

Tabu Search를 이용한 고교 시간표 자동 작성 개발을 위한 연구

김 광 석* 정 태충(jungjuks@chollian.net, tcchung@nms.kyunghee.ac.kr)

경희대학교 전자정보학부

A tabu search For High-School Timetabling Problems

Kim KwangSeok, Chung TaeChoong

Department of Computer Science, Kyunghee University

요 약

고등학교 시간표 작성 문제는 학생, 교사, 학교, 일반 교실과 특별 교실 자원등 복잡한 제약 조건과 요구사항이 있기 때문에 단순한 방법으로는 해결하기가 어렵다. 학생들은 다양한 과목을 선택할 수 있기를 바라고 교사들은 자신이 원하는 시간에 수업하기를 바란다. 또한, 학교에서는 효율적인 인적 자원 활용과 일반 교실 및 특별실의 사용으로 학교의 장학 능력과 경쟁력의 고양을 원한다. 따라서, 고등학교 수업 시간표를 작성하는데 있어서의 다양한 문제를 최적의 시간표를 최적 탐색 문제로 정의 하고, 이 문제를 휴리스틱 접근법인 Tabu Search 방법을 이용하여 해결하고자 한다.

1. 서론

과거 고등학교 시간표 작성 문제는 1961년 Robert의 "Using Machine To Make the High-School Schedule"[1]이라는 논문에서 수업 시간표의 작성 문제를 컴퓨터로 풀고자 하는 시도가 있을 후 더 많은 노력을 해 왔으나 최근에 조사된 논문에 의하면 현재 사용 중인 시간표 관련 프로그램은 불과 10여개 정도이다[2]. 아직도 대부분의 학교에서는 3-4명이 한 조가 되어 3-4일 정도 수작업으로 시간표를 짜고 있다. 그리고 시간표 작성 후에도 결강 및 보강 계획을 세우느라 1년내내 전담 교사를 두어 처리하고 있다.

2002년부터 시행될 7차교육과정에서 교육부는 수요자 중심의 학생 교과 선택제와 수준별 교육 과정 운영으로 방향이 전환됨에 따라 일선 학교에서는 컴퓨터를 이용한 시간표의 작성에 필요성을 절감하면서도 그 개발 성공 사례가 매우 적은 것은 그만큼 시간표 작성의 전산화가 어려운 문제임을 반증 해 준다.

이 어려움의 대표적인 문제는 좋은 수업 시간표의 정의가 어렵다는 것 때문이다. 수업 시간표의 주체는 학생, 교사, 학교이다. 좋은 시간표란 이 3자의 입장이 모두 만족되는 시간표를 말하는 것이라 정의할 수 있다.

따라서 3자의 입장을 모두 고려하여 시간표를 작성하는 문제는 모든 개인의 시간표에 관련된 개인적 선호까지 자료화해야 한다는 부담을 갖는다. 이러한 문제의 컴퓨터로의 접근은 대표적으로 제약 조건을 규칙의 형태로 표현하고 별점의 개념을 도입해 자료를 수치화 하고, 최적화 방법을 통해 문제를 해결하고자 한다. 그러나, 고전적인 휴리스틱 탐색 방법은 그 탐색 시간과 공간의 지수적인 증가로 인해 적용이 매우 제한적이다. 이러한 문제를 해결하기 위하여 최적화 알고리즘으로 타부 탐색을 이용해 최적해를 찾는다.

2. 시간표 자원의 할당 문제

2.1 문제의 정의

고등학교 시간표 작성 문제는 주어진 자원에 대해 각각의 선호도나 제약 조건을 기반으로 최적의 시간표를 작성하는 것이다. 먼저 가능한시간표 시간 자원의 범위를 선정해 시간표 초기 영역을 설정하는 문제는, 정해진 시간의 범위 내에서 문제를 해결할 수 있는 것인지에 대한 판단을 수행하는 작업이라 볼 수 있다. 이후 교실 자원 배정과, 각 과목의 시간 자원 배정 문제 해결을 통해 적합한 시간표를 생성하는 것이다.

2.2 시간표의 제약 조건

대부분의 시간표 관련 연구에서는 제약 조건은 반드시 지켜야 하는 강한 조건(Hard Constraint : HC)과 최대한 지키면 좋은 제약 조건(Soft Constraint : SC)으로 구분한다[2][3]. HC는 교사의 수업 시간이 동일한 시간에 편성된다거나 하는, 시간표로서의 기능을 상실하게 되는 제약 조건이며, SC는 교사의 선호 시간을 최대한 반영한다는 등의 제약 조건을 말한다.

$X_{ij} = 1$ 만약 수업 i 가 시간 j 에 배정될 수 있다.

$X_{ij} = 0$ 충돌이 발생한다.

따라서 최종적으로 시간표 작성의 목적 함수는

$$F(x) = \sum_{i=1}^n \sum_{j \in J_i} C_{ij} X_{ij}$$

여기서 J_i 는 수업 i 가 배정될 수 있는 시간의 범위이다. C_{ij} 는 강좌 i 가 j 시간에 배정되었을 때 받게 되는 벌점을 말한다. 만약 최상의 조건의 자리라면 이 변수의 값은 0이 된다.

(1) 강한 제약 조건(Hard Constraint)

- 특정과목 수업 시간 고정(이동식 수업 보장).
- 특정 요일, 특정 시간 수업 불가(부장 교사회의-화요일 5교시).
- 하루 수업 시수 고정.
- 시간, 장소 고정 과목은 반드시 지정.
- 명시 과목 중복 불가.
- 특정 과목은 특정 시간대에 배정.
- 교실 용도별 수업 배정 원칙 준수
- 연속 수업의 경우 동일한 교실을 배정.

(2) 약한 제약 조건(Soft Constraint)

- 교사는 자신이 원하는 시간에 수업.
- 특정 교사끼리 중복 불가(정보처리실,그래픽실,멀티미디어실,종합실습실,문서작성실,어학실,과학실등).
- 한 교사가 여러 과목을 수업할 경우 같은 날에 동일 과목 수업.
- 한 교사가 연속적으로 3시간 이상 수업불가.

3. 문제의 해결

3.1 휴리스틱에 기반한 문제 해결

휴리스틱 기술(간단히 휴리스틱)은 최적성의 보장이나 가능성의 문제와는 별개로 적당한 계산의 비용에 좋은 해(즉, 근사 최적)를 찾는 방법이다. 불행히도, 특정한 휴리스틱 해가 최적화에 얼마나 도달해 있는지를 알 수 없다. 비록 이 정의에도 나타나 있듯이 몇몇 중대한 문제가 있음에도 불구하고, 휴리스틱 기술은 실제로 최적의 해

를 찾아낸다는 점이 중요한 것이다.

복잡한 최적화 문제를 풀기 위한 휴리스틱 기술의 방법은 최적 우선 탐색(Best-First Search), 시뮬레이티드 어닐링(Simulated Annealing)[4], 타부 탐색(Tabu Search), 유전자 알고리즘(Genetic Algorithm) 등이 있다.

그 중에서도 본 논문에서는 타부 탐색(Tabu Search) 통해 고교 시간표 최적화 문제를 해결하고자 한다.

3.2 타부 탐색

국부 최적을 벗어나기 위해 이웃해로의 전이라는 측면에서 이 방법은 시뮬레이티드 어닐링방법과 유사하다. 그러나 무작위 선택의 방법 대신에 타부 탐색에서는 “타부 이동”의 항목을 유지하는 방법을 사용한다. 이것은 전에 탐색했던 해를 표현하고 알고리즘이 다른 해를 찾을 수 있도록 해준다. 이 제약 조건은 알고리즘이 후퇴 없이 해를 개선하는 방향으로 진행하도록 한다. 비록 그것이 더 나쁜 해일 지라도 새로운 이동은 적용된다. 탐색은 초기 해 s_0 가 이웃 해 $N(s)$ 을 탐색함으로써 진행된다. 이때, 타부 이동의 항목은 탐색이 사이클에 빠지는 것을 방지하는 목적을 갖으며 탐색의 전 과정을 기억할 수 없으므로, 최근 k 개만큼의 목록만을 유지하는 방법을 사용한다.

타부 탐색은 탐색과정이 국부 최적에 빠지는 것을 방지하고 효과적으로 이것을 극복하기 위해 aspiration 함수를 사용한다. 이 함수는 일반적으로 얼마만큼의 범위의 해를 허용할 것인지를 결정하는 함수로서 비록 현재의 해가 최선의 해보다 좋지 않더라도 이 해를 채택하는 방법을 제공한다.

4. 타부 탐색의 적용(최적해 찾기)

Andrea Schaerf[5]의 시간표 작성에 적용된 알고리즘을 우리 고교 시간표 작성에 적용해 보고자 한다.

초기 해는 각 교사에 대한 수업시간을 랜덤 하게 구하거나 interactive timetabling의 경우에 이전 실행 값에 의해 얻은 요구사항에 의해 구한다. 그런 후 RNA가 주어진 반복횟수(RNAmax)에 개선이 없는 한 무작위로 시간을 만들기 작업을 시작하였다. 이 시점에서 TS는 시작하고 개선(TSmax)이 없는 주어진 반복횟수를 만들 때까지 시행한다. 모든 과정(RNA + TS)은 최적 해를 발견할 때까지 반복하고 주어진 시간(Cycles)동안 개선되지 않을 때 멈춘다. RNA는 2가지 다른 목적으로 사용된다.

우선 TS를 위한 초기 해를 만든다. 랜덤한 해로부터 시작하는 TS의 사용은 시간을 너무 많이 사용하고, RNA가 좋은 초기해(일반적으로 아직까지는 실행 불가능한

것이 조금은 포함된다)를 만들기 위해 빠른 탐색 방법을 대신해서 나타낸다.

그 다음으로 TS가 주어진 반복횟수에 대한 개선이 없는 후에 발견한 최적해상에 RNA를 실행한다.

한편 double move를 사용하는 RNA는 TS가 발견할 수 없었던 최적해를 찾는다. 다른 한편으로, double move를 사용하는 RNA는 비록 해를 개선하지 못한다 할지라도 실제 다른 방법으로 수정을 한다. 더욱이 TS를 시작하기 전에 해를 “질질 끌기”하는 것을 개선하기 위해 TS를 다시 시작한다. 아래의 알고리즘은 underlying이라고 하는데, 이 underlying은 TS가 주어진 반복횟수동안에 좋은 해를 찾지 못한 후에 해를 유용하게 개선하기 때문에 TS는 다른 방향에서 시작할 수 있다.

이 알고리즘은 그림 2에서 보여준다.

5. 결론

시간표 자원의 핵심은 시간이다. 시간은 저장되는 자원이 아니며, 따라서 재활용 될 수 없다. 시간표 스케줄링의 핵심은 최대한 효율을 극대화하면서도 시간표를 사용하는 모든 이가 만족을 느끼는 것이다. 이 두 가지 목적은 상호 충돌하는 경향을 갖는다. 고교 시간표 문제를 위한 Tabu Search 알고리즘은 다양한 형태의 학교에 좋은 결과를 가져올 것이다. 그리고 목적 함수에 대한 다양한 가중치를 부여해준다. 또한 시간표를 수작업하는 것보다 더 좋은 결과를 가져다 준다. 그리고, 최근 교육 부에서 타 학교의 교과 과정을 이수할 수 있는 교과 선택제를 도입하여 학교간 교과를 공유해서 교과를 선택할 수 있도록 하였기 때문에 시간표 문제의 범위가 학교간의 문제로 확대됨에 따라 고교 시간표의 자동작성 시스템 개발은 더욱 중요한 의미를 가지게 될 것이다. 앞으로 학생, 교사, 학교 모두 충분히 충족시킬 수 있는 더 다양한 제약조건을 고려한 고교 시간표 자동작성 시스템 개발을 모색해야겠다.

참고문헌

[1] Abramson, D. (1991). "Constructing school timetables using simulated annealing: sequential and parallel algorithms". Management Science, 37,98-113
 [2] Ahn, J. I, and Chung, T. C.(1997), "Graph coloring algorithm to Make Timetable for Lessons Requiring Multiple Slots", Proceedings of the 2nd international Conference on the Practice and Theory of Automated Timetabling, p281-284
 [3] Azevedi, F. and Barrahoa P., "Timetabling in Constraint Logic programming", Proceedings of World

```

Algorithm High School TimeTabling
begin
  GetInitialSolution: (* generated at random or from input file *)
  InitializeSearchVariables:
  for c := 1 to Cycles
  do begin
    NoChanges :=0;
    while (NoChanges < RNAMax)(* make the RNA phase *)
    do begin
      NoChanges := NoChanges +1;
      DoubleMove :=RandomDoubleMove;
      if delta(DoubleMove) ≤ 0
      then begin
        UpdateCurrentTimeTable(DoubleMove); (*make the move*)
        if delta(Doublemove)<0 then NoChanges :=0
      end
    end;
    NoChanges :=0;
    while (NoChanges < TSmax) (*make the TS phase*)
    do begin
      NoChanges :=NoChanges +1;
      BestMove :=AtomicRandomMove;
      BestDelta :=delta(BestMove,DynamicWeights);
      forall Move in LegalAtomicNeighborhood_(CurrentTimeTable)
      begin if delta(Move,DynamicWeights)< BestDelta
        and(not Tabu(Move) or Aspiration(Move))
        and(delta(Move,DynamicWeights)<0
          or not SemiIllegal(Move))
        then begin
          BestMove :=Move;
          BestDelta :=delta_(Move,DynamicWeights)
        end
      end;
      UpdateCurrentTimeTable(BestMove); (*make the move*)
      UpdateTabuList(BestMove);
      if IsBest(CurrentTimeTable)
      then
        BestTimeTable := CurrentTimeTable;
      if UpdateIteration
      then UpdateDynamicWeights
    end;
    CurrentTimeTable := BestTimeTable
  end
end
    
```

<그림2> Timetabling 알고리즘

Congress on Expert System 94

[4] Joanthan Thompson and Kathryn A Dowsland,(1996) "General Cooling Schedules for a Simulated Annealing Bases Timetabling System", Practice and Theory of Automated Timetabling, Lecture Notes in Computer Scienced 1153, Springer Verlag, p345-363
 [5] A. Schaerf(1996), Tabu search techniques for large Chigh-school timetabling problems