최소차량운용문제에 대한 실용적 해법
A pragmatic algorithm for the Minimum Railway Stock Maintenance Routing Problem

홍성필* , 김경민**, 이경식***, 박범환**, 홍순흠**

* 서울대학교 산업공학과
** 한국철도기술연구원 철도정책물류연구본부
*** 한국외국어대학교 산업정보시스템공학부

Abstract

Given a schedule of train to be routed by a Railway Stock, Railway Stock Routing Problem determine a sequence of train while satisfying turnaround time and maintenance restrictions.

The objective is to minimize the Railway Stock during a week and each day simultaneously. And we prove that Railway Stock Routing Problem with maintenance restrictions is NP-hard.

In this paper, we present two stage approaches that solve the Railway Stock Routing Problem in a reasonable time. In first stage we relax maintenance restrictions and formulate as a Min-cost-flow problem. Then, in the second stage, we attempt to satisfy maintenance restrictions using ours heuristic algorithm. We show the computational result of applying to an actual train schedule data.

Keywords: Railway Stock Routing Problem, Minimum cost flow , Integer programming, Heuristic algorithm

1. 서 론

본 연구에서는 철도 분야의 차량 운용 계획문제를 다루고자 한다. 일반적으로 철도 차량 운용 계획 문제는 최소 비용으로 일정한 주기를 가지는 운행 계획을 완수하도록 운용을 생성 하는 것이다. 즉, 하나의 물리적 차량이 담당해야 할 일련의 열차들의 순서를 구성하는 계획이며 차량운용 계획은 열차만이 아니라 역에서의 선회(Turnaround) 시간과 주기적인 검수 조건을 고려해야 하며 최적화된 차량 운용계획을 통해 차량 소요량을 최소화 하는 것을 목적으로 한다. 또한 차량 운용계획 문제는 차단유형이 할당된 열차 스케줄(Fleet Assignment 모형 결과)과 검수 관련한 데이터들이 주어진 상황에서 각 차단유형 별로 이루어지게 된다. 검수 조건들은 차량 운용 계획문제의 해결을 어렵게 한다고 알려져 있다. 이러한 차량 운용계획 문제는 여러 분야에서 적용되고 있으며 특히 대형항공사나 철도회사의 경우에 있어서 많은 연구가 수행되어 왔다. 차량 운용계획 문제의 응용을 통해서 실제 차량 소요량을 정량적으로 계산 할 수 있으며 열차 스케줄의 갱신에 따른 차량 운용계획 작성의 자동화를 통해 다변하는 수요에 대응 가능해진다. 나아가 차량의 효율적인 사용과 안전성 문제를 해결 할 수 있다.

1.1 문제 정의

먼저 본 연구에서 사용되는 용어들을 정의하기로 한다. 열차(Train)는 항공기의 편(Flight)의 개념처럼 출발역, 도착역, 출발시간, 도착시간 등의 기본적인 정보를 가지고 있는 서비스를 의미한다. 차량(Railway Stock)은 객차와 동력차 합쳐있는 물리적 의미의 열차이라 하며, 운용(Routing)은 하나의 차량이 담당해야 할 일련의 열차들의 순서이다. 회송열차(Deadhead Train)는 승객을 태우지 않고, 검수 등 다른 이유로 인해 역과 역 사이를 운행하는 열차를 의미하며 본 연구에서는 회송열차를 허용하지 않는다.

검수 조건을 자세히 설명하면 다음과 같다. 실제 철도 산업 현장에서 이루어지는 차량의 검수 조건은 거리와 요일 두 가지로써 두 가지 모두를 만족시켜야한다. 검수의 종류는 가장 빈번히 일어나는 일상검수(3500km, 3Days)로부터 전반검수(800,000km, 1Year)까지 6가지 정도이다. 하지만 1년의 열차 스케줄로 운용계획을 세우는 것은 현실적으로 불가능하므로 차량 이력 사항이 없는 상태에서 모든 검수 조건을 고려한다는 것은 의미 없는 일이다. 이는 하나의 차량이 어떤 운용에 투입되었을 때, 다양한 검수 조건들과 차량 운행 및 검수 이력에 의해 차량의 교체가 불가피하므로 그 차량이 그 운용에 있는 모든 열차를 소화한다고는 볼 수 없기 때문이다. 그러므로 본 연구에서는 일상검수만 고려하기로 한다. 또한, 검수는 지정된 검수역에서 자정부터 새벽 4시까지로 운행 중 낮에 검수역으로 회송을 허용하지 않는다. 즉, 본 연구에서는 각 생성되는 운용이 검수역 숙박과 검수역의 숙박 사이에 3500km, 3Days를 만족시키면 되는 것이다.

1.2 문제 유형 분류

차량 운용계획 문제는 최소 비용으로 일정한 주기를 가지는 운행계획을 완수하도록 운용을 생성 하는 것이지만 문제에 대한 이해와 목적에 따라서 조금씩 다른 특징을 가지고 있으며 해결 방법의 차이를 보인다. 본 연구에서는 이를 다음과 같이 세 가지로 분류 하고자 한다.

1.2.1 차량 경로(Path)계획

이 문제는 단순히 열차를 완수하도록 운용을 생성한다. 즉, 일정한 주기를 가지는 운행계획을 만족시키지는 않는 경우이다. 그러므로 운용 계획 주기의 역별 시종착 운용의 개수는 무시한다. 그림 1.1은 차량 경로계획에 대한 예제이다.

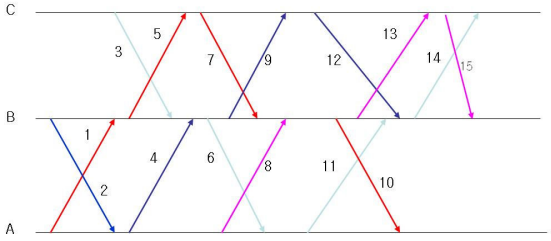

그림 1.1 (예제) 차량 경로계획

위의 그림과 같이 15개의 열차를 가지고 있을 때 우리는 다음과 같이 열차를 완수하는 운용 4가지를 찾을 수 있다. (1-5-7-10:A-A),(2-4-9-12:B-B), (3-6-11-14:C-C), (8-13-15:A-B). 하지만 이 운용들은 역별 시종착 운용의 개수가 불일치하므로 회송열차 없이는 일정한 주기를 만족 시킬 수 없다. 즉, B역에서 A역으로의 회송 열차가 추가 되어야 한다. 기존 해법은 후보 운용을 미리 생성한 후 Set Partitioning모형 적용한다. 하지만 이 해법은 부분 최적해(local optimum)의 가능성이 존재한다.

1.2.2 차량 반복(repetition)계획

이 문제는 차량 경로계획에 역별 시종착 운용의 개수를 일치 시켜주는 제약을 추가하여 일정한 주기를 가지는 운행계획을 생성하는 것이다. 그림 1.2는 차량 반복계획에 대한 예제이다.

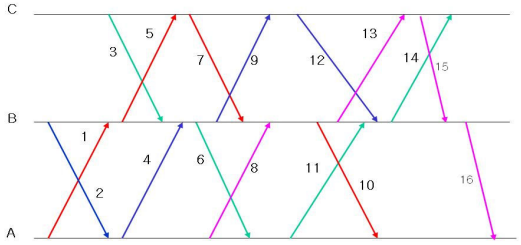

그림 1.2 (예제) 차량 반복계획

그림 1.2와 같이 16개의 열차를 가지고 있을 때 우리는 다음과 같이 운용 4가지를 찾을 수 있다. (1-5-7-10:A-A),(2-4-9-12:B-B),(3-6-11-14:C-C),(8-13-15-16 :A-A). 이 운용들은 역별 시종착 운용의 개수가 일치한다. 즉, 일정한 주기를 가질 수 있다. 기존 해법은 Set Partitioning모형에 시종착 개수 일치 조건제약을 추가한다. 하지만 이 해법은 일반적으로 다항시간을 보장 할 수 없다. 그러므로 열차 네트워크의 특별한 구조를 고려한 효율적인 모형이 필요하다.

차량 반복계획과 차량 경로계획은 열차 스케줄에 따라서 구분되어진다. 열차 스케줄에서 출, 도착 시간을 고려하지 않고 출, 도착역 만을 고려하여 그래프를 구성하였을 때 오일러 투어가 존재한다면 차량 반복계획을 할 수 있다. 오일러 투어가 존재하지 않는다면 차량 경로계획만 가능하다.

1.2.1 차량 순환(rotation)계획

이 문제는 운용이 열차 스케줄 완수와 일정한 주기를 가지는 것 뿐 만아니라 궁극적으로 모든 차량이 동일한 루트를 따라 가도록 운용을 구성. 장기적인 관점에서 모든 차량의 주행 거리 및 주행 경로를 동일하게 구성 하는 것이다. 이 문제는 실제 산업에서 차량의 마모도와 선로의 관리 등을 용이하게 한다. 그림 1.3은 차량 순환계획에 대한 예제이다.

위의 차량 반복계획과 동일한 16개의 열차를 가지고 있을 때 그림 1.3과 같이 순환하는 구조로 열차를 구성한다. (1-5-7-13-15-16)-(4-9-12-14)-(3-6-11)-(2-8-10)-(1-*-*-*)… 이 4가지의 운용을 순서대로 따라가면 주어진 열차를 완수하고 일정한 주기를 만족시키며 장기적인 관점에서 모든 차량이 동일하게 운행된다. 이 문제는 Traveling Salesman Problem으로 모형화가 가능하다.

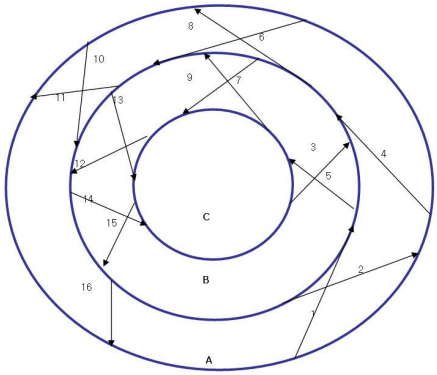

그림 1.3 (예제) 차량 순환계획

1.3 기존 연구 및 연구 성과

지금까지 차량 운용계획 문제는 주로 두 단계의 문제로 분해하여 접근하였다. 첫 단계는 한 차량이 하루에 운행할 수 있는 조건을 만족하는 몇 개의 열차를 조합하는 문제이며, 두 번째 단계는 하루 운행들을 여러 개 연결하여 실제로 한 차량의 운용을 생성하는 것이다. 하지만 두 번째 단계에서 만들어지는 운행 가능한 운용의 수는 하루 열차 수에

비례해 조합적으로 증가해 문제의 최적해를 구하는 것이 매우 어려워진다.

차량 운용계획 문제와 관련하여 다양한 해법들이 개발되어져 왔다. Kabbani *et al.*[7]는 American Airline사의 maintenance routing problem에 대한 연구로써 Aircraft는 3일 주기의 검수조건을 만족 시켜야 한다. 집합 분할(set partitioning)문제로 모형화 하였으며 일주일 주기를 가지는 운용을 다루고 있다. 이들은 하루 운용에 대한 pseudo-costs to penalize 일간운용을 개발하였고 이를 이용하여 첫 번째로 일간 근사 운용을 구하고 두 번째로 이러한 일간 운용을 연결하는 방법으로 해결하였다. Gopalan *et al.*[6]는 US Air사의 maintenance routing problem에 대한 연구로 Aircraft는 3일주기의 검수조건을 만족 시켜야 한다. 이들은 하루 운용을 FIFO나 LIFO등의 간단한 규칙을 사용하여 구한 후에 이러한 일간 운용들이 3일 검수 조건을 만족시키면서 연결되도록 하였다. 위의 두 연구는 차량 반복계획으로 분류되어진다. 또한 Barnhart *et al.*[1]은 검수 조건을 만족시키는 String들을 구하고 Branch and Price 방법으로 String들을 연결시켰다.

Clarke et al.[3]은 Traveling salesman 문제로 모형화 하였으며 문제 해결시 생성되는 Subtour에 대해서는 Lagrangian relaxation과 휴리스틱 방법을 이용하여 해결 하였다. Talluri[9]는 k-Maintenance Euler Tour를 정의 하고 3-MET 문제에 대해서는 다항시간 알고리즘을 제시하였으며 $k \geq 4$, k-MET 문제는 NP-complete임을 증명하였다. 또한 Zhao *et al.*[11]는 일간 검수를 random choice하는 TSP 휴리스틱 기법을 사용하였다. 이 연구들은 차량 순환계획으로 분류 할 수 있다. 또한 Schrijver[8]와 ForBes *et al.*[4]는 철도 차량의 하루 운용에 대하여 차량 유형에 따라 Min-cost-flow 문제와 Multi-commodity circulation 문제로 모형화 하였다. 이 방법은 다른 해법과 비교 하였을 때 계산적 효율성을 가지고 있지만 검수에 대한 고려가 없었다. 열생성기법을 이용하여 실제로 선형완화식의 최적해를 구하는 방식도 제안되었다. 열생성기법을 사용하면 모든 가능한 운용 중 일부만으로 문제의 해를 구하지만, 그 해는 모든 가능한 운용들을 고려한 문제를 풀었을 때 구한 해의 하한 값을 준다. 열생성기법의 일반적인 접근법에 대해서는 Barnhart *et al.*[2]에 나와 있으며 Decomposition Algorithm을 사용하여 Vance *et al.*[10]은 승무계획 문제에 대해서 좀 더 나은 모형과 해법을 제시하였다. 최근의 연구로 Maroti *et al.*[5]은 1 ~3 days 검수 조건을 가지고 있는 철도차량 운용문제에 대해서 정수계획법으로 수리 모형화 하였으며 Interchage Model을 제시하였고 Interchange Model이 가능해를 가지는지에 관한 문제가 NP-complete임을 증명하였다.

1.4 연구 범위 및 구성

본 연구에서는 일주일을 운행계획의 단위의 주기로 하며 첫 번째 목적은 주간 최대 차량 소요량을 최소화 하는 것이다. 즉, 최소 비용을 차량의 소요량으로 생각한다. 두 번째 목적으로는 매 요일 사용되는 차량 소요량도 동시에 최소화 하는 것을 목적으로 한다. 또한 검수 조건으로는 기존 연구와는 다르게 거리, 요일 두 가지를 동시에 만족시켜야한다.

본 연구의 구성은 다음과 같다. 2장에서는 본 연구에서 다루고자하는 문제의 계산 복잡성에 대한 고찰과 , 3장에서는 검수 조건을 고려하지 않을 때의 문제의 수리 모형 및 해법을 제시한다. 4장에서는 검수 조건 고려시의 문제의 수리 모형 및 해법을 제시하고, 5장에는 실제문제에 대하여 앞에서 제시한 해법을 적용한 실험 결과를 정리되어 있으며, 마지막으로 6장에서 결론과 향후과제에 대해 언급하였다.

2. 계산 복잡도

본 연구에서 다루고자 하는 철도 차량 운용 계획 문제가 검수 조건(시간 혹은 거리)를 가지는 경우 NP-hard의 범주에 속함을 보이고자 한다. 이를 위해 Partition 문제가 우리 문제로 다항시간 안에 변환(reduction) 됨을 보이자.

Partition 문제는 다음과 같이 정의 할 수 있다. n개의 원소로 이루어진 유한 집합 S에서 $a_j \in Z^+$, $j=1,\cdots,n$이고 $2A=\sum_{j=1}^{n} d_j$일 때 $\sum_{j \in S'} a_j = A$를 만족하는 부분집합 $S' \subseteq S$를 구하는 것이다. Partition 문제를 우리 문제로 다항시간에 변환되는 과정의 자세한 증명을 생략하고 그림 2.1로 대신한다. 본 연구의 문제인 거리 검수를 만족하는 2개의 운용이 존재 하는가에 대한 물음은 위에서 설명한 Partition 문제와 동일한 문제이다.

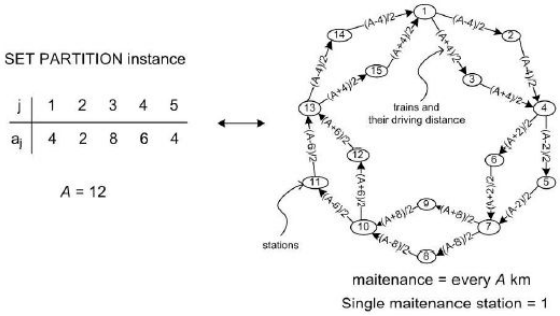

그림 2.1 변환 예제

3. 검수 조건 완화시 열차 반복 계획

본 장에서는 문제를 어렵게 만드는 검수 조건을 완

화하는 경우에 열차 반복계획을 해결하는 방법을 설명하고자 한다. 앞에서 언급한 것처럼 철도 차량 운용계획 문제를 본 연구에서는 일주일을 운행계획의 단위의 주기로 하며 첫 번째 목적은 주간 최대 차량 소요량을 최소화 하는 것이다. 두 번째 목적으로는 매 요일 사용되는 차량 소요량도 동시에 최소화하는 것이다. 그러므로 이를 주간 차량 소요량 최소화와 일간 차량 소요량 최소화로 나누어서 설명하고자 한다.

3.1 주간 차량 소요량 최소화

최소비용 흐름 모형으로 문제를 모형화하기 위해 다음과 같이 네트워크 모형을 구성한다. 하나의 열차를 출발 노드, 도착 노드 두 개로 나눈다. 각 노드에는 역과 시간 정보를 가지고 있다. 노드 사이에는 3가지 종류의 아크가 존재한다. 첫 번째로 Train 아크는 하나의 열차를 나눈 두 개의 노드 사이를 연결한다. 두 번째로 Forward Ground 아크는 연결조건을 만족하는 두 열차를 연결하는 역할을 한다. 즉, 역의 정보가 같고 선회시간을 만족하는 노드사이에 연결된다. 마지막으로 Backward Ground 아크는 차량 소요량을 계산하는 아크이다. 역의 정보가 같고 시간 정보가 앞선 노드 사이에 연결된다. 운용을 만들 때 Backward Ground 아크에 흐름이 생긴다면 이는 실제로 일주일 후의 운용과 연결된다고 생각하는 것이다.

그림 3.1은 주간 차량 소요량 최소화를 위한 네트워크 모형을 그린 것이다. (c_{ij}, l_{ij}, u_{ij})는 아크에서의 비용, 하한 값, 상한 값을 나타낸다. Train 아크의 경우에는 (0, 1, 1)로서 비용은 없으며 항상 흐름이 생겨야하므로 하한, 상한의 값이 모두 1이다. Forward Ground 아크는 (0, 0, 1)이며 Backward Ground 아크는 (1, 0, 1)이다.

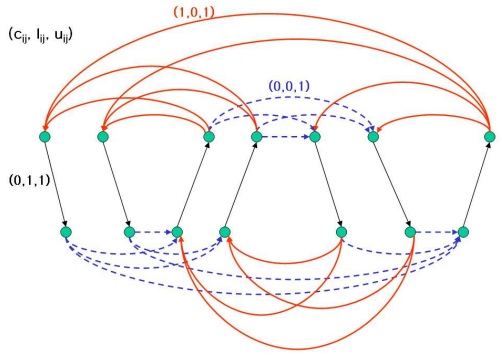

그림 3.1 기본 네트워크 모형

본 연구에서의 목적은 차량 소요량을 최소화하는 것으로 정수해를 얻는 것이다. 검수조건 완화시 열차 반복 계획을 최소비용 흐름 모형으로 문제를 수리모형화하면 Totally unimodularity의 성질을 만족 시키므로 정수해를 얻을 수 있는 장점을 가진다. 이에 관한 내용은 Wolsey에 나와 있다. 또한 최소비용 흐름 모형은 다항시간안에 최적해를 구할 수 있다는 장점을 가진다. 수리모형은 다음과 같다.

$$\min \sum_{(i,j)\in E_2} x_{ij}$$

$$s.t. \sum_{\{j:(i,j)\in E\}} x_{ij} - \sum_{\{j:(i,j)\in E\}} x_{ji} = 0, \quad i\in N \qquad (3.1)$$

$$0 \le x_{ij} \le 1 \quad (i,j)\in E_1 \cup E_2 \qquad (3.2)$$

$$1 \le x_{ij} \le 1 \quad (i,j)\in E_3 \qquad (3.3)$$

where, E_1: *Forward Ground* 아크집합

E_2: *Backward Ground* 아크집합

E_3: *Train* 아크집합

제약식 (3.1)은 각 노드의 flow blance에 관한 제약식으로 들어오는 흐름의 양과 나가는 흐름의 양의 합이 0을 만족 시켜야한다. 제약식 (3.2)와 (3.3)는 앞에서 설명한 아크의 종류별로 만족시켜야하는 하한, 상한의 값에 관한 제약식이다.

3.2 일간 차량 소요량 최소화

본 연구에서의 두 번째 목적으로는 매 요일 사용되는 차량 소요량도 동시에 최소화하는 것이다. 그 이유는 차량이 매 요일에 한번이라도 운행이 된다면 그에 따르는 비용이 발생하게 된다. 또한 차량 여유분이 늘어나므로 차량할당 측면에서 용이성을 가진다. 즉, 주간 차량 소요량은 유지하면서 각 요일별 차량 소요량을 최소화하면 매 요일에 운행되는 차량이 가능한 많은 열차들을 운행하고 하루 동안 운행되지 않는 차량은 해당 역에서 하루 동안 대기하고 있게 된다. 본 연구에서는 앞에서 설명한 주간 차량 소요량 최소화를 위한 네트워크 모형을 수정하여 일간 차량 소요량 최소화 문제를 해결하고자한다.

그림 3.2는 본 연구에서 기본 네트워크 모형의 수정에 대한 예를 보여준다. 먼저 각 역별로 하루 동안 휴식하는 가상의 열차를 매요일 생성한다. 예제 네트워크에서 운용을 생성하다보면 두 번째 날 서울과 세 번째 날 부산에서 하루 동안 휴식하는 운용이 생성됨을 알 수 있다. 기본 네트워크 모형을 유지하되 새로 추가되는 것은 다음과 같다. 두 개의 새로운 아크를 정의 한다. 먼저 Stay Train 아크는 하루 동안 운행하지 않고 역에 머무는 열차를 나타낸다. 만약 이 아크에 흐름이 생긴다면 일간 소요 차량수가 줄음을 의미한다. 두 번째로는 Stay Ground 아크이며 Stay train 아크끼리의 연결을 위해 사용된다. Stay Ground 아크가 생김으로써 이틀 이상 운용되지 않는 차량이 표현된다.

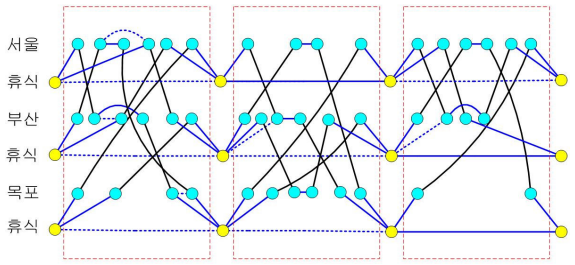

그림 3.2 (예제) 수정 네트워크 모형

그림 3.3은 일간 차량 소요량 최소화를 위한 네트워크 모형을 그린 것이다. 새로 생성되는 Stay Train 아크의 (c_{ij}, l_{ij}, u_{ij})값은 Minimum 문제이므로 목적함수 개선을 위해서 비용은 -1을 가지고 흐름의 상한 값은 제약이 없으며 하한 값은 0이상이 되어야 한다. 즉, $(-1, 0, \infty)$이다. Stay Ground 아크는 $(0, 0, \infty)$값을 가진다. 기존의 아크 중 Backward Ground 아크는 비용을 *Big-M*으로 변화 시켜주는데 그 이유는 새로 생성되는 Stay Train 아크의 비용이 -1이므로 주간 차량 소요량 최소화에 영향을 미치지 않게 하기 위해서이다. 또한 Train 아크와 Forward Ground 아크는 주간 차량 소요량 네트워크 모형과 동일한 (c_{ij}, l_{ij}, u_{ij})값을 가진다.

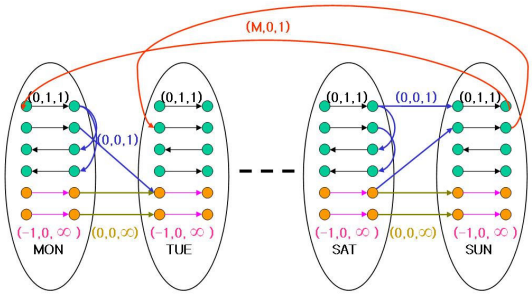

그림 3.3 수정 네트워크 모형

일간 차량 소요량 최소화 문제를 최소 비용 흐름 모형으로 수리모형화 하면 다음과 같다.

$$\min \sum_{(i,j)\in E_2 \cup E_5} x_{ij}$$

$$s.t. \sum_{\{j:(i,j)\in E\}} x_{ij} - \sum_{\{j:(i,j)\in E\}} x_{ji} = 0, \ i \in N \quad (3.4)$$

$$0 \le x_{ij} \le 1 \quad (i,j) \in E_1 \cup E_2 \quad (3.5)$$

$$1 \le x_{ij} \le 1 \quad (i,j) \in E_3 \quad (3.6)$$

$$0 \le x_{ij} \le \infty \quad (i,j) \in E_4 \cup E_5 \quad (3.7)$$

where, E_1:*Forward Ground* 아크집합
E_2:*Backward Ground* 아크집합
E_3:*Train* 아크집합
E_4:*Stay Ground* 아크집합
E_5:*Stay Train* 아크집합

제약식 (3.4)은 각 노드의 flow blance에 관한 제약식으로 들어오는 흐름의 양과 나가는 흐름의 양의 합이 0을 만족 시켜야한다. 제약식 (3.5), (3.6), (3.7)은 앞에서 설명한 아크의 종류별로 만족시켜야 하는 하한, 상한의 값에 관한 제약식이다.

4. 검수조건 고려시 열차 반복 계획

본 장에서는 거리와 시간 두 가지 검수 조건을 고려하는 경우에 열차 반복 계획을 해결하는 방법을 설명하고자 한다. 일주일을 운행계획의 단위의 주기로 하며 역은 검수가 가능한 역, 불가능한 역으로 나눌 수 있다. 검수는 지정된 검수 가능한 역에서 자정부터 새벽에 이루어진다. 또한 운행 중 낮에 검수역으로 회송을 허용하지 않는다. 즉, 본 연구에서는 각 생성되는 운용이 검수역 숙박과 검수역의 숙박사이가 3Days, 차량 운행거리는 3600km를 동시에 초과해서는 안 된다. 하지만 검수조건을 만족 시킬 수 없는 경우 밤에 검수역으로 회송을 허용하는 것으로 한다.

4.1 Crossover heuristic

그림 4.1을 보면 앞 장에서의 최소흐름비용 모형으로 구한 해를 통해 일간 운용은 결정되어진다. 우리는 일간운용의 시종착 역을 검수 가능역과 불가능역 4가지로 구분한다. 일간 운용을 검수가능역에서 시작과 끝을 o으로 표시하고 검수불가능역에서 시작과 끝을 x로 표시한다면 (o-o), (x-o), (o-x), (x-x) 4가지로 나타 낼 수 있다.

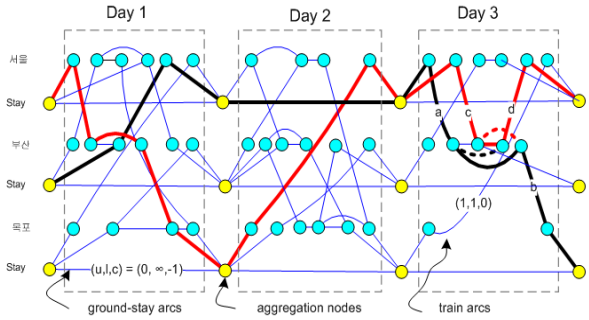

그림 4.1 ArcExchange

만약 R_1, R_2 두 개의 운용에서 (i_1, j_1), (i_2, j_2)인 열차가 다음과 같은 두 가지 조건을 만족시키면 ArcExchange((i_1, j_2), (i_2, j_1))가 가능하다.

(1) i_1, i_2 의 도착역이 같다.

(2) i_1의 도착시간 + 선회시간 ≤ j_2의 출발시간 and i_2의 도착시간 + 선회시간 ≤ j_1의 출발시간

예를 들어, 그림 4.1의 굵은 선이 하나의 운용을 나타낼 때 Day 3의 부산역에서 ArcExchange가 가능함을 볼 수 있다. $i_1 = a$, $j_1 = b$, $i_2 = c$, $j_2 = d$

본 연구에서 제안하는 Crossover heuristic은 크게 두 가지 Stage로 나뉜다. 첫째, (x-x)인 운용을 (o-o)인 운용과 Arc Exchange를 하여 (x-o), (o-x)운용으로 바꾼다. 둘째, (x-o), (o-x)인 운용을

(o-o)인 운용과 Arc Exchange를 통해서 운행거리를 줄인다. 예를 들어, 그림 4.2와 같은 (x-o) 일간 운용이 있을 때 각 열차가 끝나는 3가지 시점에서 Arc Exchange가 가능한 (o-o) 운용들을 찾아 Arc Exchange를 하였을 때 가장 (x-o)운용의 운행거리가 짧아지는 운용과 Arc Exchange를 한다.

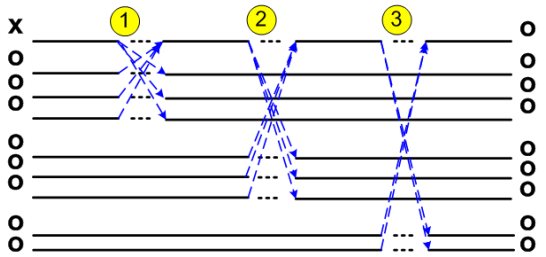

그림 4.2 Crossover heuristic

4.2 Algorithm

Step 1: 앞 장에서 설명한 최소흐름 비용 모형으로 검수 조건 완화시 열차 반복 계획 문제 푼다.

Step 2: Crossover Rule을 사용하여 운용을 생성한다.

Step 3: 생성된 운용이 검수조건을 만족하는지 체크한다. 검소조건을 만족하면 Step 7로 만족하지 않으면 Step 4로 간다.

Step 4: 현재 투입 할 수 있는 여유차량의 유무를 체크한다. 여유차량이 없으면 Step 5로 여유차량이 존재하면 Step 6으로 간다.

Step 5: 장거리 야간 회송열차를 이용하여 검수조건을 만족시킨다. Step 7로 간다.

Step 6: 최소 흐름 비용 모형에서 각 요일별 차량수를 현재의 차량소요량 + 1로 고정 시킨 후 검수조건 완화시 열차 반복 계획 문제를 푼다. 즉, 추가 차량 투입을 이용하여 평균 운행거리를 완화 시킨다. Step 2로 간다.

Step 7: 검수조건을 만족시키는 차량 반복계획 수립

그림 4.2는 위의 알고리즘을 정리 한 것 이다.

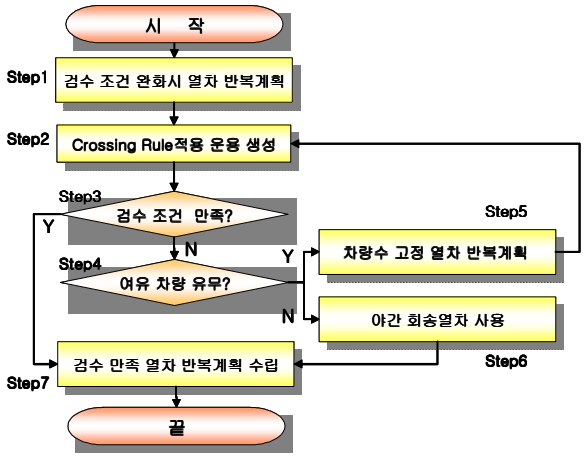

그림 4.2 알고리즘 개요

5. 실험

앞 장에서 설명한 수리모형과 알고리즘을 실제 철도 분야의 차량 운행계획 문제에 적용해 보았다. 본 연구에서는 국내에서 실제 운행되고 있는 총 972개의 열차 스케줄에 대해 일주일을 운용계획 단위의 주기로 실험 하였다. 요일 별 열차 수는 (월, 화, 수, 목, 금, 토, 일)=(135, 132, 132, 132, 137, 152, 152)로써 주중보다 주말의 열차가 많다. 이와 같이 요일별 열차 수 차이를 보면 본 연구에서 주간 차량소요량을 최소화 시키면서 동시에 일간 차량소요량을 최소화 시키는 것이 합리적인 목적이 됨을 알 수 있다. 또한 생성되는 각 운용은 일상검수(3500km, 3 Days) 만족하여야 하는 제약을 가지고 있다.

문제의 선형식의 풀이와 네트워크 구성은 Pentium 4 CPU 3.1GHz 시스템에서 CPLEX 9.0의 라이브러리들과 Visual C++ 6.0을 사용하였다.

5.1 차량 반복계획 실험결과

실제 972개의 단위 운행구간으로 구성된 차량 반복계획 문제를 풀어본 결과가 표 5.1에 정리 되어있다. 먼저 현재 철도 공사의 운용계획을 분석하여 요일별 차량소요량을 계산 할 수 있었다.

	차량소요량							검수조건위반			CPU (sec)
	월	화	수	목	금	토	일	시간	거리	최대 운행 거리	
현재	36	36	36	36	37	39	39	0	0	≤3 500	-
검수 완화시	32	32	32	32	34	37	37	4	10	4485	1.20
검수 고려시	32	32	32	32	34	37	37	0	0	≤3 500	1.98

표 5.1 차량반복계획 실험결과

검수조건 완화시 실험 결과 37개의 운용을 구하였다. 이는 4.2절의 알고리즘의 Step 1에 해당하며 본 연구에서 풀고자하는 두 가지 검수 조건을 가진 차량 반복계획 문제 최적해의 하한값(Lower Bound)이 된다. 하지만 이 운용들은 시간과 거리 검수조건을 각 4번, 10번 위반한다.

본 연구에서 제안한 4.2절의 휴리스틱 알고리즘을 적용한 결과 거리와 시간 검수를 만족시키는 37개의 운용을 구할 수 있었다. 차량소요량은 검수조건 완화시 차량 반복계획의 차량 소요량과 같으므로 최적정수해임을 알 수 있다. 또한 현재 철도공사의 차량 반복계획보다는 주중에는 차량소요량을 평균 4대 주말에는 2대 줄이는 차량 반복계획이다. 계산시간은 검수조건에 유무에 상관없이 매우 짧다. 이는 휴리스틱 알고리즘의 계산 복잡도가 n=열차수, $O(n^2)$으로 다항 시간의 계산 복잡성을 가지고 있기 때문이다.

5.2 열차 추가 문제 실험결과

이번 실험은 현재보다 열차가 추가 되었을 때 본 연구에서 제시한 해법으로 해를 구해보고자 한다. 현재 KTX 열차 스케줄에 가장 승객수가 많은 시간대인 12시 ~ 22시 사이에서 임의로 열차를 전 요일 추가하였다. 또한, 열차 추가 시 기존의 열차 노선의 비율을 고려하여 10개의 Data Set을 생성한다. 구체적으로 Data Set $j+1$ 은 Data Set j에 표 5.2에 해당하는 열차를 추가하여 생성한다.

Data Set 1	서울-부산 1편, 부산-서울 1편 증편
Data Set 2	서울-부산 1편, 부산-서울 1편 증편
Data Set 3	서울-부산 1편, 부산-서울 1편 증편
Data Set 4	용산-목포 1편, 목포-용산 1편 증편
Data Set 5	서울-동대구 1편, 동대구-서울 1편 증편
Data Set 6	서울-부산 1편, 부산-서울 1편 증편
Data Set 7	용산-목포 1편, 목포-용산 1편 증편
Data Set 8	서울-부산 1편, 부산-서울 1편 증편
Data Set 9	서울-동대구 1편, 동대구-서울 1편 증편
Data Set 10	용산-광주 1편, 광주-용산 1편 증편

표 5.2 Data Sets

표 5.3에 정리 된 것처럼 실험결과 10개의 Data Set 모두에서 최적해를 찾을 수 있었다. 이를 통해 본 연구에서 제안한 휴리스틱 기법이 많은 대안해(Alternative Route)를 가진 경우에 효과적임을 알 수 있다.

	차량소요량							검수조건위반			최적해 여부
	월	화	수	목	금	토	일	시간	거리	최대운행거리	
Data Set1	34	34	34	34	36	39	39	0	0	≤3500	Y
Data Set2	36	36	36	36	38	40	40	0	0	≤3500	Y
Data Set3	36	36	36	36	38	41	41	0	0	≤3500	Y
Data Set4	38	38	38	38	40	43	43	0	0	≤3500	Y
Data Set5	39	39	39	39	41	45	45	0	0	≤3500	Y
Data Set6	40	40	40	40	42	45	45	0	0	≤3500	Y
Data Set7	41	41	41	41	43	46	46	0	0	≤3500	Y
Data Set8	42	42	42	42	44	46	46	0	0	≤3500	Y
Data Set9	42	42	42	42	44	46	46	0	0	≤3500	Y
Data Set10	43	43	43	43	45	47	47	0	0	≤3500	Y

표 5.3 열차 스케줄 추가 실험결과

6. 결론 및 추후연구방향

본 연구에서는 거리와 시간 두 가지의 검수조건을 가지고 있을 때 철도 차량 운용계획 문제를 푸는 해법을 제안하였다. 본 연구에서 수행한 내용은 다음과 같다.

첫째, 차량 운용계획 문제를 운행주기 반복성 및 장기적인 운행 통일성에 따라서 차량 경로계획, 차량 반복계획, 차량 순환계획으로 나누고 그 특징을 분석하였다.

둘째, 거리와 시간 두 가지 검수조건을 가지는 차량 운용계획 문제가 시간 복잡도 관점에서 NP-hard에 속한다는 것을 증명하였다.

셋째, 검수조건을 완화하는 경우 최소비용 흐름 모형에 기반한 수리 모형을 제시하였으며 검수조건을 고려하는 경우에 합리적인 휴리스틱 해법을 제시하였다.

넷째, 본 연구에서 제안한 휴리스틱 알고리즘을 실제 차량 운용계획 문제에 적용해 최적해를 찾았으며 열차 스케줄을 추가하여 실험하였다.

추후 연구과제 : 기존의 Set partitioning 모형과의 비교

참 고 문 헌

[1] C. Barnhart, N.L. Boland, □□Flight String Models for Aircraft Fleeting and Routing,□□ *Alogrithica*, Vol.20(1998), pp.151-174.

[2] C. Barnhart, 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.

[3] L. Clarke, E.L. Johnson, G.L. Nemhauser, □□The Aircraft Rotation Problem,□□ *Ann. Oper. Res. Math. Ind. Syst*, Vol.69(1997), pp.33-46.

[4] M.A. Forbes, J.N. Holt and A.M. Watts, □□ Exact Solution of Locomotive Scheduling Problems,□□ *J.Opl Res. Soc*, Vol.42(1997), pp.825-831.

[5] G.Maroti and L.Kroon,□□Maintenance routing for train for train units:the interchange model,□□ *Computers and Operations Research*

[6] R. Gopalan, K.T. Talluri, □□The Aircraft Maintenance Routing problem,□□ *Operations research,* Vol.46(1992).

[7] N.M. Kabbani and B.W. Patty, □□Aircraft Routing at American Airlines,□□*Proceedings of the Thirty-Second Annual Symposium of AGIFORS,* Budapest, Hungary(1992)

[8] Schrijver, □□Minimum circulation of railway stock,□□ *technical notes*

[9] T.Talluri, □□The four-day aircraft maintenance routing problem,□□*Transportation Science*, Vol32(1998),

[10] P.H. Vance, C. Barnhart, E.L. Johnson and G.L. Nemhauser, □□Airline Crew Scheduling: A New Formulation and Decomoposition Algorithm □□ *Operations research,* Vol.45(1997).

[11] P. Zhao, P, N. Tomii, N. Fukumura, T. Sakaguchi, □□An algorithm for train-set scheduling based on probabilistic local search□□ *Computers in railways*, *2002,* pp.817-826