

분산 제약조건 만족 특성을 이용한 다중 에이전트 기반 강의 시간표 자동화 시스템 설계

황경순⁰ 전중남 이진명

충북대학교 컴퓨터과학과, 첨단정보기술 연구센터

hks@aicore.chungbuk.ac.kr

Design of a Multiagent-based Lecture-timetabling Automation System using the Properties of Distributed Constraint Satisfaction

Kyoung-Soon Hwang⁰ Joong-Nam Jun Keon-Myung Lee

Dept. of Computer Science, Chungbuk National University and AITrc

요 약

강의시간표 문제(Lecture-timetabling Problem)는 일주일을 기본으로 하는 특정 시간대 별로 학생(Course-class), 교수, 그리고 강의실과 같은 자원에 대한 스케줄링 문제로서, 각각의 자원에 대하여 적절한 조합으로 제약조건들 사이의 충돌을 최소화하여 자원을 배치하는 할당문제이다. 강의시간표 문제는 수천 명 학생들에 대하여 개개인의 시간표를 작성해야 하므로 기하급수적으로 증가하는 탐색공간에 대하여 최악의 경우 탐색 시간이 지수적으로 증가하는 NP-complete Problem이다. 이러한 거대하고 복잡한 강의시간표 문제는 계층적으로 분할하여 기능별로 처리하면서 제약조건을 완화하도록 하는 각 모듈 단위의 에이전트를 구성하므로 좀 더 작고 단순한 문제로 변환될 수 있다. 본 논문에서는 방대한 탐색 공간과 과잉-제약조건(Over-constraint) 문제의 하나인 강의시간표 작성 문제를 분산제약조건 만족 문제 특성을 이용하고 다중 에이전트 구조를 사용하여 해결하는 강의시간표 자동화 시스템 설계를 제안한다.

1. 서 론

에이전트에 관한 여러 가지 정의가 있지만, 이 논문에서는 에이전트를 사용자 대신하여 사용자가 원하는 작업을 수행하는 소프트웨어를 지칭한다. 에이전트 기반 시스템은 시스템 설계 및 구현에 에이전트가 중심적인 개념으로 사용된 시스템을 일컫는다. 이는 객체 지향 방법론에서 객체가 중심이 되는 것과 같은 이치이다. 다중 에이전트(multiagent) 시스템은 상호 작용하는 다수의 에이전트들을 이용하여 설계하고 구현된 시스템으로서, 문제에 대한 서로 다른 시각과 해결방식을 갖고 문제의 다른 측면을 해결할 수 있는 다수의 에이전트로 문제를 해결할 수 있을 때 효과적으로 이용될 수 있다[2]. 제약조건 만족(Constraint Satisfaction) 문제는 모든 제약조건이 만족하는 그것들의 도메인으로부터 각각의 변수들에 대하여 값을 찾는 할당문제이다[3]. 분산 제약조건 만족(Distributed constraint satisfaction) 문제는 제약조건 만족 문제의 변수들이 에이전트 사이에 분산되어 있는 문제이다[3]. 인공지능 기법을 적용하여 작성한 시간표 작성을 위한 연구는 많이 수행되어 왔다 [4,5,6,7,9]. 이 논문에서는 거대하고 복잡한 강의시간표 문제를 계층적으로 분할 처리하여 각각의 모듈 단위에 에이전트를 구성하여 좀 더 작고 단순한 문제로 변환할 수 있는 다중 에이전트 플랫폼에 기반하여, 방대한 탐색 공간 문제와 과잉-제약조건 문제에 대한 해를 찾는 분산 제약조건 만족 문제의 특성을 사용하는 다중 에이전트

기반 강의 시간표 자동화 시스템 설계를 제안한다.

이 논문의 구성은 다음과 같다. 2절에서 제안한 다중 에이전트 기반 강의시간표 자동화 시스템 구조와 특성을 소개하고, 3절에서는 시스템의 구현 내용과 기술을 설명하고, 마지막 4절에서는 이 논문의 결론과 향후 과제에 대하여 기술한다.

2. 시스템 구조와 특성

제안한 시스템은 강의시간표 작성에서 있어서 각 세부 기능을 담당하는 다음과 같은 다수의 에이전트가 서로 상호작용 하면서 제약조건을 만족하는 해를 찾아가는 다중 에이전트들로 구성되어 있다.

2.1 강의실 관리 에이전트

강의실 관리 에이전트는 강의실에 관한 정보를 생성/저장 관리하는 역할을 한다.

2.2 코드 관리 에이전트

개설강좌 자동화 에이전트가 요구하는 개설과목, 강의실, 교수, 그리고 학과코드에 관한 정보를 생성/저장 관리하는 역할을 한다.

2.3 수강신청 관리 에이전트

강의 시간표 자동화 에이전트가 요구하는 학생들의 수강과목신청 정보를 생성/저장 관리하는 역할을 한다.

⁰ 이 논문은 첨단 정보기술 연구센터(AITrc)를 통해서 과학재단의 지원을 받은 것이다.

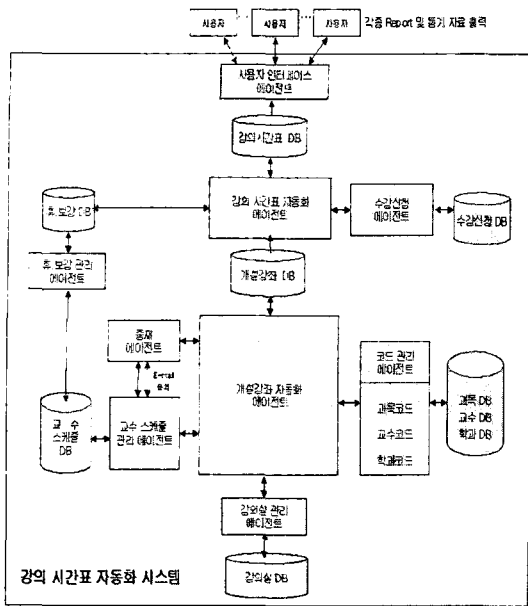


그림 1. 멀티에이전트 기반 강의시간표 자동화 시스템 구성도

2.4 교수 스케줄 관리 에이전트

강의 시간표 자동화 에이전트가 요구하는 교수의 선호시간과 제한 시간, 그리고 휴·보강 스케줄에 대한 정보를 생성/저장 관리하는 역할을 한다. 이때 고려하는 제약조건은 다음과 같다: 선호시간 (교수의 강의가 가능한 요일과 시간), 제한시간(정규강의 시간, 규칙적인 타 대학 강의 및 특강, 본인의 타 대학 수강 시간), 휴·보강시간 (출장, 휴가, 교내·외 행사참석)

2.5 개설강좌 자동화 에이전트

개설강좌 자동화 에이전트는 개설과목, 강의실, 교수의 스케줄, 그리고 분산 제약조건들에 관한 정보를 이용하여 초기 개설강좌를 자동으로 생성한 후 중재 에이전트를 통해 조정과 통합 과정을 걸쳐 강의시간표 자동화 에이전트가 요구하는 정보를 생성/저장 관리하는 역할을 한다.

2.6 중재 에이전트

개설강좌 자동화 에이전트에서 초기 개설강좌에 의해 해결되지 않은 부분에 대하여 중재 에이전트를 두어 조정 가능성이 높은 강의시간을 자동 검색하여 사용자에게 e-mail 문의를 한다. 사용자의 응답으로 개설강좌를 재조정한다.

2.7 강의시간표 자동화 에이전트

학생들의 수강과목과 개설강좌에 대한 정보로 강의 시간표를 자동 생성하고 사용자 인터페이스 에이전트가 필요로 하는 정보를 제공하는 역할을 한다.

2.8 휴·보강관리 에이전트

교수 스케줄 정보로부터 자동 휴·보강 신청서를 생성/저장하고, 휴·보강 정보에 의해 강의 시간 및 강의실 용도에 적합한 강의실을 자동 검색하여 사용자에게 제공하는 역할을 한다.

2.9 사용자 인터페이스 에이전트

사용자 인터페이스 에이전트는 각종 보고서를 사용자가 원하는 형

태로 검색/출력할 수 있도록 제공하는 역할을 한다.

3. 시스템 구현 내용

강의시간표 문제는 탐색과 부분-최적해가 아닌 일반적인 만족을 유도하려는 휴리스틱을 사용한 결합적 최적화 문제이다[7]. 결합적 최적화 문제는 에이전트간에 분산 제약조건 만족 문제이기도하다. 제약조건 만족 문제는 두 종류의 제약 조건으로 분류 할 수 있다. 우선 반드시 지켜져야 되는 강한-제약조건(Strong constraint: SC)과 약간의 유연성을 가진 약한-제약조건(Weak constraint: WC)이다. 이 논문에서 다루고 있는 제약조건들은 다음과 같다.

강한-제약조건(SC)

- C1: 한 강의실에서는 동시에 두 강의를 할 수 없다.
- C2: 한 교수가 동시에 여러 강의실에서 강의를 할 수 없다.(예외 - 교수가 카메라와 비디오 프로젝션(영상강의), 그리고 컴퓨터(Internet 강의)의 도움을 받아 동시에 여러 강의실에서 한 과목에 대하여 강의 할 수 있다)
- C3: 특정 과목들은 겹치지 않는다.
- C4: 시간 배정형식을 지킨다.
- C5: 각각의 강의실은 학생들의 수에 기대되는 충분한 공간을 가지고 있어야 한다.
- C6: 각각의 강의실은 강사의 요구에 대하여 적절한 장비를 갖추고 있어야 한다.

약한-제약조건(WC)

- C7: 교수의 제한시간을 보장한다.
- C8: 교수의 선호시간을 보장한다.
- C9: 강의실 배정은 학생의 최소한 움직임을 보장한다
- C10: 학생들의 식사시간을 보장한다.
- C11: 너무 이른 아침이나 너무 늦은 저녁은 제한한다.
- C12: 교수별 하루에 최대 수업 시간표를 제한한다.

시간표 문제는 종종 방대한 탐색공간 문제와 과잉-제약조건이 발생하는데, 이 경우 다음과 같은 방법으로 점진적으로 해결한다. 1) 제약조건을 계층적으로 두고 약한-제약조건(WC)을 반복 해서 점차 완화시켜 부분-최적해가 아닌 전체적인 해를 찾는다[8]. 2) 변수(variable)에 주석(Annotations)을 적용하는 제약조건 프로그래밍을 사용한다. 여기서 주석은 단지 특정 제약조건에 대하여 각각의 변수에 선호도, 우선 순위, 그리고 가중치 등을 표현한 것이며, 입력 데이터의 일부분으로 주어지고, 변수의 순서를 계산하기 위해서 적용한다[7]. 3) 작업에 흐름을 계층적으로 두고 작업의 단위를 분할하여 [그림 2]와 같이 전처리(Pre-processor)작업을 한다. 이 시스템의 작업 흐름은 다음과 같다

작업 흐름도

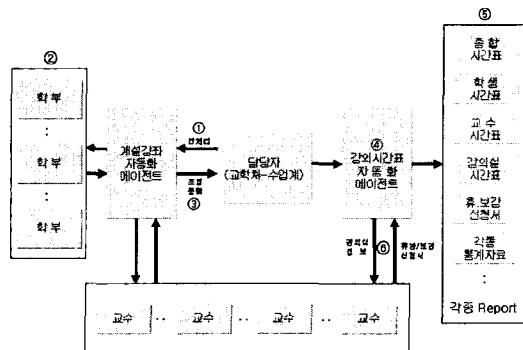


그림 2 작업 흐름도

① 교학처(수업계) 담당자에 의해 전처리(Preprocessor)작업을 한다. 전처리(Pre-processor)작업은 아래와 같다.

- 각종교과생성(교수, 강의실, 개설과목, 학과...)
 우선 순위, 선호도, 기타 제약조건 처리
 교양과목 시간표 생성
 (교양과목을 전처리 하는 이유는 학생의 개인의 사정으로 인해 전공과목과 시간표가 겹치는 경우가 있다. 이때 해당 학생에서 다과목 수강을 허용하기 위한 것이다. 뿐만 아니라 학부/학과단위 작업에서 탐색공간을 축소 할 수 있다.)
- ② 각 학부/학과별 초기 개설강좌를 자동 생성한다.
 - ③ 교학처(수업계) 담당자는 자동 생성된 각 학부/학과 초기 개설 강좌에 대하여 불합한 후 해결되지 않은 부분에 대하여 중재 에이전트를 이용하여 재조정된 후 강의시간표가 요구하는 정보를 자동 생성한다.
 - ④ 자동 생성된 개설강좌 정보와 학생들의 수강 신청서 정보에 의해서 강의 시간표를 자동 생성한다.
 - ⑤ 자동 생성된 강의 시간표에 의해서 각종 보고서를 작성한다.
 - ⑥ 시간표 문제는 주기적으로 발생함에도 불구하고 한해에 두 번만 이벤트가 발생하므로 시스템을 효율적으로 사용하지 못하는 단점을 가지고 있다 이를 보완하는 것으로 교수 스케줄링 정보에 의해 수시 발생하는 휴강/보강 처리를 자동화하므로 보다 더 효율적으로 시스템을 사용할 수 있다.

마지막으로 다중 에이전트간에 분산 제약조건 문제를 해결하기 위해서는 되추적(backtracking) 알고리즘을 적용한다. 되추적의 기본적인 알고리즘은 모든 가능한 해들을 통하여 상태공간(state-space) 그래프를 생성하고 깊이우선이나 넓이우선 탐색을 통해 최적 해를 찾아가는 것으로 더 이상 탐색이 불가능한 경우 탐색을 중단하고 이전 탐색상태로 돌아가는 것이 되추적 알고리즘이다. 상태공간 탐색 기술들로는 A*, Depth-first Branch & Bound 등을 사용하여 충돌을 최소로 하는 해를 찾는다[8]. 해를 찾기 위해서는 탐색이 필수적이다. 이것에 대한 기본적인 개념을 다음과 같다[9].

- ① 문제의 중요한 변수들의 집합으로부터 한 변수를 선택하고, 이 변수에 대한 도메인으로부터 값을 선택하여 그 변수에 값을 할당하는 것이다.
- ② 이 할당 값의 결과를 계산하기 위하여 제약조건 해결 방법으로 사용한다. (되추적 알고리즘은 제약조건 방법에 모순이 있을 때 사용되기 위한 것이다.)
- ③ 이것을 모든 문제의 변수가 값을 갖고 제약조건이 만족할 때까지 반복한다.

여기서 강의에 대한 수업시간과 강의실의 속성은 도메인 변수라고 할 있다. 그리고 도메인 변수에 대한 가능한 값은 자연수로 매핑(mapping)된다[9]. 예를 들어 본 논문에서는 수업시간 배정을 한 후 강의실을 배정하기로 한다. 강의시간표에 대한 해는 S={학년, 과목, 교수, 강의요일, 강의시간, 시수, 강의유형, 인원수, 분반번호, 강의실}의 변수들에 값을 갖는 것이다 (강의유형은 이론, 실습, 세미나, 영상, 그리고 internet강의 등으로 구분된다. 분반번호는 합반, A반, B반, ..., F반으로 구분된다. 강의시간은 50분을 기준으로 하고 시수만큼 계산된다(이론은 1시간을 1학점으로 하고 실습은 2시간을 1학점으로 시수를 계산한다)). 시간표 배정 흐름은 다음과 같다.

- ① 각 과목에 대하여 C8을 가지고 C4에 의해 초기 수업시간 배정을 한다.
- ② 초기 수업시간 배정에 대하여 C2, C3, C11, 그리고 C12 사이의 제약조건 위배가 최소가 되는 수업시간을 탐색하여 재 배정한다.
- ③ 수업시간 배정 후에 C1, C5, C6, 그리고 C9 사이의 제약조건 위배가 최소가 되는 강의실을 탐색하여 배정한다.

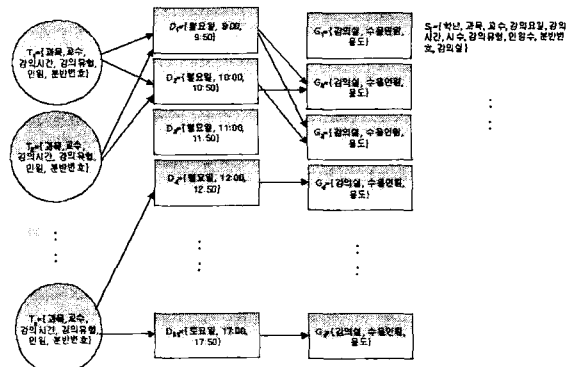


그림3 시간표 배정 흐름도

4. 결론 및 향후 과제

본 논문에서는 과잉-제약조건(over-constraint)과 큰 탐색 공간을 해결하기 위하여 분산 제약조건 만족 문제의 제약조건 완화(constraint relaxing in distributed constraint satisfaction problem)와 변수에 주석을 적용하는 제약조건 프로그래밍(constraint programming), 그리고 방대한 탐색공간을 완전히 수행할 수 없으므로 인간의 경험적 지식을 이용한 휴리스틱 탐색 등을 이용하여 다중 에이전트 기반 강의시간표 자동화 시스템 설계를 제안하였다. 향후 시간·공간 제약조건을 이용하여 자원의 분배 및 활용을 극대화하여 한정된 자원을 효율적으로 배치하기 위한 보다 지능적인 멀티에이전트 기반 강의시간표 자동화 시스템 구현에 대한 연구가 필요하다.

참고문헌

- [1] 최중민, "에이전트의 개요와 연구방향", 정보과학회지, 15권 3호, pp 7-16, 1997.
- [2] 이재호, "에이전트 시스템의 연구 및 개발 동향", 정보과학회지, 15권 제5호 통권 제 132호 ISSN 1229-6821, pp.4-9, 2000년 5월.
- [3] Makoto Yokoo, Toru Ishida, Edmund H. Durfee, Kazuhiro Kuwabara, "Distributed Constraint Satisfaction for Formalizing Distributed Problem Solving", Proc. of the 12th International Conference on Distributed Computing Systems (ICDCS-92), pp.614-621, 1992.
- [4] 정태중, "종합 강의 시간표 작성 시스템 개발 연구 개발 보고서", 경희대학교 전자계산공학과, 1996
- [5] 정용성, 김태식, "유전자 알고리즘을 이용한 강의실 배치 문제", 정보과학회 '98 봄 학술대회 발표 논문집(B), 25권 1호, pp25-260, 1998년 4월.
- [6] 김분홍, 이광호, "목포대학교 시간표 작성 프로그램 개발을 위한 기초시법", 정보과학회 '99 봄 학술 발표 논문집(B), 26권 1호, pp25-260, 1999년 4월.
- [7] Hana Rudov and Ludek Matyska, "Constraint-based timetabling with student schedules", In Edmund Burke and Wilhelm Erben, editors, PATAT 2000 - Proc of the 3rd international conference on the Practice And Theory of Automated Timetabling, pp.109-123, 2000. 6.
- [8] Makoto Yokoo, Hikaridai, Seika-cho, "Constraint Relaxation in Distributed Constraint Satisfaction Problem", Proc of the 5th International Conference on Tools with Artificial Intelligence, pp.56-63, 1993.
- [9] Hans-Joachim Goltz, Georg Kehler, and Dirk Matzke, "Constraint-based timetabling for universities " In Proceedings INAP'98, 11th International Conference on Applications of Prolog, pp.75-80, 1998.