보 주기장의 우선 순위 결정

앞의 과정들을 통해 얻어진 후보 주기장의 우선 순위에 따라 아래와 같이 가중치를 산출한다.

- 배정된 주기장의 청사 검사

항공편에 따라 배정할 수 있는 해당 청사의 주기장일 경우 10을 아닌 경우 5를 부여한다.

- 급유시설에 대한 검사 (비행시간이 7시간 이상인 국제선 항공기의 경우)

급유시설을 2개 갖춘 주기장은 가중치 10, 1개를 갖춘 주기장의 경우 5를 부여하며, 급유시설이 없는 경우 가중치를 부여하지 않는다.

- 후보 주기장의 시간차에 따른 우선 순위

앞서 후보 주기장에 이미 배정된 항공기의 점유시간과 미배정항공편의 점유 시간차를 구하였다. 이러한 시간차가 적은 주기장에 높은 가중치를 부과하기 위해 큰 순서로 순위를 매겼다. 매겨진 순위에 각각 5를 곱하여 가중치를 산출한다.

- 사용자의 선택 기준에 따른 가중치 부여

위의 여과 과정에서 사용자가 선택한 공항 사용 현황 기준의 순위에 5를 곱한 가중치를 구한다.

위에 항목 검사를 통해 얻어진 가중치들을 합하여 얻어진 후보 주기장과 주기장들의 가중치를 통해 사용자가 미배정 항공편을 배정할 후보 주기장을 선택하며 해당 주기장에 배정된 항공기를 견인하여 배정할 수 있는지 검사하고 가능하다면 미배정 항공편을 후보 주기장에 배정하고 그 결과를 동적재스케줄링 에이전트에게 평가하도록 한다.

3-4 사용자와의 상호작용을 이용한 제약조건 완화

미 스케줄링 작업에 대한 자원 재할당을 수행한 결과, 경우에 따라 사용자에게 제약조건을 모두 만족하는 최적해를 제공하지 못하는 경우가 발생한다. 이러한 경우 부분적으로 만족하는 해를 구할 수 있도록 사용자에 의한 제약조건 완화가 요구된다. 다음의 그림은 스케줄링에 실패한 작업에 대한 자원 재할당 과정과 재할당 과정에서 사용자와 상호 작용하는 것을 보여주고 있다.

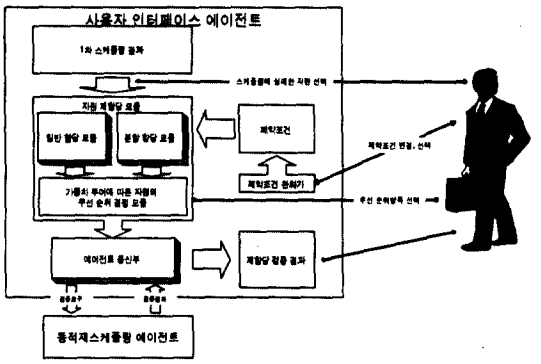


그림 2. 제약조건 완화를 위한 사용자 인터페이스 구조
Fig. 2. Architecture of user interface for constraint relaxation.

그림 2에서 1차 스케줄링에서 실패한 작업을 사용자가 선택하면 자원 재할당 모듈을 통해 자원 재할당이 이루어진다. 자원 재할당 모듈은 일반할당 모듈과 분할할당 모듈로 구분되며 분할할당 모듈은 작업을 시간으로 분할하여 자원에 할당한다. 재할당 모듈을 통해 해를 구하지 못하는 경우 제약조건 완화기를 통해 제약조건을 완화한다. 제약조건 완화기는 사용자에게 제약조건의 리스트를 보여주고 사용자는 제약조건을 추가 또는 삭제하고 수치적인 제약조건의 경우 값을 줄이거나 늘임으로써 제약조건을 완화한다. 이 때 제약조건 완화기는 사용자가 제약조건을 완화하는 작업에 대한 사용자의 제약조건 완화 플랜을 생성한다. 사용자가 제약조건을 완화하는 작업(action)은 다음과 같은 플랜 라이브러리(plan library)로 나타내어진다.

본 논문에서는 Kautz의 사용자 플랜 라이브러리를 적용하여 사용자 플랜을 생성 및 인지를 위한

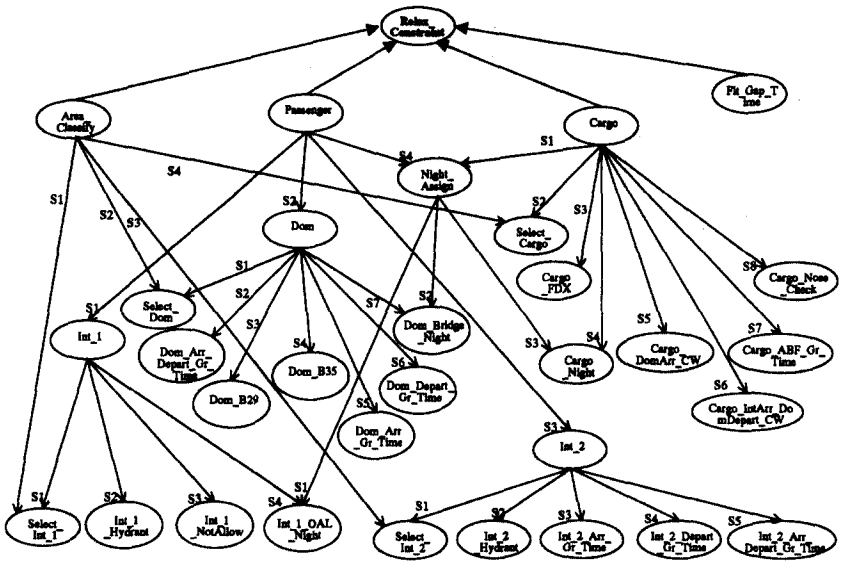


그림 3. 사용자 플랜 라이브러리

Fig. 3. User plan library.

사용자 플랜 라이브러리를 구축하였다[10],[11]. Kautz의 플랜 라이브러리 상의 각 노드는 사용자의 action을 나타내는 플랜이고 각 노드의 하위 노드는 상위 노드의 서브 플랜이다. 노드의 각 아크는 플랜의 스텝을 표시한다. 각 아크에 표시된 플랜의 스텝은 상위 플랜을 수행하기 위한 순서로 번호가 매겨진다. 본 논문에서 구축한 플랜 라이브러리 상의 각 노드들은 제약조건을 완화하는 사용자의 플랜을 나타낸다. 주기장 문제의 특성상 제약조건을 완화하는 사용자의 플랜은 하위 플랜들의 수행 순서에는 영향을 받지 않는다. 대신에 어떤 제약조건을 완화하기 위해서는 상위 제약조건을 완화해야 한다. 예를 들어, 국내선 청사의 한 주기장에 대한 제약조건을 완화하는 플랜을 생성하기 위해서는 사용자가 국내선 청사의 제약조건을 완화한다는 플랜이 생성되어야 한다. 플랜 수행순서에 영향을 받지 않는다는 점과 각 플랜 상의 상위 관계를 명확히 해야 한다는 점을 고려하여 제약조건 완화 플랜 라이브러리를 그림 3과 같이 구축하였다. 사용자 플랜 라이브러리를 통한 사용자 플랜 생성은 다음과 같다. 완화를 위해 한 제약조건이 선택되면 해당 제약조건부터 상위로 검색하여 각각의 제약조건의 완화에 대한 플랜이 다음과 같이 생성된다.

예 1. 야간주기 제약조건을 완화한 경우

{NightAssign(S4(KL101)),
Passenger(KL101),Relax_Constraint(KL101)}
∨{NightAssign(S1(KL101)),
Cargo(KL101),Relax_Constraint(KL101)}

예 2. 야간주기 제약조건 중 국내선청사의 브릿지 주기장에 대한 제약조건을 완화한 경우

{Dom_Bridge_Night(S7(KL101)),
Dom(S2(KL101)),Passenger(KL101),
Relax_Constraint(KL101)} ∨
{Dom_Bridge_Night(S7(KL101)),
Night_Assign(S4(KL101)),Passenger(KL101),
Relax_Constraint(KL101)} ∨
{Dom_Bridge_Night(S7(KL101)),
Night_Assign(S1(KL101)),Cargo(KL101),
Relax_Constraint(KL101)}

예 1의 경우 야간주기 제약조건을 완화하면 여객청사의 주기장과 화물기 주기장의 야간주기 제약조건을 모두 완화시킨다. 위와 같이 사용자의 제약조건 완화플랜이 생성되면 제약조건 완화기는 다음과 같은 알고리즘으로 상위, 하위 제약조건을 검색하여 제약조건을 완화한다.

```
begin
 User selects an activity in failed-activities.
 Resource is reassigned to resources.
 IF reassign operation fails THEN
  Constraint Relaxation starts.
  LOOP
   begin
    User select an constraint.
    IF selected constraint is not a numeric constraint THEN
     IF selected constraint is added to constraints THEN
      begin
      Search higher-constraint and check higher constraint.
       IF higher-constraint is not in constraints THEN
        LOOP
         Add higher-constraint to constraints.
        UNTIL There is no longer unselected higher-constraints.
       ENDIF
      end
     ELSEIF selected constraint is deleted in constraints THEN
      begin
      Search lower-constraints and lower constraints.
      IF lower-constraints are in constraints THEN
       LOOP
        Delete lower-constraints in constraints.
       UNTIL There are no longer selected lower-constraints.
      ENDIF
      end
     ENDIF
    ELSEIF selected constrain is numeric constraint THEN
     User changes the numeric constraint value
    ENDIF
   end
  UNTIL User stops or get Partial Solution
 ENDIF
 IF There is solution THEN
  Display the solution to User.
 ENDIF
end
```

[제약조건 완화 알고리즘]

위와 같은 알고리즘으로 제약조건을 완화하여 얻어진 후보 주기장들에 대해서는 가중치를 적용하여 우선 순위를 제시하고 사용자는 미배정 항공편을 실제로 배정할 후보 주기장을 선택한다. 사용자가 선택한 주기장은 동적 재스케줄링 에이전트에 의해 실제 배정이 가능한지 적합성 여부가 검사되고 그 결과가 사용자에게 제시된다.

Ⅳ. 실험결과

사용자는 다음과 같은 제약조건 완화 윈도우를 통해 제약조건을 완화한다.

사용자는 제약조건항목을 클릭함으로써 제약조건을 완화할 수 있다. 또한 수치적인 제약조건에 대해 제약조건 항목 옆에 나타나는 수치를 직접 입력하거나 버튼을 클릭함으로써 변경할 수 있도록 하였다.

실험은 김포공항의 운항 스케줄 데이터로서 96년 7월 6일자 미배정 항공편인 국내선 여객기 7244를 사용하였다. 실험대상에 대한 제약조건을 모두 적용시킨 경우와 청사선택에 대한 제약조건을 사용자가 완화하여 적용시킨 경우를 비교하였다. 제약 조건을 모두 적용시켜 구한 해는 주기장 S56으로 해당 주기장에 배정된 항공기를 분할 할당하려 했을 때 실패하였다. 그래서 사용자가 국내선 항공편은 국내선 청사만 이용해야 하는 제약조건을 완화하여 국제선 1청사를 이용하도록 하였다. 얻어진 해는 주기장 S146, S147, S156이었으며, 사용자의 선택에 의해 주기장 이용율을 극대화하는 방향으로 가중치를 구하였을 때 주기장 S146이 선정되었다.

표 1은 위의 예에서 사용자에 의해 청사에 대한 제약조건이 완화되어 국내선 청사 이외의 청사들을 허용하였을 때 각각 얻어진 결과이다.

표 1에서 제약조건을 모두 적용시킨 경우 재할당 모듈에 의해 선정된 주기장에 대해 분할할당 검사를 한 결과 주기장에 배정된 항공편을 분할할당을 수행할 수 없어서 미배정 항공편을 배정할 수 없었다. 부분적으로 제약조건을 만족하는 해를 구하기 위해 국내선 항공편에 대해 국내선 청사만을 사용하게 하는 제약 조건을 완화하였다. 표 1에서 보는 바와 같이 국내선 청사만을 이용하는 제

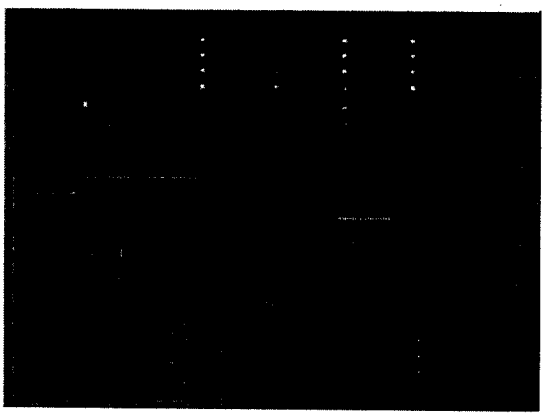


그림 4. 제약조건 완화 윈도우

Fig. 4. Window for constraint relaxation.

표 1. 제약조건 완화에 따른 배정결과 평가

Table 1. Estimation scheduling result by constraint relaxaiton.

제약조건	1차 여과 (항공편 특성고려)	2차 여과 (미배정편 시간대고려)	해
국내선 청사 사용	10개	1개	0개
국제선 1청사 포함	30개	3개	3개
국제선 2청사 포함	44개	4개	3개
화물기 청사 포함	29개	7개	7개

약조건을 완화하여 사용자의 제약조건 선택에 의해 사용할 수 있는 청사를 다음과 같이 국제선 1청사, 국제선 2청사, 화물기 청사를 사용할 수 있도록 하였을 때 제약 조건에 의해 항공기의 크기와 항공편의 특성에 따라 여과된 후보 주기장의 수, 앞서 여과된 주기장 중 미 스케줄링 항공편의 시간에 따라 여과된 주기장의 수, 두가지 여과과정을 통해 얻어진 주기장에 대해 분할할당 검사를 수행하여 최종적인 해로 얻어진 주기장의 수가 증가하는 것을 볼 수 있다. 이 결과는 제약 조건이 완화되어 부분적으로 만족하는 해를 구할 수 있음을 보여 주고 있다.

V. 결 론

본 논문에서는 동적 스케줄링에서 제약 완화를 위한 지능형 사용자 인터페이스의 구조를 정의하고 구현하였다. 구현한 지능형 사용자 인터페이스는 에이전트 기반의 스케줄링에서 스케줄링 작업 중 사용자에게 작업을 재할당할 수 있는 자원을 제공하기 위해 정해진 제약조건에 따라 작업에 대한 자원 재할당을 수행하여 모든 제약조건을 만족하는 최적해를 구하지 못할 때 사용자와 상호작용하여 제약조건을 완화하게 하였다. 이를 위해 사용자 플랜을 생성하여 제약조건을 완화함으로써 부분적으로 만족하는 해를 구할 수 있었다. 사용자 플랜 생성 및 인지를 위해 Kautz의 플랜 라이브러리를 응용하여 주기장 특성상 하위 노드의 아크에 순서를 반영하지 않고 아크 고유의 이름을 부여한 플랜 라이브러리르 사용하였다. 또한, 사용자 인터페이스 내의 재할당 모듈을 통해 얻어진 해의 적합성 검증을 위해 동적 재스케줄링 에이전트와 메시지 전송에 의한 상호 협력을 통해 적합성을 평가하여 그 결과를 사용자에게 제시하였다. 제약조건을 완화하여 부분적으로 만족하는 해를 구할 수 있음은 김포 공항의 운항 데이터들을 이용한 실험 통해 확인할 수 있었다.

에이전트들 간의 상호협력은 메시지 전송 방식을 이용하여 사용자 인터페이스와 동적 재스케줄링 에이전트 사이에서만 이루어지고 있다. 만일 사용자 인터페이스 에이전트와 예측 스케줄링 에이전트사이에 상호협력이 이루어진다면 미 스케줄링 작업을 할당하기 위한 후보 자원들을 사용자에게 제시할 때 예측 스케줄링 에이전트 내의 자원 할당 제약조건을 통해 후보 자원들의 선정이 가능할 것이다. 이것을 위해 예측 스케줄링 에이전트와의 메시지 전송에 대한 연구와 에이전트 간의 상호작용을 위한 에이전트의 플랜 생성 및 인지에 대한 연구가 요구된다. 이를 통해 스케줄링 문제에 있어서 보다 에이전트들 간의 상호협력을 향상시킬 수 있을 것이다.

참 고 문 헌

[1] A E. Croker, V. Dhar, A Knowledge Representation for Constraint Satisfaction Problems, *IEEE transaction on Knowledge and data Engineering*, Vol. 5, No. 5, pp. 740-752, October, 1994.

[2] APACHE: Constraint Logic Programming Solves Aircraft Parking Program, *The Magazine of the Advanced Systems and Software*, Vol. 7, No. 9., September 1991.

[3] B. Goodman and D. Litman. On the interaction between plan recognition and intelligent interfaces, *User Modeling and User-Adapted Interaction*, 2(1-2) : 93-116, 1992.

[4] JJ. Lee, R. McCartney. Dept of Computer Science & Engineering University of Connecticut Storrs, CT. Plan Recognition in Human Computer Interaction.

[5] J. Breuker. EUROHELP:Developing Intelligent Help System, Report on the P280 ESPRIT Project EUROHELP, Lawrence erlbaum associates, publishers, 1990.

[6] M. Bauer , G. Paul. Logic-based Plan Recognition for Intelligent Help System, Current Trends in AI Planning C. Backstrom and E. Sandwall(Eds). IOS Press, 1994.

[7] M. Bauer, S. Biundo, D.degler, J. Kohler, G. Paul. PHI - a logic-based tool for intelligent help systems, In Proceedings of the 10th European Conference on Artificial Intelligence, pages 628-632, 1992.

[8] B. Pierrre, L Bruno, M. Marie-Angie, V. Christiphe, A scheduling Problem Optimization Sovled with Constraint Logic Programming, Artificial Intelligence, Vol. 42, pp. 200 231.

[9] N. Heintze, J. Jaffar, S. Michayov, P. Stuckey and R. Yap, The CLP(R) Programmers Manual Version 1.2, IBM T.J. Watson, 1992

[10] H. A. Kautz. A formal theory of plan recognition, Technical Report 215, University of Rochester, New York, 1987, Dept of Computer Science.

[11] D. J. Litman and J. Allen. A plan recognition model of subdialogues in convertsations, Technical Report 118, University of Rochester, New York, 1984. Dept of Computer Scienece.

[12] M. Genesereth. The role of plans in intelligent training systems. In Intelligent Tutoring Systems, pp.137-155. Academic Press, 1982.

이 훈(李薰)

1996년 2월 : 인하대학교 전자계산공학과(공학사)
1998년 2월 : 인하대학교 대학원 전자계산공학과(공학석사)
1998년 1월~현재 : LG-EDS시스템 공공사업부 사업추진부 근무

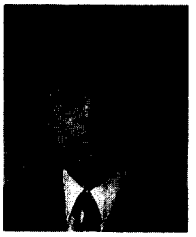
관심분야 : 인공지능, 전문가시스템, 지식기반 스케쥴링, CSP 에이전트

정 종 진(丁鐘枃)

1992년 2월 : 인하대학교 전자계산학과(공학사)
1995년 2월 : 인하대학교 대학원 전자계산학과(공학석사)
1995년 3월~현재 : 인하대학교 대학원 전자계산학과 박사과정 수료
1991년 12월~1992년 6월 : 삼성데이타 시스템 전자 종합관리 본부 컴퓨터 정보 시스템실 정보 시스템 2팀 해외영업 관리 시스템 개발

관심분야 : 지능형 에이전트 시스템, 전문가 시스템, CSP

조 근 식(趙根植)



1982년 : 인하대학교 전자계산학과 졸업 (전산학사)
1985년 : Queens College/CUNY Computer Science 전공 졸업 (전산학석사)
1991년 : City University of New-York Computer Science 전공 졸업 (전산학박사)
1988년 4월~1991년 2월 : Logic Based System Lab. 연구원
1991년 3월~현재 : 인하대학교 전자계산공학과 부교수
1997년 3월~1998년 8월 : 인하대학교 전자계산학과 학과장
1994년 1월~현재 : 한국 전문가시스템학회 학술이사
1996년 1월~1996년 12월 : 한국정보과학회 논문지편집위원
1996년 1월~현재 : 한국정보과학회 인공지능연구회 운영위원
1997년 1월~현재 : 한국 전문가시스템학회 논문지 편집위원장

관심분야 : 인공지능 전문가 시스템, 지식기반 스케쥴링, CSP