

강의 시간표 최적화를 위한 제약 프로그래밍 모델

김춘식[○], 황준하^{*}

^{○*}금오공과대학교 컴퓨터공학과

e-mail: {20090350, jhhwang}@kumoh.ac.kr^{*}

A Constraint Programming Model for Lecture Timetable Optimization

Chun-Sik Kim[○], Junha Hwang^{*}

^{○*}Dept. of Computer Engineering, Kumoh National Institute of Technology

● 요약 ●

본 논문에서는 강의 시간표 최적화를 위한 제약 프로그래밍의 적용 방안을 제시한다. 제약 프로그래밍은 제약 만족 문제를 해결하기 위한 기법으로 대상 문제를 결정 변수, 도메인, 제약조건으로 표현한다. 본 논문에서는 시간표 작성 최적화 문제의 결정 변수로 강의실, 요일, 교시를 사용하였으며, 추가로 요일과 교시를 결합한 변수를 사용함으로써 보다 쉽게 제약 조건을 표현할 수 있도록 하였다. 또한 제약 프로그래밍에 의해 도출된 초기해를 또 다시 제약 프로그래밍을 통해 반복적으로 개선함으로써 더 좋은 강의 시간표를 작성할 수 있도록 하였다. 특정 학과의 강의 시간표 문제를 대상으로 한 실험 결과, 본 논문에서 제안한 방법을 통해 보다 빠른 시간 내에 초기해를 도출할 수 있을 뿐 아니라 최종적으로 더 좋은 해의 도출이 가능함을 확인하였다.

키워드: 제약 프로그래밍(constraint programming), 시간표(timetable)

I. Introduction

강의 시간표 작성 문제는 다양한 필수 제약조건과 선호 제약 조건을 포함하는 제약 만족 최적화 문제의 일종이다. 지금까지 이 문제를 해결하기 위해 정수 계획법과 제약 프로그래밍 등 다양한 최적화 기법들이 동원되어 왔다[1, 2]. 그러나 대상 문제마다 요구하는 제약조건 및 목적함수가 서로 다르기 때문에 동일한 기법을 적용한다 하더라도 기존 방법을 그대로 적용하기는 어렵다. 본 논문에서는 강의 시간표 작성 문제를 해결하기 위한 제약 프로그래밍의 적용 방안을 제시한다. 기본적으로 각 수업은 강의실, 요일, 교시로 표현될 수 있지만, 본 논문에서는 요일과 교시를 결합한 변수를 추가로 사용한다. 이를 통해 제약조건 표현 시 보다 쉽게 제약조건들을 표현할 수 있도록 하였다. 뿐만 아니라 실험 결과에 의하면, 추가 변수를 사용한 경우 최적화 성능 또한 향상될 수 있음을 확인할 수 있었다.

II. The Lecture Timetabling Problem

일반적으로 강의 시간표 작성 문제의 제약조건은 반드시 지켜야 하는 필수 제약조건과 반드시 지킬 필요는 없지만 지켜지면 좋은 선호 제약조건으로 나뉘며, 선호 제약조건은 최적화 문제의 목적함수로 다루어진다. 본 논문의 대상 문제에서 제약 조건들은 참고문헌 [3]과 같이 9개의 필수 제약조건(C1~C9)과 1개의 선호 제약조건(C10)으로 구성된다. 단, [3]과는 달리 C4 제약 조건이 본 논문에서는 “동일한 학년의 교과목들 사이에는 한 교과목의 각 분반에 대해 겹치지 않는 분반이 하나 이상 존재해야 한다”로 변경되었다. 그리고 [3]에서는 선호 제약조건을 고려하지 않은 제약 만족 문제를 대상으로 하였지만, 본 논문에서는 이를 고려하여 제약 만족 최적화 문제를 대상으로 하였다.

III. Constraint Programming Model

본 논문에서는 대상 문제를 기본적으로 강의실, 요일, 교시의 3가지 결정변수로 표현하며, 추가로 요일과 교시를 결합한 새로운 변수를 사용한다. [그림 1]과 같이 각 과목-분반-수업시간에 대해 최종적으로 강의실(Room)과 요일(Week) 그리고 교시(Period)가 결정되어야 한다. 이때 WeekPeriod는 한 주 내의 모든 시간을 도메인으로 갖는 결정 변수로 Week와 Period가 결정되면 대응되는 WeekPeriod의 값 또한 결정되며 역 또한 성립한다. 제약조건에 따라서는 Week나 Period 또는 WeekPeriod 변수를 선별적으로 사용하여 표현할 수 있다. 예를 들어, 앞서 설명한 C4 제약조건의 경우 WeekPeriod 변수를 사용하는 것이 보다 편리하다.

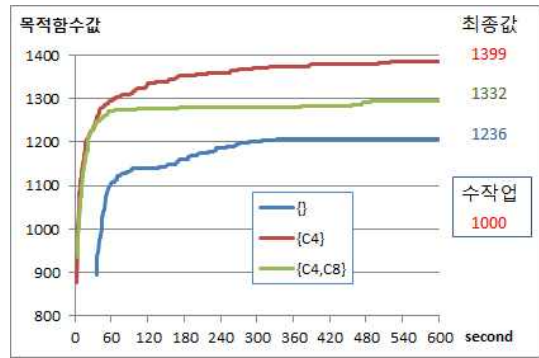
	과목 a			과목 b-1분반			b-2분반		과목 c-1분반		c-2분반	
	1	2	3	4	5	6	7	8	9	10	11	
Room	3	3	3	1	1	1	1	5	5	5	5	
Week	2	2	4	1	1	3	3	4	4	3	3	
Period	5	6	2	3	4	5	6	7	8	3	4	
WeekPeriod	23	24	38	12	13	32	33	43	44	30	31	
	교수 A						교수 B					

[그림 1] 변수의 표현 예

IV. Experimental Results

제약 프로그래밍 개발 도구로는 IBM ILOG CP Optimizer를 사용하였으며, 각 실험 당 수행시간은 2시간으로 설정하였다. 실험 데이터는 총 27과목 65분반, 192시간으로 구성되어 있으며, 13명의 정교수와 6명의 시간강사가 강의를 담당한다. 그리고 강의실 3개와 실습실 4개로 구성되어 있다.

[그림 2]는 실험 결과를 보여주고 있으며 이해를 돕기 위해 그래프는 600초까지만 표현하였고 최종 결과값은 별도로 표시하였다. 목적함수 값은 선호 제약조건인 선호 시간에 대한 점수의 합으로 선호하는 수업 시간에 더 큰 점수를 부여하여 그 합을 최대화하도록 하였다. {}는 WeekPeriod 변수를 사용하지 않았음을 의미하고, {C4}와 {C4,C8}은 각각 해당 제약조건을 표현하기 위해 WeekPeriod 변수를 사용했음을 의미한다. {C4}와 {C4,C8}의 경우 WeekPeriod 변수를 사용하지 않았을 때보다 초기해도 더 빨리 도출되고 최종 결과 또한 더 좋을 수 있다. 수작업의 경우 며칠에 걸쳐 작성되었는데, 목적함수값이 1,000으로 확인되었다. 본 논문에서 제시한 기법이 수작업에 비해 훨씬 빨리 더 좋은 해를 도출할 수 있음을 알 수 있다.



[그림 2] 기존 방법과 결합 변수 적용 결과

V. Conclusions

본 논문에서는 강의 시간표 최적화를 위한 제약 프로그래밍의 적용 방안으로 기존 결정 변수들 중 일부를 결합한 변수를 추가하여 활용하는 방법을 제시하였다. 이를 통해 보다 쉽게 제약 조건을 표현할 수 있었으며, 제약 프로그래밍을 통한 반복적 개선 시 기존 방법보다 더 좋은 결과를 도출할 수 있음을 확인하였다. 추가 변수를 통한 성능의 향상은 제약 프로그래밍에 있어서 변수 순서나 제약 전과와 관련이 있을 것으로 추정되나 정확한 이유를 파악하기는 어려웠다. 따라서 향후 이와 관련된 추가 실험 및 분석이 필요할 것으로 사료된다. 아울러 제약 프로그래밍을 통한 반복적 개선 방법을 개선하기 위해 다른 최적화 기법과 결합하는 방안에 대해서도 연구할 예정이다.

References

- [1] S. Daskalaki, and T. Birbas, "Efficient Solutions for a University Timetabling Problem Through Integer Programming," European Journal of Operational Research, Vol. 160, No. 1, pp.106-120, Jan. 2005.
- [2] L. Zhang, and S. Lau, "Constructing University Timetable Using Constraint Satisfaction Programming Approach," Proceedings of CIMCA-IAWTIC'05, Vol. 2, pp.55-60, Nov. 2005.
- [3] C. S. Kim, J. Hwang, G. Yoo, H. Lee, and J. Yoon, "Automatic Generation of Lecture Timetable Using Constraint Programming," Communications of the Korean Institute of Information Scientists and Engineers, Vol. 33, No. 12, pp.137-140, Dec. 2015.