

철도차량소요계획에 대한 집합분할문제 접근에서 후보운용생성을 위한 열거알고리즘

An Enumeration Algorithm for Generating the Candidate Routings in a Set Partitioning Problem Approach to the Rolling Stock Requirement Plan

김성호¹ · 김동희² · 최태성³

Seong-Ho Kim · Dong-Hee Kim · Tae-Sung Choi

Abstract

A routing is the path that an actual trainset follows as it moves from one train to another train in a timetable. The number of routings is equivalent to the number of trainsets required to cover the timetable. The primary factors of rolling stock requirement plan include evaluating the minimum number of routings. This can be formulated as a set partitioning problem and solved using enumeration method or column generation method. In this paper we presents an enumeration algorithm which is useful to implement the enumeration method for the rolling stock requirement plan.

Keywords : Rolling Stock Requirement Plan(철도차량소요계획), Set Partitioning Problem(집합분할문제), Enumeration Algorithm(열거알고리즘)

1. 서론

열차스케줄설계는 철도운영기관이 해결해야할 문제중 가장 복잡한 문제이며 철도운영기관의 여러 측면에 큰 영향을 미치는 중요한 문제라고 할 수 있다. 열차스케줄은 철도운영기관이 판매하는 교통서비스의 목록으로 각 열차의 출발역, 도착역, 출발시각, 도착시각이 포함된다. Table 1은 전형적인 열차스케줄의 일부분을 나타낸 것이다. 출발시각 및 도착시각을 결정하고 나면 열차의 좌석수를 결정해야 한다. 일반적으로 이 문제는 몇 가지 편성유형(fleet type)을 설정해 두고 이를 열차에 할당하는 방식으로 접근하며 이를 편성유형할당문제(fleet assignment problem)라고 한다.

각 열차에 편성유형이 할당되면 철도운영기관이 보유하고 있는 편성유형별 차량數로 주어진 열차스케줄을 실행하는 것이 가능한가를 검토하기 위한 철도차량소요계획(rolling stock requirement plan)을 수립해야 한다.

철도차량소요계획에 관한 기존 연구는 다상품네트워크 흐름문제(multicommodity network flow problem)로 접근한 연구(Booler 1980; Wright 1989; Forbes, Holt and Watts 1991; Ziarati, et al. 1997)와 집합분할문제(set partitioning problem)로 접근한 연구(Ben-Khedher et al. 1998)의 두 가지로 대별할 수 있다. Ben-Khedher et al.(1998)은 철도차량소요계획에 관한 집합분할문제접근법의 최초 보고사례라 할 수 있다. Ben-Khedher et al.(1998)의 모태가 된 연구들(예를 들면 Deboo(1973), Simpson(1969) 등)이 AGIFORS에 발표된 바 있으나 이는 철도분야에서의 차량소요계획 수립을 위한 연구가 아닌 항공분야에서 항공기소요계획 수립을 위한 것이다. 한편 Ziarati et al(1977)는 캐나다의 CN North America의 현실자료를 사용하여 차량소요계획에 관한 해를 도출하였다. 특히 Ben-Khedher et al.(1998)은 프랑스 SNCF의 스케줄 최적화 시스템인 RailPlus에 반영되어 실무에서 활용되고 있다.

철도차량소요계획을 수립하기 위한 집합분할문제 접근법에서의 해법으로는 명시적 열거법(explicit enumeration method)과 열생성기법(column generation method)이 사용될 수 있다. 본 논문에서는 명시적 열거법으로 철도차량소

1 정회원, 인하대학교 경영대학 경영학부 조교수
2 정회원, 한국철도기술연구원 운영시스템연구그룹
3 비회원, 인하대학교 경영대학 경영학부 교수

Table 1. 열차스케줄의 예

열차번호	출발역	도착역	출발시간	도착시간
101	서울	부산	06:00	08:40
102	부산	서울	06:00	08:46
103	서울	부산	06:30	09:16
106	부산	서울	06:30	09:10
⋮	⋮	⋮	⋮	⋮

요계획을 수립할 경우 유용하게 사용될 수 있는 열거알고리즘(enumeration algorithm)을 제시하고자 한다.

본 논문의 구성은 다음과 같다. 제 1절에서는 서론으로서 철도차량소요계획에 관한 기존의 접근방법을 소개하였다. 제 2절에서 철도차량소요계획에 대한 집합분할문제 접근법에 관하여 문제의 정의 및 해법을 구체적인 예를 들어 설명한다. 제 3절에서 열거알고리즘을 의사코드로 제시하고 제 4절에서 계산실험의 결과를 설명한다. 제 5절에서 결론을 맺는다.

2. 철도차량소요계획에 대한 집합분할문제 접근법

2.1 문제의 정의

철도차량소요계획(rolling stock requirement plan)은 열차스케줄에 포함되어 있는 모든 열차에 견인서비스를 제공하기 위해 필요한 최소 차량數 또는 최소비용 차량수를 파악하는 것이다. 철도차량소요계획을 수립하기 위해 필요한 입력자료는 열차시각표에 포함된 각 열차의 속성자료 즉 출발역, 도착역, 출발시간, 도착시간 등이며 출력자료는 열차시각표를 구성하는 모든 열차에 견인서비스를 제공하는 최소 또는 최소비용의 차량수가 된다.

철도차량소요계획을 수립하기 위한 집합분할문제(set partitioning problem)는 다음과 같은 이가정수계획법모형(SPP)로 나타낼 수 있다.

(SPP)

$$\text{Minimize } \sum_{j=1}^n c_j x_j \quad (1)$$

$$\text{subject to } \sum_{j=1}^n a_{ij} x_j = 1, \quad \text{for } i=1, 2, \dots, m \quad (2)$$

$$x_j = 0 \text{ or } 1, \quad \text{for } j=1, 2, \dots, n \quad (3)$$

여기서 i : 열차를 나타내는 첨자
 m : 열차의 수

j : 후보운용(candidate routing)을 나타내는 첨자

n : 후보운용의 수

x_j : j 번째 후보운용의 선택여부를 나타내는 결정변수

c_j : j 번째 후보운용의 비용계수

a_{ij} : i 번째 열차가 j 번째 후보운용에 포함되는가의 여부를 나타내는 계수

$a_{ij}=1$: i 번째 열차가 j 번째 후보운용에 포함되어 있음을 나타냄

$a_{ij}=0$: i 번째 열차가 j 번째 후보운용에 포함되어 있지 않음을 나타냄

(SPP)의 제약조건은 후보운용(candidate routing)으로 구성되는데, 여기서 운용(routing)은 물리적인 철도차량 1편성이 견인력을 제공할 대상이 되는 열차들의 집합을 의미한다. 그리고 후보(candidate)는 운용이 제약조건을 만족하고 있음을 나타낸다. (SPP)의 최적해로 선택된 후보운용의 개수는 철도차량소요계획의 최종결과인 소요 차량수가 된다.

(SPP)를 구성하는 후보운용이 충족시켜야 할 기본제약조건은 후보운용을 구성하는 열차들 중에서 이웃한 두 열차가 ① 후행열차의 출발역과 선행열차의 도착역이 일치해야 하며, ② 후행열차의 출발시간이 선행열차의 도착시간 이후이어야 한다는 것이다. 기본제약조건만을 고려한 후보운용으로 (SPP)를 구성하여 구한 소요차량수는 철도차량소요계획을 위한 하한값의 역할을 할 수 있다. 차량의 유지보수 제약조건을 추가로 고려하면 소요차량수는 기본제약조건만을 고려한 경우보다 증가할 것이다.

2.2 두 가지 해법

철도차량소요계획에 대한 집합분할문제 접근법의 해법에는 명시적열거법(explicit enumeration method)과 열생성기법(column generation technique)이 있다. 명시적 열거법은 제약조건을 만족하는 모든 후보운용을 명시적으로 평가한 후 최적 후보운용집합을 선택하는 방법이다. 즉 제약조건을 만족하는 후보운용을 명시적으로 모두 열거하고 이들로 (SPP)를 구성하여 최적 운용집합을 선택한다. 이 방법은 운용이 만족시켜야 할 제약조건이 다양하고 복잡하며 후보운용의 수가 비교적 적을 때 유용하다.

열생성기법은 제약조건을 만족하는 모든 후보운용을 명시적으로 평가한 후 최적 후보운용집합을 선택하는 방법이다. 열생성기법은 마스터문제(master problem)와 하위문제(subproblem)의 해를 구하는 과정을 번갈아가며 반복적으로

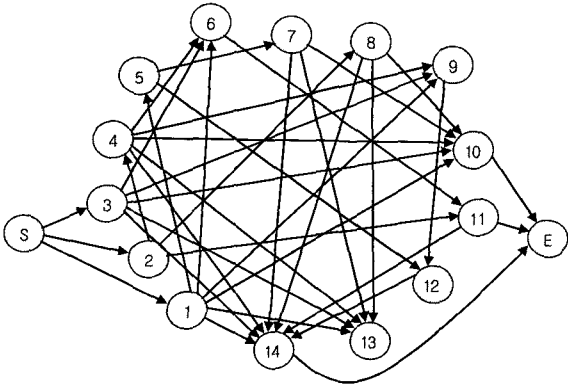


Fig. 3. 후보운용을 생성하기 위한 네트워크

영구꼬리표알고리즘이나 제약프로그래밍은 Fig. 3과 같은 네트워크에 부여되는 추가제약조건을 융통성있게 반영할 수 있으나 모든 형태의 제약조건을 자유롭게 반영하지는 못한다. 영구꼬리표알고리즘이나 제약프로그래밍은 Fig. 3과 같은 네트워크에서 시작노드 S로부터 종료노드 E로의 경로를 따라 아크비용(arc cost)의 누적값을 일정 범위로 제한하는 형태의 제약조건은 반영할 수 있다. 그러나 개별 노드 고유의 제약조건(예를 들면 Fig. 3에서 11번 노드는 14번 노드로 연결할 수 없고 반드시 종료노드 E로 연결되어야 한다)은 반영하기 어렵다.

후보운용이 만족시켜야 할 제약조건으로 영구꼬리표알고리즘이나 제약프로그래밍에서 반영하기 어려운 형태의 제약조건을 고려할 경우 명시적열거법이 철도차량소요계획을 수립함에 있어서 상대적으로 효과적일 수 있다.

3. 후보운용생성을 위한 열거알고리즘

철도차량소요계획에 대한 집합분할문제 접근법의 해법으로 명시적열거법을 사용할 경우 후보운용을 생성하기 위한 열거알고리즘이 필수적이다. 여기서는 열차시각표를 입력 자료로 하여 기본 제약조건을 만족하는 모든 후보운용을 생성하는 열거알고리즘 generateRotation을 C++ 의사코드(C++ pseudocode) (Neapolitan and Naimipour 2004)로 제시한다.

```
void generateRotation (set_of_train ST,
                      stack_of_node SC );
{
    bool c, c1, c2
    node n1, n2
    stack_of_node SN, SC
```

```
while ( ! empty(SN) ) {
    n1: = last(SN);
    remove(SN, last(SN));
    f1: = last(n1.path);
    for each t2 ∈ ST
        c1: = f1.endStation == t2.startStation;
        c2: = f1.endTime < t2.startTime;
        c = c1 && c2;
        if ( c ) {
            n2: = n1;
            add(n2.path, t2);
            add(SN, n2);
            add(SC, n2)
        }
    }
```

generateRotation의 입력은 열차객체들의 집합을 나타내는 ST이다. 열차는 다음과 같은 다음과 같은 클래스로 나타낼 수 있다.

```
class train {
    int startTime; // 출발시각
    int endTime; // 도착시각
    string startStation; //출발역
    string endStation; //도착역
}
```

Fig 4는 node 개체의 개념을 그림으로 나타낸 것이다. generateRotation에서 가장 중요한 역할을 하는 node는 후보운용을 만들기 위한 경로(열차들의 집합)를 유지하고 있는 멤버 path를 갖는 클래스이다.

c1은 선행열차의 도착역과 후행열차의 출발역이 일치하면 true를 갖는 bool형 변수이고, c2는 선행열차의 도착시각이 후행열차의 출발시각 이전이면 true를 갖는 bool형 변수이다. 이들 두 변수는 후보운용이 만족해야 하는 기본제약조건을 나타낸다. 유지보수제약조건 등 후보운용이 만족해

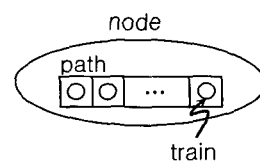


Fig. 4. node 개체

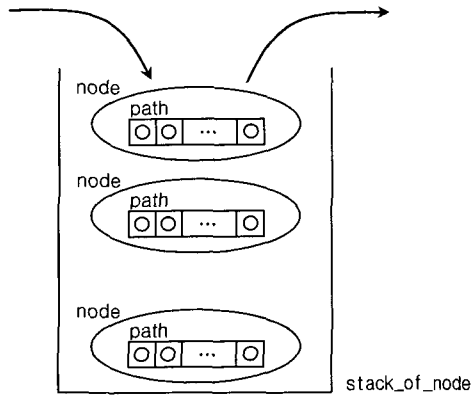


Fig. 5. stack_of_node 개체

야 할 추가적인 제약조건을 고려할 경우 bool형 변수를 추가하여 알고리즘에 반영할 수 있다. 영구꼬리표알고리즘이나 제약프로그래밍에서는 반영하기 어려운 제약조건을 자유롭게 bool형 변수로 작성할 수 있으며 이러한 특성이 본 논문에서 제시하는 알고리즘의 장점이라 할 수 있다.

Fig. 5는 stack_of_node 개체의 개념을 나타낸 것이다. stack_of_node 개체는 표준템플릿라이브러리(standard template library: STL)에서 스택(stack)의 자료구조를 구현해 놓은 컨테이너인 vector를 사용하여 구현할 수 있다.

empty(SN)은 stack_of_node 개체 SN이 비어있으면 true를 돌려주는 함수이다. remove(SN, last(SN))은 SN에서 SN의 마지막 개체를 제거하는 함수이다. add(n2, path, l2)는 n2.path에 l2를 추가하는 함수이다.

4. 계산실험

앞절에서 제시한 열거알고리즘 generateRotation에 대한 수치적 계산실험을 수행하기 위해 차량소요계획 수립 프로그램 RG³⁾를 C++로 작성하였다. RG에는 generateRotation이 포함되어 있으며 제2절에서 제시한(SPP)의 해를 구하는 작업은 ILOG CPLEX 8.0을 사용하였다. 모든 계산실험은 Pentium IV CPU가 탑재된 PC에서 수행하였다. RG에서 반영한 제약조건은 다음과 같다. ①선행열차의 도착역과 후행열차의 출발역이 일치해야 하며, ②(선행열차의 도착시각 + 최소반복시간)이 후행열차의 출발시각 이전이어야 한다는 것의 두 가지이며 최소반복시간은 40분으로 설정하였다. 그리고 (SPP)의 목적함수 식 (1)에서 후보운용의 비용계수는

2) STL에 관한 자세한 내용은 Josuttis(1999)를 참조할 수 있음.
 3) RG(Routing Generator)에 관한 자세한 내용 및 계산실험에서 사용된 3가지 열차시각표는 <http://seongho.inha.ac.kr/research/setpar>에서 찾을 수 있음.

Table 3. 계산실험 결과

열차시각표	열차시각표의 특성		계산실험 결과			
	열차수	포함된 역의 수	후보운용 수	후보운용 생성시간 (초)	SPP의 최적해 (최소 운용수)	SPP의 해를 구하는 시간(초)
t160d1	160	5	26,729	0.871	33	9.904
t180d1	180	6	31,657	0.941	46	14.310
t192d1	192	8	29,303	0.981	45	70.071

$c_j = 1$, for $j = 1, 2, \dots, n$ 으로 설정하였다.

Table 3은 계산실험 결과를 요약한 것이다. 160개 열차와 5개 역으로 구성된 열차시각표 t160d1에서는 후보운용이 26,729개 생성되었으며(SPP)의 최적해로 33개의 운용이 선택되었다. 후보운용을 생성하는데 0.871초가 소요되었고 (SPP)의 해를 구하는데 9.904초가 소요되었다. 180개의 열차와 6개 역으로 구성된 t180d1에서는 31,657개의 후보운용이 생성되었고(SPP)의 최적해로 46개의 운용이 선택되었다. 후보운용을 생성하는데 0.941초가 소요되었고, (SPP)의 해를 구하는데 14.310초가 소요되었다. 192개의 열차와 8개 역으로 구성된 t192d1에서는 29,303개의 후보운용이 생성되었으며 45개의 운용이 선택되었다. 후보운용 생성에 0.981초가 소요되었고 최적 운용을 선택하는데 70.071초가 소요되었다.

5. 결론

본 논문에서는 철도차량소요계획을 수립할 때 유용하게 사용될 수 있는 열거알고리즘 generateRotation을 제시하고 3가지 열차시각표를 대상으로 계산실험을 수행하였다. 본 논문에서 제시한 알고리즘은 유지보수제약조건 등 후보운용이 만족해야 할 추가적인 제약조건을 고려할 경우 영구꼬리표알고리즘이나 제약프로그래밍에서는 반영하기 어려운 제약조건을 자유롭게 bool형 변수로 작성할 수 있다. 따라서 후보운용 생성을 위한 추가제약을 갖는 최단경로문제에서 영구꼬리표알고리즘이나 제약프로그래밍으로 반영하기 어려운 형태의 제약조건이 존재할 경우 열생성기법을 대체할 수 있는 대안으로 유용하게 사용될 수 있을 것으로 기대한다.

참고 문헌

1. Barnhart, C., E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance (1996), Branch-and-Price: Column Generation for Solving Huge Integer Programs, Georgia Institute of Technology, School of Industrial and Systems Engineering
2. Ben Khedher, N., J. Kintanar, C. Queille, and W. Stripling (1998), "Schedule Optimization at SNCF: From Conception to Day of Departure," *Interfaces*, Vol.28, No.1, pp.6-23
3. Boole, J. M. P. (1980), "The Solution of a Railway Locomotive Scheduling Problem," *Journal of the Operational Research Society*, Vol.31, pp.943-948
4. Deboo, N. M. (1973), Aircraft Routing and Maintenance Assignment System : An Aid in Developing Airline Schedules, 13th AGIFORS Symposium, Acapulco, Mexico.
5. De Silva, A. (2001), "Combining Constraint Programming and Linear Programming on an Example of Bus Driver Scheduling," *Annals of Operations Research*, Vol.108, pp.277-291
6. Desrochers, M., and F. Soumis (1988), "A Generalized Permanent Labelling Algorithm for the Shortest Path Problem with Time Windows," *Infor*, Vol.26, No.3, pp.191-211.
7. Forbes, M. A., J. N. Holt, and A. M. Watts (1991), "Exact Solution of Locomotive Scheduling Problems," *Journal of the Operational Research Society*, Vol.42, No.10, pp.825-831
8. Josuttis, N. M. (1999), *The C++ Standard Library: A Tutorial and Reference*, Indianapolis, IN, Addison-Wesley.
9. Neapolitan, R. E., and K. Naimipour (2004), *Foundations of ALGORITHMS Using C++ Pseudocode*, 3rd Edition, Jones and Bartlett Publishers Inc., MA.
10. Simpson, R. W. (1969), A Review of Scheduling and Routing Models for Airline Scheduling, 9th AGIFORS Symposium, Broadway, England.
11. Teodorovic, D. (1988), *Airline Operations Research*. Transportation Studies (N. Ashford, and W. G. Bell, Eds.) Gordon and Breach Science Publishers, New York
12. Wright, M. B. (1989), "Applying Stochastic Algorithms to a Locomotive Scheduling Problem," *Journal of the Operational Research Society*, Vol.40, No.2, pp.187-192
13. Ziarati, K., F. Soumis, J. Desrosiers, S. Gelin, and A. Saintonge (1997), "Locomotive Assignment with Heterogeneous Consists at CN North America," *European Journal of Operational Research*, Vol.97, pp.281-292