韓國經營科學會誌
第27卷 第4號
2002年 12月

지하철 일간승무계획문제의 정수계획해법

변종익* · 이경식* · 박성수* · 강성열**

An Integer Programming Approach to the Subway Daily Crew Scheduling Problem

Jongik Byun* · Kyungsik Lee* · Sungsoo Park* · Sungyel Kang**

■ Abstract ■

This paper considers subway crew scheduling problem. Crew scheduling is concerned with finding a minimum number of assignments of crews to a given timetable satisfying various restrictions. Traditionally, crew scheduling problem has been formulated as a set covering or set partitioning problem possessing exponentially many variables, but even the LP relaxation of the problem is hard to solve due to the exponential number of variables. In this paper, we propose two basic techniques that solve the subway crew scheduling problem in a reasonable time, though the optimality of the solution is not guaranteed. We develop an algorithm that solves the column-generation problem in polynomial time. In addition, the integrality of the solution is accomplished by variable-fixing technique. Computational result for a real instance is reported.

Keyword : Crew Scheduling, Subway, Column Generation, Variable-Fixing

1. 서 론

본 연구에서는 지하철 승무계획문제를 다루고자 한다. 일반적으로 승무계획문제는 최소 비용으로 운행계획을 완수하도록 승무원을 할당하는 문제이다. 이러한 승무계획문제는 산업현장 전반에서 발생하며, 특히 대형항공사나 철도회사의 경우에 있어서 많은 연구가 수행되어 왔다. 지금까지 대형항

논문접수일 : 2001년 6월 4일 논문게재확정일 : 2002년 10월 10일
* 한국과학기술원 산업공학과
** 홍익대학교 경영정보학과

공사나 철도회사는 승무계획문제를 두 단계의 문제로 분해하여 접근하였다. 첫 단계는 한 명의 승무원이 하루에 운행할 수 있는 조건을 만족하도록 몇 개의 기본적인 운행단위를 조합하는 문제이며, 이 단계에서는 특정 승무원이 어느 운행을 수행할지는 결정되지 않는다. 두 번째 단계는 하루 운행들을 여러 개 연결하여 실제로 한 명의 승무원이 운행가능한 작업스케쥴들을 만들고, 이러한 스케쥴들을 각각의 승무원들에게 할당하여 승무계획을 결정하는 문제이다. 항공사나 철도회사의 경우 승무원이 한 번 운행에 나서면 경유지들을 거쳐 다시 출발지로 돌아오는데 수일 이상 소요되는 경우가 발생하므로, 하루 운행들을 여러 개 연결한 것이 승무원이 한 번에 운행하는 스케쥴이 된다. 항공사나 철도회사의 경우에는 첫 번째 단계에서 생성하는 하루 스케쥴의 수는 그리 많지 않으나, 두 번째 단계에서 만들어지는 운행가능한 작업스케쥴의 수는 하루 스케쥴의 수에 비례해 조합적으로 증가해 문제의 최적해를 구하는 것이 매우 어려워 진다. 예를 들면 실제 산업현장에서 발생하는 문제에서는 가능한 작업스케쥴의 수가 수십억개에 달하는 것으로 알려져 있다.

이와는 달리 지하철 승무계획은 지하철이라는 산업의 특성상 도시라는 비교적 좁은 공간에서 작업이 이루어지고 승무원들의 작업환경이 비슷하기 때문에, 승무계획문제는 승무원이 하루에 작업할 양을 결정해 주기만 하면 된다. 하지만 비슷한 시간과 환경에서 이루어지는 기본운행단위들이 많은 이유로 인해서, 하루에 운행가능한 스케쥴의 수가 항공회사나 철도회사의 경우와 비교하여 최소한 수백 배 이상 많다는 특징이 있다. 즉 지하철 승무계획문제에 나타나는 변수의 수는 기본적인 운행단위의 수가 같은 경우라 하더라도 항공회사나 철도회사의 문제보다 최소한 수백 배 크게 된다. 이것이 지하철 승무계획문제가 기존의 승무계획문제들과 가장 크게 구별되는 점이다.

먼저 본 연구에서 사용되는 용어들을 정의하기로 한다. 지하철승무계획의 가장 기본적인 운행단위인 단위운행구간(trip)은 동일한 승무원이 휴식을 취하지 않고 운행하여야 하는 작업단위이다. 각각의 단위운행구간은 출발시간, 출발역, 도착시간, 도착역 등의 기본적인 정보와 몇몇 추가적인 데이터를 가진다. 한 명의 승무원이 하루에 작업할 수 있는 단위운행구간들의 집합을 일간운행구간(duty)이라 하며, 모든 일간운행구간은 단체협약, 회사규정 등 여러 가지 제약조건들을 만족하여야 한다. 각 일간운행구간은 출발역과 도착역이 같아야 하며, 그 역을 승무기지라고 한다. 본 논문은 지하철운행계획이 주어졌을 때, 주어진 제약조건을 최소의 비용으로 만족시키는 일간운행구간집합(duty set)을 찾아내는 문제로 이해될 수 있다. 항공회사의 승무계획의 경우에는 단위운행구간을 flight라 부르며, 일간운행구간(duty)에는 승무기지로 돌아오는 제약이 존재하지 않으며, 장기운행구간(pairing)은 몇 개의 일간운행구간을 연결하여 다시 출발한 승무기지로 돌아오는 운행계획을 수립하는 것을 말한다.

일반적으로 승무계획문제는 다음과 같은 집합피복(set covering)문제 (SC)로 모형화된다. 본 논문에서 다루어지는 지하철 승무계획문제는 (SC)이외에 추가적인 제약식을 가진다.

$$(\mathrm{SC})\quad \min \sum_{j\in J} c_j x_j$$

$$s.t.\ \sum_{j\in J} a_{ij} x_j \ge 1,\quad i\in I,$$

$$x_j \in \{0,1\},\qquad j\in J,$$

여기서, I : 단위운행구간들의 집합,

J : 일간운행구간들의 집합.

여기서 x_j는 일간운행구간 j와 관련된 결정변수로서 일간운행구간 j가 승무계획에 포함되면 $x_j = 1$이고, 포함되지 않는 경우는 $x_j = 0$이다. 제약식에 있는 결정변수 x_j의 계수로 사용되는 a_{ij}는 다음과 같이 주어진다. 만약 단위운행구간 i가 일간운행구간 j에 포함되면 $a_{ij} = 1$이고, 그렇지 않을 경우에는 $a_{ij} = 0$이다. c_j는 일간운행구간 j를

운행하는데 소요되는 비용이다. 위의 수리 모형에 나타나는 변수의 수는 제약을 만족하는 모든 일간 운행구간의 수와 같게 된다.

승무계획문제에 대한 (SC)모형의 특징은 문제를 최적으로 풀기 위해서는 고려해야 하는 변수의 수가 매우 많다는 것이다. 일간운행구간의 수는 운행되어야 하는 단위운행구간의 수가 많아짐에 비례해 지수적으로 증가한다. 이러한 문제를 해결하기 위해서 본 연구에서는 열생성기법을 이용하였다. 열생성기법이란 모든 가능한 변수들을 고려하여 (SC)를 푸는 대신, 일부의 변수들만을 사용하여 (SC)의 기저가능해를 구한 후 이를 개선시킬 수 있는 변수들을 생성하여 추가하는 방식으로 (SC)의 해를 구하는 기법을 말한다. 반복적인 변수의 추가를 통해서 (SC)에 나타나는 변수들 중 일부만을 가지고 최적해를 구해 나가게 된다.

승무계획문제와 관련하여 다양한 해법들이 개발되어져 왔다. Anbil *et al.*[1]과 Gershkoff[12]는 지역최적화기법(local optimization)을 일정계획문제에 적용하였다. 우선 제약식을 만족시키는 해를 하나 구한 뒤, 그 해보다 나은 해를 구하기 위해 지역최적화기법을 사용한다. 지역최적화기법은 사용한 것보다 좀 더 나은 해를 구하기 위해서 (SC)의 선형완화식을 이용하기도 하였다. 이 방법은 가능한 장기운행구간(pairing)들 중에서 기준을 만족시키는 것들을 생성한 후, 생성된 장기운행구간들만으로 구성된 문제의 선형완화식을 푸는 것이다. Anbil *et al.*[1]은 800여개의 단위운행구간들로 이루어진 항공승무계획문제에 대해서 수백만개의 장기운행구간을 생성한 후 선형완화식을 풀었다. 수백만개라는 수는 많은 듯이 보이지만 실제로 가능한 장기운행구간의 수가 수십억개 정도인 것을 감안하면 그 비율이 매우 작은 것을 알 수 있다. 선형완화식의 해는 SPRINT라는 기법에 의해서 구해진다. SPRINT 기법은 수백만개의 장기운행구간들 중에서 다시 수천개만을 골라 최적화 한 후, 그중 해의 개선에 도움이 되지 않는 것들은 제거하고 해를 개선시킬 수 있는 것들을 추가하여 다시 최적화 한다. 이러한 방식은 더 이상 추가할 장기운행구간이 없을 때까지 반복된다. 그러나 이러한 방식으로 구한 해는 (SC) 선형완화식의 최적해를 보장해 주지는 못한다.

Caprara *et al.*[8]과 Beasley *et al.*[6]은 라그랑지 완화법을 일정계획문제에 적용하였다. Hoffman *et al.*[13]은 절단평면기법을 이용하여 30만개의 장기운행구간을 가지는 승무계획문제의 최적해를 구하였다. Marsten *et al.*[15]과 Falkner *et al.*[11]은 집합분할문제(Set partitioning)로 모형화한 승무계획문제의 최적 알고리듬을 제안하였으며, Ball *et al.*[3], Ball *et al.*[4], Bianco et al.[7], Martello *et al.*[16]은 항공사나 운송회사의 승무계획문제에 대한 발견적 해법들을 제안하였다. Paias *et al.*[19], Paixao *et al.*[20]은 버스회사의 승무계획 문제에 대한 해법을 제안하고 있다.

열생성기법을 이용하여 실제로 선형완화식의 최적해를 구하는 방식도 제안되었다. 열생성기법을 사용하면 수리모형 상에서는 모든 가능한 장기운행구간들의 일부만으로 문제의 해를 구하지만, 그 해는 모든 가능한 장기운행구간들을 고려한 문제를 풀었을 때 구한 해와 같은 목적값을 준다. 열생성기법의 일반적인 접근법에 대해서는 Barnhart *et al.*[5]을 참고하기 바라며, 통신망과 일반화된 할당문제(general assignment problem)에 대한 응용사례가 Park *et al.*[21]과 Savelsbergh[22]에 잘 나타나 있다. 열생성기법의 적용을 위해 열생성문제라는 새로운 문제의 해결이 필요하다. Lavoie *et al.*[14]은 승무계획문제에서 나타나는 열생성문제를 특수한 구조를 가지는 그래프에서의 최단경로문제로 변환하였으며, 열생성기법을 사용하여 선형완화식을 풀 경우 그 해가 정수해인 빈도가 매우 높아지는 현상을 보고하고 있다. Vance *et al.*[23]은 수리모형에 약간의 변화를 주어 선형완화식에서 얻는 정수해의 빈도를 더욱 높이기도 하였다. Crainic *et al.*[9]은 열생성기법을 사용한 발견적 해법을 항공승무계획에 적용했으며, Desrochers *et al.*[10]도 최단경로문제를 이용하는 열생성기법을 버스운송문

제에 적용하였다. Mingozzi *et al.*[18] 또한 열생성 기법을 사용한 승무계획 방법에 대해 논하고 있다.

본 연구에서의 열생성문제는 계층화된(layered) 그래프에서 최장경로문제로 변형된다. 또한 본 논문의 계층화된 그래프에서는 회로(cycle)가 존재하지 않으므로 최장경로문제는 음수비용을 가진 최단경로 문제로 변형되며 이는 위상정열(topological ordering)을 통하여 다항시간 내에 풀수 있음이 알려져 있다. 이와 관련하여 Ahuja *et al.*[2]을 참고하기 바란다. 이와 비슷한 열생성 문제를 Lavoie *et al.*[14]에서도 찾아볼 수 있다. 주의할 점은 열생성기법을 사용하여 선형완화문제의 최적해를 구할 수 있다 하더라도, 얻어진 해가 정수해가 아닌 경우에는 추가적으로 정수해를 구해주는 과정이 필요하다는 점이다. 그러나, 일반적으로 열생성기법을 이용한 승무계획문제는 그 선형완화식의 최적해를 구하는 것도 상당한 시간을 필요로 하는 어려운 문제이다.

본 연구에서는 문제의 정수해를 빠른 시간에 구하기 위해 다음과 같은 방식을 제안한다. 우선 문제의 선형완화식을 풀때 최적해만을 고집하지 않고 근사최적해를 구하는 것을 허용한다. 그런 다음 근사최적해에 기반하여 몇 개의 변수를 정수로 고정시키는 변수고정법을 사용한다. 변수고정 후 다시 열생성기법을 사용하여 선형완화식의 최적해를 구하고 이에 기반하여 다시 변수고정을 수행한다. 이러한 과정은 정수해를 하나 구할 때까지 반복된다. 본 연구에서는 열생성기법과 변수고정법을 사용하여 정수해를 구하였으며, 더불어 수리모형에 나타나는 변수의 수를 더욱 줄이기 위해서 변수를 제거하는 열제거기법도 사용하였다.

위에서 설명한 방식으로 실제 지하철회사의 데이터에 실험해 본 결과 변수고정 후에는 선형완화식의 최적해를 매우 빠른 시간 내에 얻을 수 있는 경우가 많이 발생하였다. 또한 모든 경우에 문제의 최적정수해를 구해낼 수 있었다.

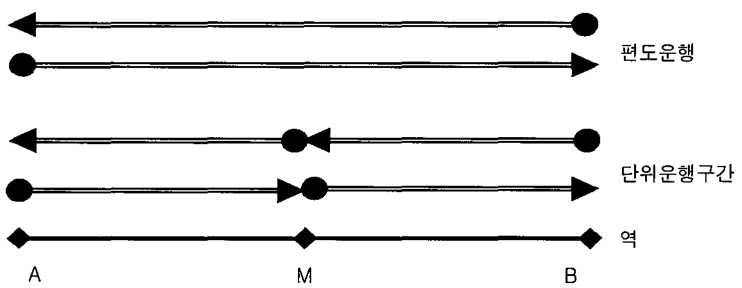
본 논문의 구성은 다음과 같다. 2장에서 지하철 승무계획문제의 세부사항을 설명한다. 3장에서는 문제의 정수계획모형을 제시하고, 4장에서는 세부 알고리듬의 구현을 다루었으며. 알고리듬을 적용한 결과는 5장에 정리되어 있으며, 마지막으로 6장에서 결론과 향후과제에 대해 언급하였다.

2. 문제 설명

본 장에서는 지하철 승무계획문제의 세부사항을 자세하게 설명하고자 한다. 지하철회사의 승무원들은 하루를 단위로 동일한 지하철운행을 반복한다. 즉 승무계획의 기간단위는 하루이다. 참고로 본 연구에서 승무원들의 작업시간은 5 : 20에서 24 : 40까지이다.

지하철 역과 선로의 구조는 매우 간단하다. 선로는 하나의 직선으로 생각할 수 있으며, 수십 개의 역이 선로를 따라 위치하고 있다. 그중 세 개의 역은 승무계획에서 중요한 역할을 한다. 선로의 맨 끝 쪽에 있는 두 개의 역 A와 B는 승무기지라고 불린다. 모든 일간운행구간들은 승무기지 중 하나에서 운행이 시작되어야 하며, 출발한 승무기지에서 운행을 끝내야 한다. 나머지 하나의 역은 선로의 가운데에 위치하며, 이것을 역 M이라 부른다. 승무기지를 제외하고는 역 M에서만 지하철의 승무원 교체가 가능하다. 승무원들은 승무기지를 출발한 후 역 M에서 휴식을 취하면서 다음 운행을 기다릴 수 있다. 따라서 앞서 말한 단위운행구간의 정의에 따라 [그림 1]과 같이 네 종류의 단위운행구간이 발생하게 된다.

편도운행은 선로의 한쪽 끝에서부터 다른 쪽 끝까지 운행하는 것으로, 이것은 승무계획문제에 주어지는 지하철 운행계획의 기본단위이다. 하나의 편도운행은 역 M을 기준으로 2개의 단위운행구간으로 분해된다. 결과적으로 지하철운행계획이 T개의 편도운행계획을 포함하고 있다면, 2T개의 단위운행구간이 있다고 생각할 수 있다. 실제로는 하나의 단위운행구간으로만 만들어지는 편도운행도 존재하므로 정확하게 2T개의 단위운행구간이 발생하지는 않지만, 이것은 문제상황과 관련하여 중요한



[그림 1] 단위운행구간과 편도운행

내용이 아니므로 약 2T개의 단위운행구간이 발생한다고 이해하면 되겠다. 본 연구에서 다루는 승무계획문제는 실제 지하철 운행현장에서 운행되어지고 있는 814개의 단위운행구간으로 구성되어 있으며, 이들 중 최초단위운행구간과 최종단위운행구간은 각각 하나의 편도운행을 형성하고 있다.

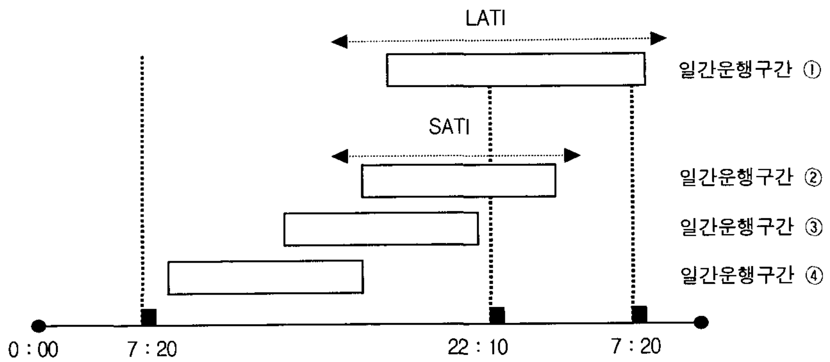
모든 단위운행구간의 운행은 실제로는 수십 개의 일간운행구간들을 통해 이루어진다. 따라서 운행 가능한 일간운행구간의 생성이 중요한 문제가 된다. 일간운행구간의 생성에는 많은 제약조건이 따르게 되는데, 여기에는 운행 환경, 노사합의, 관련 법규 등 여러 가지가 있다. 지금부터는 일간운행구간의 생성과 관련된 제약조건들에 대하여 자세히 설명하도록 하겠다.

승무기지제약조건이란 일간운행구간의 출발역과 도착역이 동일하여야 한다는 제약조건이다. 이 조건은 승무원의 출근장소와 퇴근장소가 같아야 한다는 상황이 반영된 제약조건이다.

하나의 일간운행구간이 시작해서 끝날때까지 걸리는 총 작업시간에도 제약이 따른다. 모든 일간운행구간은 제약된 시간(ATI : Allowed Time Interval) 이내에 하루의 작업을 모두 마쳐야 한다. ATI는 일간운행구간 내의 수면시간 배정여부에 따라 2가지 값을 가진다. 일간운행구간 내에 수면시간이 배정되면 LATI(Long ATI)를 적용받고, 작업 중 수면시간이 배정되지 않은 일간운행구간은 SATI (Short ATI)를 적용받는다. ATI의 적용과 관련하여 일간운행구간의 종류에 대해 알아보자. 일간운행구간은 주간일간운행구간과 야간일간운행구간 2가지 종류가 있다. 주간일간운행구간은 모든 작업이 07 : 20에서 22 : 10 사이에 완료되는 일간운행구간을 의미하며, 주간일간운행구간이 아닌 다른 모든 일간운행구간들은 야간일간운행구간이 된다. 따라서 모든 주간일간운행구간들이 SATI를 적용받는다. 야간일간운행구간의 경우에는 SATI를 적용받는 경우와 LATI를 적용받는 두 가지 경우로 나뉘어 진다. [그림 2]의 일간운행구간 ①의 경우는 야간일간운행구간이면서 LATI를 적용받으며, 일간운행구간 ②의 경우와 같이 야간일간운행구간이긴 하지만 허용되는 총 작업시간은 SATI인 경우도 있다. [그림 2]에서 일간운행구간 ①과 일간운행구간 ②는 야간일간운행구간이고, 일간운행구간 ③과 일간운행구간 ④는 주간일간운행구간이다.

지하철 승무계획에 포함되는 총 야간일간운행구간의 수는 어떤 값을 넘어서는 안되며, 이 값이 작을수록 문제의 해를 구하는 것이 매우 어려워진다.

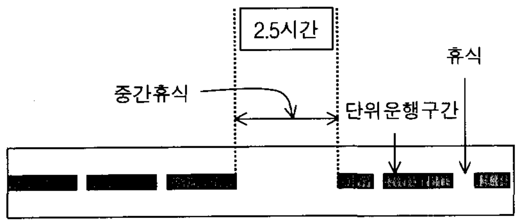
하나의 일간운행구간이 수행하는 단위운행구간의 수에도 제약이 따른다. 일반적으로 승무원들에게 작업량을 고르게 할당하기 위해서 각 일간운행구간들은 비슷한 수의 단위운행구간으로 구성되어져야 한다. 본 연구에서는 모든 일간운행구간이 정확히 10개의 단위운행구간으로 구성되어지는 것으로 가정한다. 10개 이하의 단위운행구간들로 구성된 일간운행구간들은 고려하지 않기때문에, 이론적으로는 최적정수해의 값이 증가할 가능성도 있다. 즉, 10개의 단위운행구간으로 구성된 일간운행구간과 8개의 단위운행구간으로 구성된 일간운행구간로는 운행 가능한 지하철 운행계획이, 10개의 단위



[그림 2] 야간일간운행구간과 주간일간운행구간

운행구간으로 구성된 2개의 일간운행구간로는 운행 불가능한 예를 쉽게 만들 수 있다. 하지만 본 연구에서 다루는 지하철 운행계획은 10개의 단위운행구간으로 구성된 일간운행구간들만으로도 최적 정수해를 구할 수 있었다. 일간운행구간들이 같은 수의 단위운행구간으로 구성된 관계로 하나의 단위운행구간에 둘 이상의 일간운행구간이 할당되어지는 경우가 발생하기도 한다. 이러한 경우 실제 단위운행구간의 운행은 한 명의 승무원이 담당하고 나머지 승무원은 다음 운행을 위해 이동하는 상황이 된다. 하지만 이러한 경우에도 그 단위운행구간과 관련된 모든 승무원은 업무를 수행한 것으로 인정된다.

승무원이 하나의 일간운행구간을 수행함에 있어서 일간운행구간을 구성하는 단위운행구간들 사이에는 휴식시간이 배정되어 있어야 한다. 이러한 휴식시간은 지하철 운행의 안전을 위하여 중요한 요소로 인식되고 있다. 단위운행구간들 사이의 휴식시간은 연속운행일 경우와 같이 없는 경우도 있고, 몇 시간에 걸쳐 배정되는 긴 것도 있다. 본 연구에서는 이러한 휴식시간의 배정에 대해 제약조건이 존재하는데, 그것은 작업시간동안 최소한 한번은 '중간휴식' 이라 불리는 2시간 30분 이상의 휴식시간이 배정되어야 한다는 것이다. 중간휴식을 기준으로 하루의 작업량은 두 부분으로 나누어지게 되는데, 이 두 부분은 각각 6개 이하의 단위운행구간들로 구성되어야 한다.



[그림 3] 중간휴식의 예

결과적으로, 중간휴식은 작업시간 중 세 곳 중에 한 곳에 배정되게 되는데, 그 세 지점은 일간운행구간을 이루는 열 개의 단위운행구간 중 4 번째, 5 번째, 혹은 6 번째 단위운행구간 후가 된다. 이렇게 배정된 중간휴식은 식사시간 또는 작업자 교육 등의 용도로 쓰인다. 러시아워 대에는 승무원의 교대로 인한 지하철의 운행지연을 막기 위해 역 M에서의 승무원 교대를 하지 않으므로, 승무원들은 러시아워 대에는 연속으로 2개의 단위운행구간을 운행하여야 한다. 즉 역 A에서 역 B까지 계속 운행하거나 역 B에서 역 A까지 계속 운행하여야 한다. 하루에 두 번 출 · 퇴근시간대가 러시아워로 지정되어 있다.

운행가능한 일간운행구간을 만들려면 위에서 언급한 모든 제약조건들을 만족시켜야 한다. 이러한 제약조건들은 운행가능한 일간운행구간의 수를 상당히 줄여주지만, 제약조건을 모두 만족시키는 일간운행구간들의 수가 여전히 한꺼번에 다루어지기에는 너무나 많다는 상황에는 변함이 없다. 생성된

일간운행구간들은 각각 운행에 필요한 비용을 가지게 되는데 본 연구에서 고려되는 지하철 승무계획의 지하철 운행에 소요되는 승무원의 수를 줄이는 것을 목적식으로 가지므로 일간운행구간과 관련된 비용은 모두 같은 값을 가지는 것으로 한다.

운행가능한 일간운행구간들로 구성된 일간운행구간집합(duty set)이 지하철 운행계획에서 주어진 단위운행구간들을 모두 운행한다고 해서 그 일간운행구간집합이 실제로 운행가능한 것은 아니며 몇 가지 추가적인 제약이 존재한다. 일간운행구간집합에 포함된 야간일간운행구간의 수에 대한 제약, 각 승무기지에서 출발하는 일간운행구간의 수에 대한 제약들이 그것이다. 이러한 제약들은 일간운행구간 생성시에는 고려되지 않고, (SC)에 추가적인 제약식으로 첨가되게 된다.

3. 수리모형 및 해법

본 장에서는 지하철 승무계획문제의 정수계획모델을 제시하고자 한다. 정수계획모델은 주문제와 부문제로 나뉘어져 설명되어 있다. 더불어 변수고정법의 의미에 대해서도 논의하도록 한다.

3.1 주문제

승무계획문제에는 주로 집합피복(set covering) 모델이 사용되어져 왔으며, 지하철 승무계획문제는 전통적인 집합피복모델에 몇 개의 제약식을 추가하면 된다. 2장의 끝부분에서 언급된 일간운행구간 집합과 관련된 세 가지 제약조건들이 바로 그것이다. 승무기지 A, B는 각각 출발하는 일간운행구간의 수가 U_A, U_B개 이하여야 하며, 승무계획에 포함된 야간일간운행구간의 수는 U_O개 이하여야 한다. 세 가지 제약조건 이외에 2장에서 언급한 각각의 일간운행구간이 만족하여야 하는 여러 가지 제약조건들은 다음 절에서 다루는 부문제를 해결할 때 고려되어 진다. 부문제는 집합피복모델에 나타나는 일간운행구간들을 생성하는 문제로서 승무기지제약조건, 중간휴식제약조건 등을 만족하는 일간운행구간을 생성하여 주문제에 새로운 변수로 추가하게 된다. 집합피복모델에 세 개의 제약조건을 추가한 수리모형이 (MP)에 주어져 있다.

$$\text{(MP)} \quad \min \sum_{j \in J} c_j x_j$$

$$s.t. \quad \sum_{j \in J} a_{ij} x_j \geq 1, \qquad i \in I, \tag{1}$$

$$\sum_{j \in J} a_{pj} x_j \leq U_A \tag{2}$$

$$\sum_{j \in J} a_{qj} x_j \leq U_B \tag{3}$$

$$\sum_{j \in J} a_{rj} x_j \leq U_O \tag{4}$$

$$x_j \geq 0 \ \text{integer}, \quad j \in J, \tag{5}$$

여기서, I : 단위운행구간들의 집합,
J : 일간운행구간들의 집합.

여기서 x_j는 일간운행구간 j와 관련된 결정변수이며, (MP)에서 x_j는 일반적인 정수값을 가지는 것으로 모형화되어 있지만, 집합피복문제의 특성상 (MP)의 최적해에서 x_j의 값은 0이나 1 중의 하나가 된다. 일간운행구간 j가 승무계획에 포함되면 $x_j = 1$, 포함되지 않는 경우는 $x_j = 0$의 값을 갖는다. 제약식에 있는 결정변수의 계수들은 다음과 같이 주어진다. 만약 단위운행구간 i가 일간운행구간 j에 포함되면 $a_{ij} = 1$이고, 그렇지 않을 경우에는 $a_{ij} = 0$이다. 일간운행구간 j가 역 A를 승무기지로 가질 경우 $a_{pj} = 1$, $a_{qj} = 0$이며, 역 B를 승무기지로 가질 경우 $a_{pj} = 0$, $a_{qj} = 1$이다. 일간운행구간 j가 야간일간운행구간인 경우 $a_{rj} = 1$, 아닐 경우에는 $a_{rj} = 0$이다.

지하철 승무계획문제는 필요한 승무원의 수를 최소화하는 것이 목적이다. 따라서 일간운행구간과 관련된 비용은 모두 $c_j = 1$이 된다. 예외적으로 실제 문제를 풀 때에 초기해를 구하기 위한 가상변수에 대응되는 가상일간운행구간의 비용은 1이 아닌 값을 가지게 된다. 문제의 초기변수의 생성에 관련

해서는 4.1절에서 자세히 설명하고 있다.

주문제 (MP)를 풀기 위하여 우선 (MP)의 선형완화식의 최적해를 구한다. (MP)의 선형완화식은 제약식 (5)의 정수조건을 제거한 문제이며, 선형완화식의 최적해는 다음절에서 설명하는 열생성기법을 이용하여 구한다.

3.2 부문제

승무계획문제에는 지수적으로 많은 운행가능한 일간운행구간들이 있고, 그에 따른 수리모형에도 같은 수의 변수가 (MP)에 나타나게 된다. 본 연구에서는 가능한 모든 변수를 수리모형에서 한꺼번에 다루는 대신, 문제의 목적값을 개선시킬 가능성이 있는 변수들을 순차적으로 첨가하여 문제의 선형완화식을 최적화하는 기법인 열생성기법을 사용하고자 한다. 즉, 열생성기법은 가능한 변수들 중 일부만을 가지고 문제의 선형완화식을 최적화 한 후, 목적식의 값을 감소시키기 위해서 새로운 변수를 생성하여 선형완화식에 추가하는 기법이며, 이때 발생하는 새로운 최적화 문제를 열생성문제 혹은 부문제라 한다. 열생성기법을 승무계획문제에 적용한 예로 Crainic *et al.*[9], Desrochers *et al.*[10], Lavoie *et al.*[14], Vance *et al.*[23] 등이 있다.

열생성문제를 지하철 승무계획문제에 적용하여 설명하기 위하여 우선 일간운행구간들의 전체집합 J와 그 부분집합 J'를 생각해 보자. (MP)에서 J를 J'로 치환함으로써 새로운 문제 (RMP)가 발생한다. (RMP)는 (MP)보다 제한된(restricted) 수의 일간운행구간만을 고려하는 승무계획문제이다. u_i $(i \in I), u_p, u_q, u_r$를 각각 (RMP)의 선형완화식에 등장하는 제약식 (1), 식 (2), 식 (3), 식 (4)와 관련된 쌍대변수들이라 하고, $\overline{u_i}\,(i \in I), \overline{u_p}, \overline{u_q}$, $\overline{u_r}$를 최적쌍대값이라 하자. 그렇다면 선형계획법의 특성에 따라 최적화된 (RMP)의 선형완화식에 존재하는 변수들의 수정비용(reduced cost)는 모두 0 이상이 된다. 이를 수식으로 표현하면 아래와 같다.

$$c_j - \Big(\sum_{i \in I} a_{ij}\overline{u_i} + a_{pj}\overline{u_p} + a_{qj}\overline{u_q} + a_{rj}\overline{u_r}\Big) \geq 0,\ j \in J',$$

$$\text{단},\ \overline{u_i} \geq 0,\ i \in I,\ \overline{u_p},\ \overline{u_q},\ \overline{u_r} \leq 0$$

여기서 (MP)의 제약식 (2), 식 (3), 식 (4)의 부등호의 방향은 그와 관련된 쌍대변수 $\overline{u_p}, \overline{u_q}, \overline{u_r}$, 의 값을 0이하로 제약한다는 사실에 주의하기 바란다. 만일 (RMP)의 선형완화식의 최적해가 (MP)의 선형완화식의 최적해가 아닐 경우 할인가가 음수인 일간운행구간 j가 존재한다.

$$1 - \Big(\sum_{i \in I} a_{ij}\overline{u_i} + a_{pj}\overline{u_p} + a_{qj}\overline{u_q} + a_{rj}\overline{u_r}\Big) < 0,\ j \in J \setminus J',$$

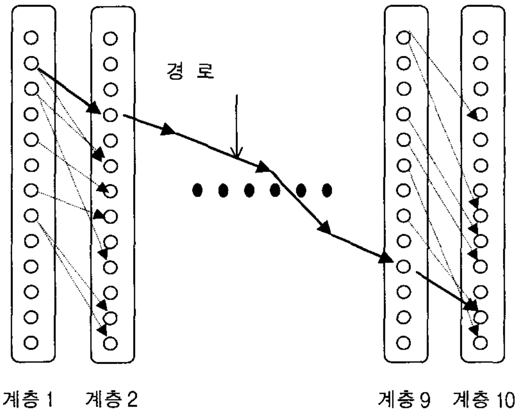
위 식에 나타나는 j와 같은 일간운행구간들을 (RMP)에 추가하여 다시 최적화 함으로써 (RMP)의 선형완화식의 해가 (MP)의 선형완화식의 해에 가까워지게 되며, 이러한 방식을 더 이상 추가할 변수가 없을 때까지 반복함으로써 (MP)의 선형완화식의 최적해를 얻을 수 있게 된다. 일간운행구간 j와 같이 (RMP)에 추가될 일간운행구간들을 찾는 문제를 부문제라 하며 (SUB)에 제시되어 있다.

$$\text{(SUB)}\ \max \sum_{i \in I} a_{ij}\overline{u_i} + a_{pj}\overline{u_p} + a_{qj}\overline{u_q} + a_{rj}\overline{u_r},\ j \in J,$$

(SUB)를 최적화한 결과 목적식의 값이 1보다 큰 일간운행구간이 존재한다고 하자. 이것은 (MP)에 존재하는 변수들 중에서 수정비용(reduced cost)이 음수인 변수가 존재한다는 것을 뜻하며, 이 경우 (RMP)의 선형완화식의 최적해는 (MP)의 선형완화식의 최적해가 아닐 가능성이 존재하게 된다. 따라서 할인값이 음수인 일간운행구간을 (RMP)에 추가하여 다시 최적화하게 된다. 이러한 변수추가와 최적화의 과정을 반복하여 (SUB)의 최적해의 값이 1이하가 되면 (MP)의 선형완화식에 존재하는 모든 변수들의 할인값이 0 이상이 되므로, (RMP)의

선형완화식의 최적해는 (MP)의 선형완화식의 최적해가 되는 것이다. 이러한 방식으로 일부분의 변수만으로 (MP)의 선형완화식을 최적화하게 된다.

본 연구에서는 (RMP)에 추가할 일간운행구간을 찾는 최적화문제인 (SUB)가 회로가 없는 계층구조의 그래프에서 최장경로를 찾는 문제로 변환된다. 이 문제는 음수비용을 갖는 무회로 그래프에서 최단경로를 찾는 문제가 되어 위상정열(topological ordering)을 사용하면 호의 개수에 비례하는 시간에 문제를 풀수 있다. 지금부터는 최장경로 문제가 어떻게 선형완화식의 개선에 필요한 일간운행구간들을 찾아주는지에 대해 설명한다.



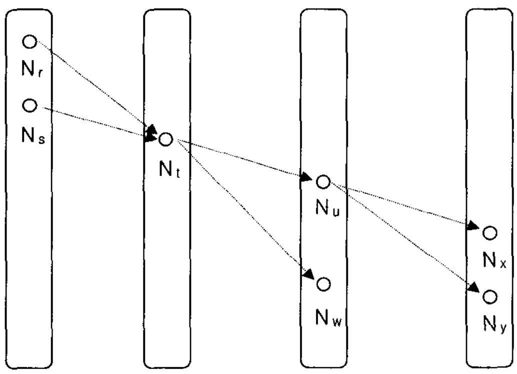
[그림 4] 열생성 그래프

[그림 4]의 그래프는 일간운행구간을 구성하는 10개의 단위운행구간을 나타내는 10개의 계층으로 나뉘어져 있다. 하나의 계층은 각각의 단위운행구간을 나타내는 814개의 노드로 구성되어 있으며, 서로 인접한 두 개의 계층간에만 아크가 존재하며 아크의 방향은 항상 좌측에서 우측으로 향한다. 두 단위운행구간이 아크로 연결되어 있다는 것은 한 승무원에 의해 두 단위운행구간이 연속적으로 운행될 수 있다는 것을 의미한다. 즉, 우측에 있는 단위운행구간의 출발시간이 좌측에 있는 단위운행구간의 도착시간 이후이며, 좌측단위운행구간의 도착역이 우측 단위운행구간의 출발역과 같다는 뜻이다. 일반적으로 하나의 단위운행구간에서 시작되는 아크의 수는 매우 많지만, 러시아워 대인 경우에는 앞에서 설명한 것처럼 연속된 단위운행구간을 수행하여야 하므로 하나의 단위운행구간에서 시작되는 아크의 수가 하나인 경우가 발생한다. [그림 4]에서 계층 1에서 계층 10까지 연결된 경로는 한 승무원이 연속으로 운행할 수 있는 10개의 단위운행구간으로 구성되어 있으므로 하나의 일간운행구간으로 해석될 수 있다. 이것이 계층구조의 그래프에서 경로는 찾는 방식으로 일간운행구간을 생성하는 기본개념이다.

승무기지제약과 총 작업시간제약을 만족하는 경로들은 경로의 출발점과 종착점을 관찰하는 것만으로 쉽게 찾아낼 수 있다. 즉, 일단 출발점이 정해지면 그 출발점과 연관된 경로들 중에서 출발점과 종착점이 같은 경로들만을 고려하는 것으로 승무기지제약을 만족시킬 수 있으며, 그러한 경로들 중에서 출발점과 종착점의 시간차가 총작업시간제약안에 드는 경로들만을 고려함으로써 총작업시간제약도 만족시키는 경로를 구할 수 있다. 하지만 위에서 설명한 모델은 중간휴식이 들어가지 않는 일간운행구간도 생성해 내고 있다. 중간휴식이 들어가는 일간운행구간의 생성은 그래프에 약간의 변형을 가하는 것으로 해결될 수 있다. 이것은 경로상의 단위운행구간이 중간휴식이 추가된 이후의 단위운행구간인지 아닌지를 구분하는 정보를 추가하는 방식이며, 이를 위하여 그래프의 노드의 수를 두 배로 복제한다. 앞서 설명한 바와 같이 중간휴식은 일간운행구간상의 세 지점 중 한 곳에 배정되어야 하며, 그 위치는 4번째, 5번째, 또는 6번째 단위운행구간 이후가 된다. 이러한 경우 4번째 단위운행구간이전은 모두 중간휴식을 취하기 전의 상태여야 하며, 7번째 이후의 단위운행구간은 모두 중간휴식 이후가 되어야 한다. 따라서 이중 5번째와 6번째 단위운행구간들만이 중간휴식 이전인지 이후인지를 나타내는 정보가 필요하며, 이러한 상황을 그래프에 반영하기 위해서 노드복제를 수행한다. 즉 계층 5와 계층 6에 있는 각 노드들은 2개의 노드로 복제된다. 복제된 노드들 중 하나는 중

간휴식 이전의 상태를 나타내는 노드이고 나머지 하나는 중간휴식 이후의 상태를 나타내는 노드이다. 노드복제 전후의 상황이 [그림 5]와 [그림 6]에 비교되어져 있다. 노드복제 이전의 그래프인 [그림 5]에 나타나는 노드 N_t는 노드복제 후 [그림 6]의 노드 N_{t1}과 노드 N_{t2}로 갈라지게 된다. N_{t1}은 중간휴식 이전의 상태를 나타내고, N_{t2}는 중간휴식 이후의 상태를 나타낸다.

노드복제에 따라 아크도 수정하여야 한다. 이를 설명하기에 앞서, 앞으로 자주 사용될 용어를 하나 정의하기로 한다. 단위운행구간 t와 단위운행구간 u사이에 정의되는 '여유시간'은 (u의 출발시간) - (t의 도착시간)의 값을 가진다. 따라서 단위운행구간 t와 단위운행구간 u 사이의 여유시간이 양수이고, 단위운행구간 t의 도착역과 단위운행구간 u의 출발역이 같으면 그래프상에 t에서 u로 가는 아크가 존재한다.

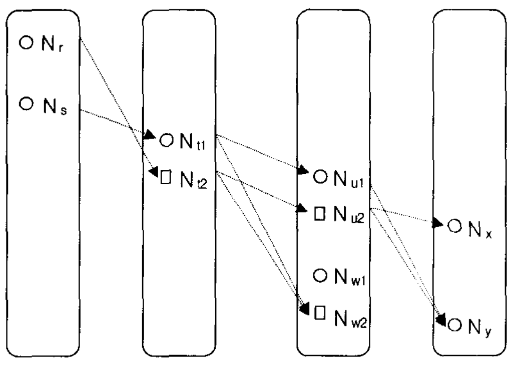


[그림 5] 노드복제 이전의 그래프

이제 아크의 재정렬에 대해 설명하자. 먼저 계층 4와 계층 5 사이의 아크들에 대해 생각해 보자. [그림 5]의 아크(N_r, N_t)는 여유시간이 중간휴식 시간보다 큰 단위운행구간 r과 단위운행구간 t를 연결하고 있음을 나타낸다. 이 아크는 [그림 6]과 같이 아크(N_r, N_{t2})로 수정해야 하며 이것은 중간휴식이 삽입되었음을 의미한다. [그림 5]의 아크(N_s, N_t)는 여유시간이 중간휴식 시간보다 짧은 단위운행구간들을 연결하고 있음을 나타내며, [그림 6]에서 아크(N_s, N_{t1})로 수정하였다. 이것은 아직 중간휴식을 삽입하지 않았음을 의미한다.

다음으로 계층 5와 계층 6사이의 아크들을 살펴보자. [그림 5]의 아크(N_t, N_u)는 시작점이 다른 두 개의 아크로 갈라진다. 하나는 N_{t1}에서 시작하는 것으로 단위운행구간 t와 단위운행구간 u 사이의 여유시간이 중간휴식 시간보다 짧으면, 아크(N_{t1}, N_{u1})로 대응되며, 그렇지 않은 경우 아크(N_{t1}, N_{u2})로 대응된다. 나머지 하나는 N_{t2}에서 시작하는 것으로 [그림 6]에서 아크(N_{t2}, N_{u2})가 된다. 이것은 이미 단위운행구간 t 이전에 중간휴식을 삽입하였으므로 단위운행구간 t와 단위운행구간 u사이의 여유시간과 관계없이 중간휴식을 삽입한 것을 나타내는 다음 계층의 노드로 연결한다.

마지막으로 계층 6과 계층 7사이의 아크를 살펴보자. 계층 6과 계층 7사이의 아크들은 여유시간에 따라 하나의 아크로 재정렬되는 경우와 두개의 아크로 갈라지는 경우가 발생한다. 여유시간이 중간휴식보다 짧은 단위운행구간들을 연결하는 아크(N_u, N_x)의 경우, [그림 6]에서 아크(N_{u2}, N_x)로 변형된다. 이것은 6번째 단위운행구간 이전에 중간휴식을 먼저 삽입했어야 한다는 것을 의미한다. 여유시간이 중간휴식보다 긴 단위운행구간들을 연결하는 아크(N_u, N_y)의 경우에는 [그림 6]과 같이 (N_{u1}, N_y), (N_{u2}, N_y) 2개의 아크로 나누어지게 된다.



[그림 6] 노드복제 이후의 그래프

노드복제 이후의 그래프에서 발견되는 경로는 항상 하나 이상의 중간휴식을 갖게 되므로 운행가능한 일간운행구간의 요구조건을 모두 만족시키는 경로들을 구할 수 있게 된다. 결국 계층화된 그래프에서의 경로를 찾는 문제를 해결하는 것으로 운행가능한 일간운행구간을 생성하는 문제를 해결할 수 있음을 알았다.

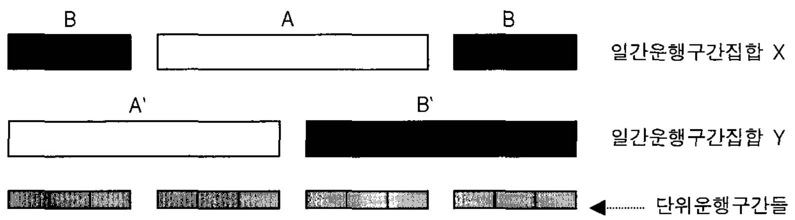
부문제는 이러한 경로들 중에서 (SUB)의 목적식의 값이 가장 큰 경로를 찾는 문제이다. 이 문제를 풀기 위해 그래프의 노드들에 가중치들을 부여한 후 그 가중치들의 합이 최대인 경로를 찾는 문제인 최장경로문제를 풀어야 한다. 노드에 주어지는 가중치는 노드에 대응하는 단위운행구간과 관련된 제약식의 쌍대변수값 $\overline{u_i}(i \in I)$이 된다는 것은 (SUB)를 보면 쉽게 이해된다. 이렇게 찾은 경로들에 경로의 특성과 관계된 (SUB)의 마지막 세 항을 추가적으로 계산함으로써 (SUB)를 최적화 할 수 있게 된다. 앞서 언급한 것처럼 회로를 갖지 않는 계층화된 그래프에서의 최장경로 문제는 다항시간에 풀 수 있으므로, 우리는 부문제인 (SUB)를 다항시간 안에 풀 수 있는 것이다.

3.3 변수고정법

(MP)의 선형완화식을 최적화하는 것은 그 자체로도 상당한 시간을 필요로 하는 문제이지만 정수해를 보장해 주지 않는다. 실제로 문제를 풀어 볼 경우 선형완화식의 해는 소수점 이하의 값을 가지므로, 정수해를 구하기 위한 추가적인 방법론이 요구된다. 하지만 본 연구에서 다루는 것과 같이 큰 문제의 경우, 최적정수해를 빠른 시간안에 구해주는 일반적인 방법론이 아직 알려져 있지 않다. 본 장에서는 문제의 정수해를 구하기 위한 간단하지만 강력한 방법인 변수고정법을 소개한다.

변수고정법은 몇 개의 변수값을 1로 고정시키는 방법이다. 변수고정법을 사용하는 이유는 승무계획문제가 가지는 대칭성 때문이다. 정수계획문제들 중에는 여러개의 최적정수해를 가지며, 하나의 최적정수해에서 다른 최적정수해를 용이하게 구할 수 있는 문제들이 있다. 그러한 문제의 경우에는 여러 최적정수해 중 하나만 찾아낸다면 다른 해들을 구할 수 있으므로, 하나의 최적정수해를 찾는 것이 상당한 중요성을 갖는다. 지하철 승무계획문제는 이러한 대칭적 속성을 가진 문제로 이해될 수 있다. 지하철 승무계획문제의 최적정수해 중 하나를 [그림 7]의 일간운행구간집합 X 라 하고 이것이 모든 단위운행구간을 만족시키는 승무계획이라고 하자. X를 구성하는 두 일간운행구간 A, B를 약간 이동시켜 새로운 일간운행구간 A', B'로 구성된 일간운행구간집합 Y를 만들어 낼 수 있다. 이러한 방식으로 수많은 최적정수해를 생성해 낼 수 있는 것이다.

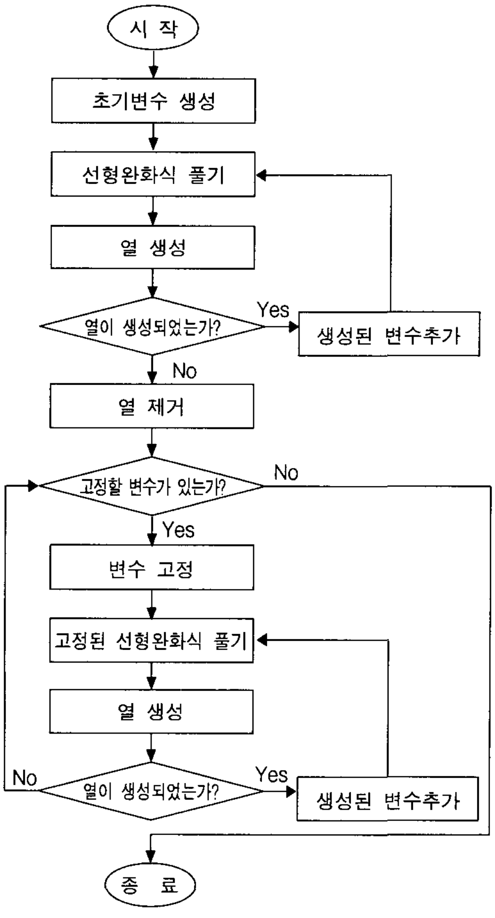
많은 최적정수해 중 특정 일간운행구간을 포함하는 일간운행구간집합들이 존재할 것이며, 이러한 일간운행구간집합들은 특정 일간운행구간과 관련된 변수의 값을 1로 고정시키는 방법으로 얻을 수 있다. 변수고정법을 사용함으로써 변수고정된 수리모형에서는 얻을 수 없는 최적정수해들이 존재하



[그림 7] 여러 개의 최적해

겠지만, 여전히 고정된 일간운행구간을 포함하는 최적정수해들은 구할 수 있다. 따라서 본 연구에서는 선형완화식에서 큰 값을 가지는 변수들을 1로 고정시키는 과정을 반복하여 사용함으로써 최적정수해를 구하여 보고자 한다. 변수고정과 관련된 발견적 해법에 대해서는 Miller *et al.*[17]이 잘 정리하고 있다.

본 연구에서는 모든 변수값이 정수가 될때까지 변수고정을 반복해서 사용한다. 먼저 기준을 만족시키는 몇 개의 변수값을 1로 고정한 다음, 고정된 선형완화식을 열생성기법을 사용하여 최적화한다. 그 다음 다시 고정시킬 변수들을 찾고 다시 선형완화식을 최적화한다. 이러한 방식은 정수해를 하나



[그림 8] 정수해 구현 알고리듬

찾을 때까지 반복해서 적용된다. 수리 모형상에서의 변수고정은 그 변수의 하한한계치를 1로 바꾸어 주는 방식으로 이루어진다. 이렇게 한계치가 바꾸어진 수리모형은 한계변수단체법(bounded variable simplex method)을 사용하여 그 선형완화식을 최적화한다. 고정된 변수를 가지는 선형완화식에서 풀어야 하는 부문제는 고정된 변수가 없는 선형완화식의 부문제와 같다. [그림 8]에서는 정수해를 하나 구하기 위한 방법을 간단하게 정리하고 있다.

4. 알고리듬 구현

4.1 초기해

열생성기법을 사용하여 문제를 접근할 때에는 열생성기법이 필요로 하는 쌍대변수값들을 최초로 제공할 수 있는 초기열집합이 존재하여야 한다. 이러한 초기열집합을 초기변수라 부르며, 초기변수들만으로 선형완화식을 풀었을때 초기기저해를 확보할 수 있어야 열생성기법을 계속적으로 적용할 수 있다. 그러나 (MP)의 모든 제약조건들을 만족시키는 초기변수들을 찾는 것은 어려운 일이다. 특히 제약식 (4)에 주어진 조건처럼 특정한 수 이하의 야간 일간운행구간을 가지는 일간운행구간집합을 생성하는 것은 매우 어렵다. 따라서 가상의 일간운행구간을 나타내는 가상변수를 포함하는 초기변수들을 만들어 사용하고자 한다.

가상 일간운행구간은 모든 단위운행구간을 운행하는 일간운행구간이다. 따라서 가상 일간운행구간에 대응하는 열은 모든 행에 1의 값을 가진다. 가상변수의 비용은 매우 큰 값(=10000)을 가지는 것으로 하였다. 이와는 별도로 운행되어져야 하는 814개의 단위운행구간들을 운행가능하게 하는 174개의 일간운행구간들을 초기변수에 추가하였다. 이들은 각각 비용값으로 1을 가지며 승무기지 제약조건 등 개별 일간운행구간과 관련된 제약조건들을 모두 만족한다. 가상변수를 제외한 변수들로는 제약식 (2), 식 (3), 식 (4)를 만족시키는 일간운행구간

집합을 찾을 수는 없지만, 가상변수로 인해 (RMP)는 초기기저해를 갖게 된다. 이러한 초기변수들로 (MP)의 최적화가 시작된다. 만약, 제약식을 만족하는 승무계획이 존재할 경우 가상변수는 열생성을 이용하는 선형완화식의 최적화 과정을 거치면서 그 값이 0으로 고정된다.

좀더 좋은 초기변수들을 구하는 것은 최종해의 질과 전체적인 계산시간에 큰 영향을 주기 않는 것으로 보인다. 주문제의 선형완화식의 목적값은 초기의 열삽입시에는 매우 빨리 줄어들다가, 최적값에 접근하면 접근할수록 그 속도가 매우 늦어지기 때문이다.

4.2 열생성

앞서 3.2절에서 설명한 바와 같이 (RMP)에 추가할 열의 생성은 노드에 가중치가 주어진 무회로 계층화 그래프(acylic layered graph)에서의 최장경로문제로 변형된다. 일반적으로 최장경로문제는 아크에 가중치가 주어지는데, 본 문제에서 생성한 그래프도 노드에 주어진 가중치를 아크로 옮겨 일반적인 최장경로문제로 생각할 수 있다. 각 노드의 가중치는 그 노드에서 출발하는 모든 아크들의 가중치로 옮겨진다. 이 경우 그래프의 마지막 계층에 속하는 노드들의 가중치를 반영하기 위하여 가상적인 아크들의 생성이 필요하며, 이를 위해 그래프 오른쪽에 평행한 아크들로 연결되는 계층이 추가로 생성된다. 무회로 계층화된 그래프는 최장경로문제를 호의 개수에 비례하는 계산시간 내에 풀수 있다.

일간운행구간이 출발하는 단위운행구간이 주어진 경우 사전적인 위상정열(topological ordering)을 수행한 후 동적계획기법(dynamic programming)을 사용하여 최장경로를 구하게 된다. 하나의 출발점에서는 주간일간운행구간인 최장경로와 야간일간운행구간인 최장경로 2개를 구하며, (SUB)에 의해서 두개의 최장경로중 야간일간운행구간인 최장경로에서는 야간일간운행구간 제약조건과 관련된 쌍대변수값 $\overline{u}_r$를 더하게 된다. $\overline{u}_r$의 부호가 음수이므로 실질적으로 야간일간운행구간인 최장경로값은 줄어들게 된다. 이후 두 개의 최장경로의 경로값을 비교하여 더 큰 값을 가지는 것을 주어진 단위운행구간에서 출발하는 최장경로로 삼는다. 본 연구에서는 일간운행구간이 시작할 수 있는 모든 단위운행구간마다 그곳에서 출발하는 최장경로를 모두 구한다.

최장경로들에 대응하는 일간운행구간들은 (SUB)의 목적식 값을 기준으로 정렬시킨다. 계산된 값이 1보다 큰 것들은 (RMP)에 추가되어 목적식의 값을 개선시킬 여지가 있는 것들이다. 본 연구에서는 (SUB)의 목적식 값이 큰 일간운행구간들을 (RMP)에 추가하며 한 번에 추가되는 일간운행구간들의 수는 40~100개 사이에서 결정하였다. 한 번에 추가되는 변수의 수가 너무 많으면 문제의 크기가 너무 커지는 문제가 발생할 수 있고, 한 번에 추가되는 변수의 수가 너무 작으면 (RMP)의 목적값의 변화가 너무 작아서 비효율적일 경우가 발생하기도 한다.

열생성과 관련해서는 여러가지 전략적인 결정들을 해야 한다. 특히 선형완화식의 근사최적해를 사용할 경우와 관련하여 많은 결정을 내려야 한다. 선형완화식의 목적값은 최적정수해의 하한한계치로 쓰이는데 이러한 목적을 위해서는 대부분 선형완화식의 근사최적해로 충분하였다. 따라서 선형완화식의 목적값이 최적값에 근사하게 되어 열삽입으로 얻을 수 있는 최적값의 증가가 미미해지게 되면, 더 이상의 열삽입을 중지하고 그때까지 얻은 근사최적해를 가지고 변수고정에 들어가는 방식을 사용하였다. 근사최적해의 사용은 불필요한 계산을 줄이는데 큰 효과를 발휘하였다. 본 연구에서는 근사최적해와 관련하여 열삽입에 두 가지 기준을 적용하였다. 변수고정 이전의 문제에 대해서는 문제에 나타나는 변수의 수가 주어진 상한을 넘으면 열삽입을 중지하고 그때까지 얻은 근사최적해를 사용하며, 본 연구에서는 변수개수의 상한치로 11900~12000의 값을 사용하였다. 변수고정 이후의 문제에 대해서는 열삽입의 회수에 제한을 두고, 그 회

수가 주어진 상한을 넘으면 근사최적해를 사용하여 다음 변수고정으로 넘어가게 하였으며, 사용된 열삽입 회수의 상한치는 8~20이다.

4.3 열제거

열생성기법을 사용하여 수리 모형에 나타나는 변수의 수를 줄이긴 하였으나, 열생성이 계속되면서 수리모형에 나타나는 변수의 수는 계속 증가하게 된다. 이러한 이유로 문제의 크기를 줄이는 것이 다시 중요한 문제가 된다, 열제거는 이러한 필요를 만족시키기 위해 사용된다.

본 연구에서는 변수고정전의 선형완화식의 최적해 중에서 0의 값을 가지는 변수들을 모두 제거하는 방식을 사용하였으며 이 방법으로 문제의 풀이시간을 상당히 줄일 수 있었다. 열제거 이후 문제의 해를 구하지 못하는 상황이 발생하는 것을 방지하기 위하여 가상일간운행구간은 제거대상에서 제외하였다.

변수고정후의 선형완화식에서도 연속적으로 0의 값을 갖는 변수들을 제거시키는 방법을 사용하기도 하였으나 오히려 문제의 풀이시간을 증가시켰다.

4.4 변수고정

열생성기법을 사용하여 (MP)의 선형완화식의 근사최적해를 구한 후, 그 해에 기반하여 (MP)의 정수해를 구하기 위해 변수고정법을 사용한다.

지하철 승무계획문제의 경우 선형완화식에서 얻은 변수의 값 중 0이 아닌것은 모두 소수값을 가졌다. 본 연구에서는 이러한 상황에서 변수고정을 진행해 나가기 위해 다음과 같은 기준을 사용하였다. 첫째로, 선형완화식에서 0.5보다 큰 값을 가지는 변수는 모두 1로 고정하였다. 둘째로, 0.5보다 큰 값을 가지는 변수가 없을 경우에는 그 중 큰 값을 가지는 순서에 따라 몇 개의 변수들을 1로 고정하였다. 이때, 너무 작은 값을 가지는 변수가 1로 고정되는 것을 방지하기 위해 고정되는 변수들은 특정한 값 이상이 되어야 한다는 제약을 두었다. 만약 이러한 기준으로도 고정시킬 수 있는 변수가 하나도 없을 때에는 그중 가장 큰 변수 하나를 1로 고정시켜 변수고정이 계속 진행될 수 있도록 하였다.

변수의 값을 1로 고정한다는 것은 그 변수에 대응되는 일간운행구간을 승무계획에 포함하겠다는 의미이다. 고정된 일간운행구간에 포함된 단위운행구간과 관련된 제약식은 이미 만족되었기 때문에, 변수고정은 문제의 제약식의 수를 줄이는 효과가 있다. 따라서 변수를 고정한 후에는 선형완화식을 푸는 속도가 빨라지게 되며, 이러한 이유로 정수해를 빠른 시간 내에 찾는 것이 가능하게 된다.

변수고정법을 무리하게 사용하면, 변수고정 후 문제의 최적정수해를 얻을 수 없는 상황이 발생할 수도 있다. 변수고정 후 선형완화식의 최적값이 갑자기 증가하는 현상이 관찰되면 무리한 변수고정의 가능성을 의심해 봐야 한다. 이럴 경우에는 한번에 고정되는 변수의 수를 줄이고, 변수고정이 되기 위한 하한한계치의 값을 높이는 등 변수고정의 기준을 좀더 강화한 알고리듬을 적용하였다.

변수고정 후에는 선형완화식을 열생성기법을 사용하여 다시 최적화한다. 선형완화식의 최적화가 이루어지면 다시 고정할 변수들을 찾아서 변수고정을 시행하고 그 선형완화식을 최적화하는 과정을 반복한다. 앞서 언급한 바와 같이 본 연구에서 선형완화식을 최적화 할때 열삽입의 회수가 제한치를 초과하게 되면, 더이상의 선형완화식의 최적화를 중지한 후 근사최적해를 이용하여 변수고정을 하였다. 변수고정법은 선형완화식의 해가 모두 정수값을 가질 때까지 반복되며, 이러한 과정을 거쳐 정수해를 하나 얻을 수 있게 된다. 아래는 본 문제의 알고리듬을 정리한 것이다.

단계 1(초기화) : '고정여부' = No, '열삽입회수' = 0. 초기변수들을 생성한다.

단계 2(선형완화식 풀기) : 열제거 기준에 따라 변수제거를 수행하고 선형완화식을 최적화한다.

단계 3(열생성문제 풀기) :

'고정여부' 값이 Yes이고 열삽입회수가 기준

치보다 높은 경우 단계 5로 넘어간다.
'고정여부' 값이 No이고 선형완화식의 변수 개수가 기준치 이상인 경우 단계 5로 넘어간다. 쌍대변수에 기반하여 열생성문제인 최장경로문제를 해결한다. 최장경로 중 (SUB) 값이 1보다 큰 것이 있으면 단계 4로, 그렇지 않으면 단계 5로 넘어간다.

단계 4(열 삽입) : '열삽입회수' 1 증가.
선형완화식에 기준을 만족시키는 열들을 추가한 후 단계 2로 넘어간다.

단계 5(변수고정) : '고정여부' = Yes, '열삽입회수' = 0.
변수고정 기준을 만족시키는 변수들을 고정한다. 고정되는 변수가 있을 경우 단계 2로 넘어가며, 고정할 변수가 없는 경우 단계 6으로 넘어간다.

단계 6(종료) : 알고리즘 종료.

실제 알고리듬의 구현은 CPLEX 4.0에서 제공하는 라이브러리들을 사용하였다. 구체적으로 언급하면 단계 2의 선형완화식의 풀이는 CPLEX에서 제공하는 선형계획루틴을 사용하였으며, 단계 4의 열삽입은 CPLEX의 변수추가루틴을, 단계 5의 변수고정은 CPLEX의 변수한계치조정루틴을 사용하여 구현하였다.

5. 실험 결과

앞장에서 설명한 알고리듬을 실제 지하철 승무계획문제에 적용해 보았다. 본 연구에서는 국내 모 지하철 공사에서 실제 운행되고 있는 814개의 단위운행구간을 갖는 문제에 대해 실험을 행하였다. 하루에 운행해야 하는 814개의 단위운행구간의 시간표가 주어진 상황에서 앞서 언급한 제약조건들을 만족하면서 각각의 단위운행구간에 승무원을 배치하여야 하는 상황인 것이다. 문제의 선형완화식의 풀이와 변수고정 후의 한계변수선형식의 풀이는 모두 CPLEX 4.0에서 제공하는 라이브러리들을 사용하였다. 하나의 일간운행구간이 10개의 단위운행구간으로 구성되므로 본 문제의 경우 최소한 82개 이상의 일간운행구간이 필요하며, 실험결과 82개의 일간운행구간으로 구성된 승무계획을 구해낼수 있었다. 즉, 실제문제에 대하여 앞서 제시한 알고리듬을 적용한 결과 최적의 승무계획을 작성해 낼 수 있었다.

5.1 파라메터

본 장에서는 지하철 승무계획문제와 그의 해법에 관련된 파라메터들에 대해 설명하고자 한다. 우선 지하철 승무계획문제의 문제상황과 관련하여 주문제 (MP)에 나타 파라메터들은 다음과 같다.

U_A : 승무기지 A에서 출발하는 일간운행구간 수의 상한한계치

U_B : 승무기지 B에서 출발하는 일간운행구간 수의 상한한계치

U_O : 야간일간운행구간 수의 상한한계치

지하철 승무계획문제에서 중요하게 생각하는 파라메터는 U_O의 값이다. 이 값이 작을수록 야간일간운행구간의 수가 줄어들어 승무원들이 작업하기에 편한 상황이 되지만, 그만큼 문제의 해를 구하기는 어렵게 된다. 본 논문에서는 U_O의 값을 18~20으로 조정하면서 문제상황을 바꾸어 보았다. U_O의 값이 18보다 작은 경우에는 선형완화식조차 가능해가 없는 상황으로, 이는 야간일간운행구간 제약을 만족하는 승무계획이 존재하지 않음을 의미한다. U_A와 U_B의 값은 그 중요성이 높지 않아 고정된 값을 사용하였다.

열생성과 변수고정법의 구현과 관련해서도 몇가지 한계치와 파라메터들이 필요하다. 이와 관련된 파라메터들은 지하철 승무계획문제의 문제상황과는 무관한 것들로 단지 문제해법과 관련한 전략적인 결정에 관한 것들이다. 이러한 파라메터 값들을 적절히 선택하여 문제의 최적해를 구하는 동시에 문제해결에 필요한 시간을 최소화하여야 한다. 파라메터 값들을 잘못 정하면 문제의 최적정수해

를 구하지 못하는 상황이 발생하거나, 반대로 최적 정수해는 구하지만 해를 구하는 시간이 매우 길어지는 상황이 발생하기도 한다. 일단 파라메터들이 정해지고, 이에 기반하여 본 논문에서 제시한 알고리듬을 작동시키면 그 결과 하나의 정수해를 얻을 수 있게 되며, 알고리듬의 확정적(deterministic) 특성으로 인해 파라메터를 변화시키지 않는 한 같은 근을 얻게 된다.

열삽입개수상한 : 한 번의 열삽입에 들어가는 열 수의 상한한계치.

열삽입회수상한 : 한 번의 변수고정 후 열삽입 회수의 상한한계치.

변수고정기준치 : 선형완화식에서 변수고정기준치보다 큰 값을 가지는 변수들은 모두 1로 고정.

고정변수개수 : 변수고정기준치를 만족하는 변수가 없을 경우 변수값에 따라 변수고정을 수행할때 한 번에 고정되는 변수 수의 상한한계치.

변수고정하한치 : 변수고정이 되기 위한 변수 값의 하한한계치(변수고정하한치보다 큰 값을 가지는 변수가 하나도 없을 시에는, 가장 큰 값을 가지는 변수 하나를 1로 고정함).

5.2 테스트 결과

실제 814개의 단위운행구간으로 구성된 지하철 승무계획문제를 풀어 본 결과가 <표 1>에 요약되

〈표 1〉 실험 결과

실 험 번 호	1-1	2-1	2-2	2-3	3-1	3-2	3-3
U_A	50	50	50	50	50	50	50
U_B	50	50	50	50	50	50	50
U_O	200	19	19	19	18	18	18
열삽입개수상한	40	40	100	50	40	40	50
선형완화치	81.4	81.4	81.4	81.4	81.52	81.52	81.535
완화식최적여부	Y	Y	Y	Y	N	N	N
변수개수(고정전)	3175	7525	10175	10175	11935	11935	12025
계산시간(고정전)	131m	250m	180m	230m	482m	482m	480m
변수제거(고정전)	N	N	N	Y	N	N	Y
열삽입회수상한	8	8	15	15	8	12	15
변수고정기준치	0.5	0.5	0.5	0.5	0.5	0.5	0.5
고정변수개수	5	1	2	2	2	1	2
변수고정하한치	N	N	0.3	0.3	N	N	0.3
정수근	85	84	82	82	83	82	82
정수근최적여부	N	N	Y	Y	N	Y	Y
변수제거(고정후)	N	N	Y	N	N	N	N
변수개수(고정후)	14583	23450	2223	15647	25471	26530	13831
계산시간(고정후)	212m	870m	744m	270m	270h	670m	198m

어 있다. 같은 문제에 대해서도 파라메터들을 여러 가지 상황으로 변화시켜 보면서 어떠한 파라메터 값에서 문제의 최적해를 신속하게 구할 수 있는지를 비교해 보았으며, 총 7개의 파라메터 조합에 대한 실험결과가 요약되어져 있다. <표 1>의 결과는 Pentium-333 시스템에서 CPLEX 4.0을 사용하여 얻은 결과이다. 실험결과 열생성기법과 변수고정 방법을 이용하여 최적정수해를 구할 수 있었다.

실 험 번 호	: (문제번호) - (파라메터 변화에 따라 주어지는 번호)
선형완화치(고정전)	: 변수고정 전의 선형완화식을 풀어 얻는 근사최적해
완화식최적여부	: 변수고정 전 얻은 선형완화치의 최적여부
변수개수(고정전)	: 변수고정 전의 선형완화식에 쓰인 변수의 수
계산시간(고정전)	: 변수고정 전의 선형완화식을 푸는 데 걸린 시간
변수제거(고정전)	: 변수고정 전의 선형완화식 최적화 후 변수제거 여부
정 수 근	: 얻은 정수해의 값
정수근최적여부	: 얻은 정수해의 최적여부
변수제거(고정후)	: 변수고정 후의 선형완화식 최적화 후 변수제거 여부
변수개수(고정후)	: 정수해를 얻고 나서 남은 변수의 개수
계산시간(고정후)	: 변수고정 후 정수해를 얻을 때까지 걸린 시간

야간일간운행구간 수의 상한한계치가 작을 경우 변수고정 이전의 선형완화식의 최적해를 구하는 데에도 많은 시간이 걸린다(실험 3-1, 실험 3-2, 실험 3-3). 하지만 변수고정 이후에 정수해를 구하기까지 걸린 시간은 매우 짧았다. 정수해를 하나 얻기 위해서는 수십 번의 변수고정 과정을 거쳐야 하는데 이것을 수십번의 고정된 선형완화식을 풀어야 한다는 것을 의미한다. 즉 최악의 경우 정수해를 하나 구하는데 걸리는 시간이 선형완화식의 최적해를 구하는 시간의 수십배가 될 수도 있다. 그러나, <표 1>에서는 변수고정후 정수해를 하나 구할 때 까지 걸리는 시간이 변수고정 전의 선형완화식을 최적화하는데 걸린 시간의 절반보다 짧은 경우도 있다는 것을 보여주고 있다.

열제거는 문제의 풀이시간을 상당히 줄여주었다. 변수고정 이전의 선형완화식의 최적해에서 0의 값을 가지는 변수를 제거한 결과 문제의 해를 훨씬 빠른 시간안에 구해낼 수 있다는 것을 실험 2-3과 실험 3-3을 통해 확인할 수 있다. 하지만 변수고정 단계에서 열제거를 사용하는 것은 문제풀이의 시간에 좋지 않은 영향을 끼치기도 하였다(실험 2-2). 실험결과 한번에 고정되는 변수의 수가 작고, 한번의 변수고정 후에 수행되는 열삽입회수가 많을 수록 좋은 해를 준다는 것을 알 수 있었다. 본 논문에서 제시한 알고리듬이 항상 문제의 최적해를 보장하는 것은 아니지만 실험결과 열생성, 변수고정법, 열제거기법을 사용하여 문제의 최적해들을 구해낼 수 있었다.

지하철 승무계획 알고리듬을 변형된 문제에 적용한 결과가 <표 2>에 나타나 있다. 변형된 문제는 기존 문제와 다른 상황은 동일하나 야간일간운행구간의 총작업시간제약을 더 강화한 것으로, LATI 값이 2시간 짧아진 문제이다. 변형된 문제는 실제 지하철 운행에 있어서 야간일간운행구간의 작업시간을 줄이는 것이 승무원의 피로도를 줄여 작업환경을 개선하는데 크게 기여하는 상황을 반영한 것이다. <표 2>의 결과는 Pentium-400 시스템에서 CPLEX 4.0 을 사용하여 얻은 결과이다. 변형된 문제의 경우 변수고정 이전의 선형완화식의 최적값이 증가하였음을 알수 있다. 이는 제약이 강화되면서 선형완화식조차 최적값이 증가하는 상황을 나타내는 것이다. 하지만 이러한 상황에서도 변수고정의 기준을 강화하여 알고리듬을 적용한 결과 문제의 풀이시간이 늘어나기는 하였으나 역시 최적정수해를 구해낼 수 있었다.

〈표 2〉 변형된 문제의 실험결과

실 험 번 호	4-1	4-2	4-3
U_A	50	50	50
U_B	50	50	50
U_O	18	18	18
열삽입개수상한	50	50	50
선형완화치	81.8	81.8	81.8
완화식최적여부	Y	Y	Y
변수개수(고정전)	8575	8575	8575
계산시간(고정전)	245m	245m	245m
변수제거(고정전)	Y	Y	Y
열삽입회수상한	20	9	20
변수고정기준치	0.5	0.5	0.5
고정변수개수	2	1	1
변수고정하한치	0.3	0.3	0.3
정수근	86	83	82
정수근최적여부	N	N	Y
변수제거(고정후)	N	N	N
변수개수(고정후)	25499	22729	23358
계산시간(고정후)	480m	457m	509m

6. 결 론

본 연구에서는 문제의 크기가 매우 큰 지하철 승무계획문제를 푸는 해법을 제안하였다. 지수적으로 많은 수의 변수를 가지는 문제의 선형완화식은 다항식의 복잡성을 가지는 열생성문제을 사용하여 최적화하였다. 열생성문제는 무회로 계층구조의 그래프에서 최장경로를 찾는 문제로 변형되어 다항시간내에 풀이가 가능하였다. 또한, 본 연구에서는 기본적인 변수고정법을 사용하여 지하철승무계획문제의 최적정수해를 빠른 시간에 찾아줌으로써, 변수고정법이 지하철승무계획문제에 대하여 최적해를 찾는 매우 실용적인 방법임을 보여주고 있다.

현재 지하철회사에서 승무계획 작성하는 방식에 따라 수립된 승무계획은 일간운행구간이 만족시켜야 하는 여러 제약을 충족시키지 못하는 일간운행구간이 30~40%에 이르며, 그러한 상황에서도 본 논문에서 제시한 최적정수해보다 많은 수의 승무원인 84명의 승무원을 필요로 한다.

본 연구에서 제안한 알고리듬은 승무일정을 새로 작성하는 데에만 사용되는 것이 아니라 현재 운행중인 승무일정을 부분적으로 개선하는데 사용할 수 있는 것으로 보인다. 즉 이미 운행중인 승무일정을 초기해로 사용하여 승무일정을 보다 빠른 시간에 개선해 낼 수도 있는 것이다. 이러한 모든 과정들은 현재 산업현장의 승무계획과정보다 수십배 이상 빠르며 이는 승무일정의 유연한 변화를 가능하게 한다. 작업조건이나 승무조건의 변화가 발생할 때마다 쉽게 승무일정을 변경할 수 있고 결과적으로 작업자나 회사의 요구사항을 현장에 직접적으로 반영해 낼 수 있게 된다.

본 연구에서 제시한 알고리듬에 나타나는 근사최적해의 사용과 열제거 기법은 문제의 풀이시간을 매우 단축시켜 주었다. 이러한 이점으로 인해 근사최적해의 사용과 관련하여 본 연구에서 사용된 방법외에 보다 진보된 방법에 대한 연구가 필요할 것으로 보이며, 열제거에 대한 다양한 실험들이 필요한 것으로 보인다. 또한 각 변수에 주어지는 비용모델의 변화도 연구해 볼 내용으로 생각된다. 마지막으로 본 연구는 지하철의 구조가 선형인 상황에 대한 해법이므로, 보다 일반적인 구조를 가지는 경우에 대한 연구가 필요하다고 생각된다.

참 고 문 헌

[1] Anbil, R., E.R. Tanga and E.L. Johnson, "A global optimization approach to crew scheduling," *IBM systems J.*, Vol.31(1991), pp. 71-78.

[2] Ahuja, R.K., T.L. Magnanti and J.B. Orlin, *Network flows : Theory, Algorithms, and Applications*, Prentice-Hall, NJ, 1993.

[3] Ball, M., L. Bodin and R. Dial, "A matching

based heuristic for scheduling mass transit crews and vehicles," *Transportation Science*, Vol.17(1983), pp.4-31.

[4] Ball, M. and A. Roberts, "A graph partitioning approach to airline crew scheduling," *Transportation Science*, Vol.19(1985), pp.107-126.

[5] Barnhart, C., E.L. Johnson, G.L. Nemhauser and M.W.P. Savelsbergh, P.H. Vance, "Branch-and-Price : Column generation for solving huge integer programs," *Operations Research*, Vol.46(1998), pp.316-329.

[6] Beasley, J.E. and B. Cao, "A tree search algorithm for the crew scheduling problem," *European journal of Operational Research*," Vol.94(1996). pp.517-526.

[7] Bianco, L., M. Bielli, A. Mingozzi, S. Ricciardelli, M. Spadoni, "A heuristic procedure for the crew rostering problem," *European Journal of Operational Research*, Vol.58(1992), pp.272-283.

[8] Caprara, A., M. Fischetti, P. Toth and D. Vigo, "Algorithms for railway crew managements," *Mathematical Programming*, Vol.79(1997), pp.125-141.

[9] Crainic, T.G. and J.M. Rousseau, "The column generation principle and the airline crew scheduling problem," *INFOR*, Vol.25 (1987), pp.136-151.

[10] Desrochers, M. and F. Soumis, "A column generation approach to the urban transit crew scheduling problem," *Transportation Science*, Vol.23(1989), pp.1-13.

[11] Falkner, J.C. and D.M. Ryan, "A bus crew scheduling system using a set partitioning model," *Asia-Pacific Journal of Operational Research*, Vol.4(1987), pp.39-56.

[12] Gershkoff, I., "Optimizing flight crew schedules," *Interfaces*, Vol.5(1989). pp.29-43.

[13] Hoffman, K. and M. Padberg, "Solving airline crew scheduling problems by branch-and-cut," *Management Science*, Vol.39 (1993), pp.657-682.

[14] Lavoie, S., M. Minoux and E. Odier, "A new approach for crew pairing problems by column generation with an application to air transportation," *European Journal of Operational Research*, Vol.35(1988), pp.45-58.

[15] Marsten, F. and Shepardson, "Exact solution of crew scheduling problems using the set partitioning model : recent successful applications," *Networks*, Vol.11(1981), pp. 165-177.

[16] Martello, S. and P. Toth, "A heuristic approach to the bus driver scheduling problem," *European Journal of Operational Research*, Vol.24(1986), pp.106-117.

[17] Miller, J.L. and Franz, L.S, "A binary-rounding heuristic for multi-period variable-task-duration assignment problems," *Computers & Operations Research*, Vol.23 (1996), pp.819-828.

[18] Mingozzi, A., M.A. Boschetti, S. Ricciardelli and L. Bianco, "A set partitioning approach to the crew scheduling problem," *Operations Research*, Vol.47(1999), pp.873-888.

[19] Paias, A. and J. Paixao, "State space relaxation for set covering problems related tobus driver schedule," *European Journal of Operational Research*, Vol.71(1993), pp. 303-316.

[20] Paixao, J. and M. Pato, "A structural lagrangean relaxation for two-duty period bus driver scheduling problems," *European Jorunal of Operational Research*, Vol.39

(1989), pp.213-222.

[21] Park, K., S. Kang and S. Park, "An integer programing approach to the bandwidth packing problem," *Management Science*, Vol.42(1996), pp.1277-1291.

[22] Savelsbergh, M., "A Branch-and-Price algorithm for the generalized assignment problem," *Operations Research*, Vol.45 (1997), pp.831-841.

[23] Vance, P.H., C. Barnhart, E.L. Johnson and G.L. Nemhauser, "Airline crew scheduling : A new formulation and decomposition algorithm," *Operations Research*, Vol.45 (1997), pp.188-200.