■ 論 文 ■

다시간대에 대한 버스 스케줄링 모형 개발

A Development of Multi-Period Bus Scheduling Model

고 종 섭
((주)천일기술단 교통연구실 이사)

고 승 영
(명지대학교 교통공학과 부교수)

목 차

요 약

현재까지 버스 스케줄은 일반적으로 순수배정문제 혹은 네트워크 유량문제의 형태로 하루의 모든 운행을 한꺼번에 처리하는 일괄처리 방식으로 작성되고 있다. 이 방법에 의할 경우 버스 스케줄링을 위한 계산량은 운행수(n)의 3승(n^3)에 비례하기 때문에 하루 동안의 운행회수가 많으면 문제의 크기가 커져 버스 스케줄을 구하는데 어려움이 발생한다. 만약 일정시간 동안 버스가 동일한 형태로 배차된다면 선입선출법(FIFO)과 같은 간단한 방법으로 해당 시간대에 대한 확장운행을 작성할 수 있기 때문에 스케줄링의 기본단위의 개수를 줄일 수 있고, 문제의 크기가 작아짐으로 인해 버스 스케줄을 쉽게 구할 수 있다.

본 논문에서는 시간대별로 동일한 배차형태를 갖는 버스운행형태에서 버스 스케줄링 문제의 크기를 줄이기 위해 하루 중에 운행시격 및 운행소요시간이 동일한 시간대를 분할하고, 분할된 시간대에 대한 시간대별 확장운행을 선입선출법으로 작성한 다음, 시간대별 확장운행을 기본단위로 하는 새로운 네트워크에서 하루 전체에 대한 버스 스케줄을 작성하는 시간분할에 의한 다시간대 버스 스케줄링 방법을 제시하였다.

Ⅰ. 서론

1. 연구의 목적 및 범위

우리나라에는 시내버스, 시외버스, 고속버스, 지방벽지노선버스 등 다양한 형태의 버스가 운행되고 있다. 이중에서 운행회수가 많은 시내버스, 시외버스 및 고속버스의 대부분은 시간대별로 운행시격을 달리하여 운행하는 단일노선 운행방식이며, 운행회수가 적은 시외버스, 고속버스, 시외벽지노선 등은 한 대의 버스가 몇 개의 노선을 교차 운행하는 복수노선 운행방식이다. 단일차고지 단일노선 버스 스케줄링은 다른 노선으로의 공차운행을 감안하지 않기 때문에 버스 스케줄링 문제 중에 가장 단순한 형태로서 선입선출법으로도 버스 스케줄을 작성할 수 있다. 그러나 복수노선 운행방식의 경우에는 공차운행에 의한 다른 노선 운행을 함께 고려해야 하기 때문에 단일노선 운행방식에 비하여 고려해야할 제약조건과 대안의 수가 많아 최적의 버스 스케줄을 수 작업으로 작성하는 것은 거이 불가능하다고 할 수 있다. 그리고 운행회수가 많은 경우에는 하나의 버스 스케줄 대안을 작성하는데 많은 인력과 시간이 소요되어 버스운행여건변동에 대처하거나, 운행개선 효과 등을 판단하기 위한 새로운 버스운행 대안을 제시하는데 많은 어려움을 겪고 있는 실정이다.

효율적인 버스 스케줄은 최소의 차량운행비용으로 주어진 노선의 승객들을 수송할 수 있도록 할뿐만 아니라 승객의 정류장대기시간을 단축시키는 부수적인 효과를 제공하기 때문에 선진외국에서는 대중교통의 차량 및 승무원 스케줄링을 주제로 한 국제워크숍을 개최[1]하는 등 효율적인 스케줄작성을 위한 기술개발에 많은 노력을 기울이고 있으며, 1970년대 중반부터 외국의 많은 대형버스회사들은 컴퓨터를 이용한 스케줄링 시스템을 개발사용 중에 있다.

반면 국내의 대부분 버스회사들은 규모의 영세성 등으로 인해 경험에 의한 수 작업으로 버스 스케줄을 작성하거나 혹은 버스운행에 대한 구체적인 계획 없이 운전자로 하여금 하루에 주어진 노선을 몇 회씩 운행토록 하고 있는 실정이다. 그러나 버스회사의 운영개선을 위해서는 버스 및 운전자의 운영 효율을 높여야하며, 선진화된 스케줄링 시스템 개발이 필요한 시점이라고 판단되며, 국내의 버스운행 환경에 적합하고 간편한 버스 스케줄링 방법이 필요한 상황이다.

기존의 버스 스케줄링 방식은 계산량이 운행 회수의 3승(O(n^3))에 비례하는 교통배정문제(transportation assignment problem) 혹은 네트워크유량모형(network flow model) 알고리즘 등으로 하루 동안 운행할 버스 스케줄을 한번에 구하는 일괄처리방식이 주로 이용되고 있다. 그런데 이 방법에 의할 경우 하루의 운행회수가 많으면 버스 스케줄을 구하기 위한 계산량이 많아진다. 분할 처리방식으로 버스 스케줄을 작성하여 스케줄링의 기본단위의 개수를 줄일 수 있다면, 버스 스케줄링을 위한 계산량을 대폭적으로 줄일 수 있을 것이다.

본 논문은 하루의 버스운행을 운행특성이 동일한 몇 개의 시간대로 분할하고, 각 시간대별 확장운행을 선입선출법으로 작성한 다음, 이들을 교점으로 하는 네트워크에서 하루 전체에 대한 버스 스케줄을 작성하는 시간분할에 의한 다시간대버스 스케줄링 방법을 제시하는 것이 목적이다. 단, 버스는 단일차고지에서 동일한 등급의 버스가 배차되는 경우로 한정한다.

본 논문의 목적을 달성하기 위해 제Ⅰ장에는 본 논문의 목적과 관련 연구에 대한 현황을 요약하고, 제Ⅱ장에는 기존연구에 의한 버스 스케줄링 문제의 기본모형 및 관련 용어들을 정의하며, 제Ⅲ장에는 본 논문의 주목적인 시간분할에 의한 다시간대 스케줄링 방법의 배경 및 과정을 설명하고, 제Ⅳ장에서는 시간분할에 의한 다시간대 버스 스케줄링 방법과 효과를 예를 들어 설명하였다. 그리고 마지막 제Ⅴ장에 본 논문의 결과를 종합 정리하고 앞으로 수행되어야할 연구의 내용들을 요약하였다.

2. 관련 연구의 현황

Bodin, Gloden, Assad와 Ball(1983)은 네트워크유량모형으로 단일차고지 버스 스케줄링 문제를 설명하였고, 이를 확장한 네트워크에서 계산량이 $O((n+1)^3)$인 알고리즘으로 버스 스케줄을 구하였다. 그리고 Gavish와 Shilfer(1978)는 계산량이 $O((n+1)^3)$인 수송문제(Transportation problem) 알고리즘을 이용하여

1) Chicago(1975), Leeds(1980), Montréal(1983), Hamburg(1987), Montréal(1990), Lisbon(1993)

단일차고지 버스 스케줄링 문제의 해를 구하였다. 배정문제(Assignment Problem)는 수송문제의 특수 형태로서 Orloff(1976)는 계산량이 $O(n^3)$인 순수배정문제 알고리즘으로 단일차고지 버스 스케줄링 문제의 해를 구하였다(Daduna, 1995).

버스 스케줄링 문제는 보통 수 백 개에서 수 천 개의 운행들을 기본단위로 하기 때문에 수리적 모형에 의해 해를 구하기 위해서는 문제의 크기를 감안하여야 한다. Kirkman 과 Prinz(1968)는 스케줄링 문제의 크기를 줄이기 위해 'Hungarian Matrix Reduction' 기법을 사용하였고, 배정문제 알고리즘에 의해 해를 구하는 방법을 제안하였다. 크기가 큰 원래 문제를 크기가 작은 몇 개의 소규모 문제로 분할하고, 각각의 소규모 문제에 대한 해를 독립적으로 구한 후, 이들을 하나로 모은 버스 스케줄을 작성하였다. 이와 같은 방법은 버스 스케줄을 구하기 위한 계산에서는 매력적일 수 있으나, 최적의 버스 스케줄을 보장하는 것은 아니다. Hoffstadt(1980)는 버스 스케줄링 문제를 배정문제로 모형 화하고, Hungarian Method를 사용하여 해를 구하는 방법을 제시하고 있다. 이 프로그램 역시 규모가 큰 문제를 작은 규모로 줄이기 위한 절차와 복수차고지를 다루기 위한 발견적방법들을 통합한 것이다(EL-AZM, 1985).

이상에서 보는 것처럼 단일차고지 버스 스케줄링문제는 결정적알고리즘(deterministic algorithm)의 다항식(polynomial) 계산량을 갖는 배정문제 혹은 네트워크유량문제의 알고리즘을 이용하여 최적해를 구할 수 있다. 그런데 버스운행시간표에 주어진 운행수가 많으면 많을수록 해를 구하는 계산량은 다항식으로 증가하여 최적해를 구하는데 많은 어려움이 있을 뿐만 아니라 많은 시간이 소요된다. 선행연구들은 문제의 크기를 줄이기 위하여 원래의 문제를 몇 개로 분할하고 배정문제알고리즘의 하나인 Hungarian Method를 적용하여 분할된 각 소그룹의 버스 스케줄을 구하였다. 이런 방법은 운행수가 N인 버스 스케줄링 문제의 해를 일괄처리방식으로 구할 경우 계산량은 O(N^3)이나, 주어진 문제를 J개의 소그룹으로 분할하여 각 소그룹별로 해를 구한다면(N = $\sum_{j=1}^{J} n_j$ 라고 가정) j소그룹에 대한 계산량은 O(n_j^3)임으로, 계산량을 O(N^3)에서 $\sum O(n_j^3)$로 줄이기 위한 것이다.

그러나 본 논문은 하루 운행을 버스의 운행특성이 동일한 시간대로 분할하고 해당하는 시간대에는 동일한 연결운행 원칙을 적용함으로써 N개의 운행을 n(〈N)개의 확장운행으로 변환시킴으로써, 계산량이 O(N^3)인 문제를 O(n^3)(〈O(N^3))로 변환시켜 해를 구하는 것이다.

Ⅱ. 버스 스케줄링 문제의 기본모형

1. 용어 정의 및 네트워크 표현

운행(trip)은 하나의 버스가 동일 노선의 한 지점에서 다른 한 지점까지 이동하는 것을 말한다. 일반적으로 버스운행시간표에 주어진 기점에서 종점까지 혹은 종점에서 바로 회차하는 경우에는 기점에서 기점까지를 버스의 1회 운행이라고 할 수 있으며, 각 운행에는 i(또는 j)로 표현되는 번호가 부여된다. 그리고 매회 운행(i)은 t_i^1, p_i^1, t_i^2, p_i^2과 같은 속성치를 갖는다.

여기서,

t_i^1 : i 운행의 출발시간

p_i^1 : i 운행의 출발지점

t_i^2 : i 운행의 종료시간

p_i^2 : i 운행의 종료지점

계획기간의 모든 운행들을 원소로 하는 운행집합(I ={ 1, 2, ⋯, n })을 설정할 수 있고, 운행 i의 종료지점(p_i^2)에서 운행 j의 출발지점(p_j^1)까지의 이동시간을 $d\,(p_i^2,\ p_j^1)$라고 하면, 운행집합(I)에 속한 운행 중에서 $t_i^2 + d\,(p_i^2,\ p_j^1) \le t_j^1$ 를 만족하는 운행 i 와 j는 서로 "연결(혹은 순차)운행 할 수 있다(compatible)"라고 한다(Daduna, 1995). 그리고 승객을 수송하지 않는 이동을 공차운행(dead-heading trip)이라 하고, 스케줄링 과정에서 필요에 따라 공차운행을 삽입할 수 있다. 연결운행 조건을 만족시키는 2개 이상의 운행들을 연결한 확장운행을 본 논문에서 새롭게 정의한다.

버스 스케줄링 문제는 그래프이론(Graph Theory)

으로 표현할 수 있으며, 네트워크 알고리듬을 이용하여 해를 구하고 있다. 운행(i)을 교점(node)으로 하고, 연결 운행할 수 있는 교점(운행)들을 서로 연결한 호(arc)로 구성되는 네트워크로 표현된다. 이 네트워크를 G=(V, A)라고 하면, 여기서, V는 교점들의 집합이고, A는 연결운행 할 수 있는 교점 i와 j를 연결한 호들의 집합으로서 다음과 같이 정의한다.

$$V = I \cup \{n+1\}$$

여기서, n+1은 차고지 번호

$$A = \{(i,\ j):\ t_i^2 + d(p_i^2, p_j^1) \le t_j^1\} \cup \{(n+1,\ i):\ i \in I\} \cup \{(i,\ n+1):\ i \in I\}$$

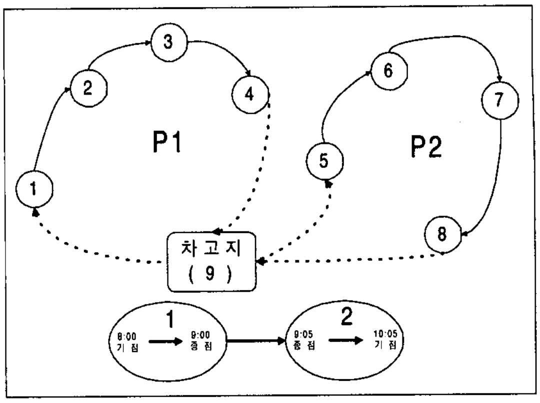
그리고 호 A는 버스운행시간표에 주어진 운행들을 연결하는 호 A_1과 버스의 차고지 유・출입을 나타내는 호 A_2로 구분할 수 있으며, 다음과 같이 정의할 수 있다.

$$A_1 = \{(i,\ j);\ t_i^2 + d(p_i^2, p_j^1) \le t_j^1;\ i,\ j \in I\}$$

$$A_2 = \{(n+1,\ i);\ i \in I\} \cup \{(i,\ n+1);\ i \in I\}$$

〈그림 1〉의 네트워크에서 각각의 교점(1번~8번)은 운행을 뜻하며, 출발시간, 출발지점, 종료시간, 종료지점의 특성치를 갖는다. 그리고 1번 운행의 도착시간에 1번 운행의 종료지점에서 2번 운행의 출발지점까지의 이동시간을 더한 값($t_1^2 + d(p_1^2,\ p_2^1)$)이 2번 운행의 출발시간(t_2^1) 보다 작거나 같다면, 1번 운행과 2번 운행은 '연결운행 할 수 있다'라고 한다. 여기서 I={ 1, 2, ⋯, 8} 이고, 교점 V={ 1, 2, 3, ⋯, 8, 9 }이며, 연결운행 할 수 있는 교점간을 연결한 호들의 집합을 A라 한다. 그림에서 실선으로 표시된 호는 수입운행을 위한 연결운행들을 나타내며, 점선으로 표시된 호들은 차고지에서의 버스출고 혹은 차고지로의 버스입고를 나타내는 공차운행을 뜻한다.

그리고 순환로(circuit)란 주어진 네트워크의 교점들을 정확히 한번씩 방문한 후 다시 출발지로 돌아오는 경로를 말하며, 그림은 2개의 순환로(P_1, P_2)로



〈그림 1〉 버스 스케줄링 문제의 네트워크 표현

구성된 네트워크 G이다. 여기에서 기점을 제외한 모든 교점이 단 하나의 순환로에만 포함된 순환로들을 '중복되지 않는 순환로(disjointed circuit)'라 칭한다. 순환로 P_1과 P_2는 각각 한 대의 버스가 운행할 하루 동안의 운행량이다. 버스 스케줄링 문제는 주어진 버스운행시간표의 모든 운행을 교점으로 하고, 연결운행이 가능한 교점들을 연결한 호로 표현되는 네트워크 G에서 최소의 비용(최단경로)으로 모든 교점을 정확히 한번씩 방문하는 중복되지 않는 여러 개의 순환로를 찾는 문제라고 정의할 수 있다.

2. 버스 스케줄링 문제의 수리적 표현

버스회사는 시간대별 승객수요 및 교통여건을 고려한 버스운행시간표를 작성하고, 버스운행시간표의 모든 운행을 제한된 수의 버스에 정확히 한번씩 배정하는 버스 스케줄을 다음과 같은 제약조건과 목적에 만족하도록 작성한다(Daduna, 1995).

- 모든 운행에 정확히 한 대씩의 버스가 배정되어야 한다.
- 회사별 혹은 기타 기술적인 제약 조건들에 의해 결정되는 연결운행 준비시간 또는 공차운행 등과 관련한 규정에 만족해야 하며, 연결운행 할 수 있어야 한다.
- 버스운행과 관련한 총 비용이 최소화되어야 한다.

버스 스케줄링의 목적은 주어진 조건을 만족시키면서 불필요한 비수입 운행비용의 발생을 최대한 억제하여 버스운행과 관련한 총 운행비를 최소화하는 것이다. 그러나 수입운행에 따른 운행비용은 주어진 버

스운행시간표에 의해 일정하게 발생되는 상수값으로 버스 스케줄링 문제에서는 고려되지 않고, 공차운행시간(혹은 거리), 연결운행시간 등과 관련하여 발생하는 비수입 운행비용의 발생을 최소화하는 것이다.

버스 스케줄링문제는 다음과 같은 교통배정문제로 모형화 할 수 있다(EL-AZM,1985).

$$\text{Minimize } Z=\sum_{i\in V}\sum_{j\in V}c_{ij}x_{ij} \quad (1)$$

$$\text{Subject to } \sum_{j\in V}x_{ij}=1,\ i\in I \quad (2)$$

$$\sum_{i\in V}x_{ij}=1,\ j\in I \quad (3)$$

$$\sum_{(i,j)\in A_2}x_{ij}\le k \quad (4)$$

$$x_{ij}\in\{0,1\}\quad i,j\in V \quad (5)$$

여기에서 식(1)은 V에 속한 교점들을 연결하는데 필요한 총 운행비용(공차운행시간, 연결운행시간 등)을 최소화하는 것이다. 이때 계수(c_{ij})는 식(6)과 같이 정의된다. 그리고 식(2)는 i 운행을 종료한 버스는 후속의 j를 운행하는 것을 의미하고, 식(3)은 j를 운행하기 전에 i를 운행했어야 함을 뜻한다. 식(4)는 스케줄에 투입할 수 있는 버스대수(k)를 제한하는 것으로, 하루 동안에 버스의 차고지 유·출입 회수를 제한하는 상한치이다. 그리고 식(5)에서 결정변수(x_{ij})는 i와 j가 연결운행 할 수 있으면, $x_{ij}=1$이 되고, 그렇지 않으면 $x_{ij}=0$이다.

$$C_{ij}=\begin{cases}\alpha_{ij} & i,\ j\in I,\ t_i^2+dp_i^2,p_j^1)\le t_j^1\\ \beta_{n+1,j} & j\in I\\ \gamma_{i,n+1}, & I\in I\\ \infty & i,j\in I,\ i=j,\ t_i^2+d(p_i^2,\ p_j^1)>t_j^1\end{cases} \quad (6)$$

여기서 α_{ij}는 i, j∈I인 i와 j간의 연결운행 비용이고, $\beta_{n+1,j}$는 버스가 차고지에서 출고되어 j를 운행하는 비용이며, $\gamma_{i,n+1}$은 i를 운행한 버스가 차고지로 입고하는데 소요되는 비용이다. 그리고 운행 i와 j가 연결운행 할 수 없거나, i와 j가 동일한 운행일 때는 최적해(x_{ij}^*)가 1이 되지 않도록 하기 위해 c_{ij}에 임의의 큰 값(∞)을 부여한다.

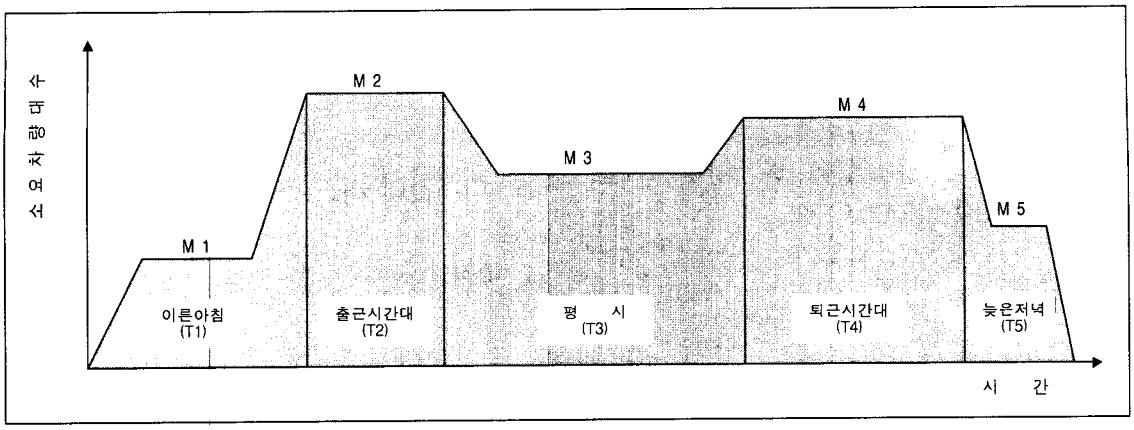
III. 다시간대 버스 스케줄링 방법

1. 배경

각 시간대별로 배차시격과 운행소요시간이 동일한 경우에는 〈그림 2〉에서 보는 것처럼 이른 아침 시간대(T_1)에는 M_1대, 출근 시간대(T_2)에는 M_2, 평시(T_3)에는 M_3, 퇴근 시간대(T_4)에는 M_4, 저녁 늦은 시간대(T_5)에는 M_5대의 버스가 운행된다. 이처럼 각 시간대별로 동일한 버스대수가 운행할 때는 차고지에서 버스를 추가로 출고시키거나 차고지로 입고시킬 필요가 없다. 그리고 차고지로 버스를 입·출고시키지 않는 시간에 운행하는 버스들은 모두가 동일한 규칙에 따라 후속 운행을 연결시킬 수 있다. 즉 한 대의 버스에 대한 최적의 연결운행방식이 다른 버스에 동일하게 적용될 수 있으며, 단일노선 운행의 경우 제약조건을 만족시키는 선입선출법에 의한 연결운행으로 최적의 확장운행들을 구할 수 있다.

그런데 〈그림 2〉의 T_1시간대에서 T_2시간대로 전이될 때는 현재 운행중인 버스대수(M_1) 보다 M_2-M_1대의 버스가 추가로 필요($M_2>M_1$인 경우)하고, T_2시간대에서 T_3시간대로 전이될 때는 운행중인 버스(M_2대) 중에서 M_2-M_3대의 버스를 차고지로 입고($M_2>M_3$인 경우)하여야 한다. 시간대별로 소요 버스대수에 차이가 날 때에는 '새로이 출고하는 버스를 어떤 운행에 배정할 것인가? 혹은 운행 중에 있는 버스들 중에서 어떤 버스를 차고지로 입고시키는 것이 효율적인가'를 고려해야 하기 때문에 버스들 간에 동일한 연결운행 원칙이 적용될 수는 없다.

앞의 단일차고지 버스 스케줄링 문제에서 목적함수의 비용계수(c_{ij})에서 연결운행 할 수 있는 운행 i 와 j 간의 연결운행비용을 α_{ij}로 정의하였다. 그런데 α_{ij}는 식(2)와 같이 2가지로 세분할 수 있다. 하나는 i 운행의 종료지점(p_i^2)과 j 운행의 시작점(p_j^1)이 동일($p_i^2=p_j^1$)한 경우로 공차이동 없이 운행 i와 j를 연결



〈그림 2〉 시간대별 필요 버스대수

할 수 있는 경우이다. 그리고 다른 하나는 p_i^2와 p_j^1의 위치가 서로 달라($p_i^2 \neq p_j^1$) 버스의 공차이동을 통하여 i와 j운행을 연결할 수 있는 경우이다.

$$\alpha_{ij} = \begin{cases} (t_j^1 - t_i^2)\alpha_w, \text{ 단, } p_i^2 = p_j^1 & (7) \\ (t_j^1 - t_i^2 - d(p_i^2, p_j^1))\alpha_w + d(p_i^2, p_j^1)\alpha_o \\ \text{단, } p_i^2 \neq p_j^1 \end{cases}$$

여기서, α_{ij} : 연결운행할 수 있는 운행 i 와 j의 연결운행비용

α_w : 버스의 단위시간당 대기비용

α_o : 버스의 단위시간당 운행비용

식(7)에서 운행 i가 주어졌을 때 i와 연결운행 할 수 있는 운행 j의 출발지점이 동일하지 않을 경우 주어진 운행 i와 최소비용으로 연결할 수 있는 운행은 식(8)을 만족시킨다.

$$\alpha^*_{ij} = \min\{\alpha_{ij}, : t_i^2 + d(p_i^2, p_i^1) \le t_j^1 \text{ 인 운행 } j\} \qquad (8)$$

주어진 노선의 운행시격을 h라고 하면, h동안에는 한 대의 버스가 출발하기 때문에 임의의 시간에 도착한 버스는 늦어도 h이내에 출발할 수 있다. 그렇기 때문에 최소비용에 의한 연결운행시간은 운행시격(h)을 초과하지는 않는다. 즉, 운행 i와 j가 최소비용에 의한 연결운행일 경우에는 $t_j^1 - t_i^2 \le h$의 관계를 만족시킨다. 그리고 차량운행비에는 시간비용을 포함하고 있기 때문에 $\alpha_o \ge \alpha_w$의 관계 역시 성립한다. 만약 $p_i^2 \neq p_j^1$이고, $h \le d(p_i^2, p_j^2)$이면, $\alpha_w \le \alpha_o$임으로 $h\,\alpha_w \le d(p_i^2, p_j^2)\,\alpha_o$ 역시 성립한다. 결과적으로 운행시격 h가 다른 지점으로 이동하는 공차운행 시간보다 작다면, 동일지점에서 연결운행하는 것이 다른 지점으로 이동하여 연결운행하는 것보다 경제적이기 때문에 동일 지점에서 선입선출법에 의한 연결운행으로 최적의 시간대별 확장운행을 구할 수 있다. 버스들은 정해진 시간마다 1대씩 배차되기 때문에 선입선출법에 의해 단 하나의 시간대별 확장운행 집합이 작성된다.

이들 시간대별 확장운행들은 전체네트워크를 시간대별로 중복없이 분할한 부분네트워크에서의 최단경로들이다. 이들 부분네트워크의 최단경로들 각각을 교점으로 하는 전체네트워크에서 최단경로를 구하면, 하루 전체에 대한 최적의 버스 스케줄을 구할 수 있다.

이와 반대로, 운행시격(h)이 다른 지점으로 공차이동하는 시간보다 긴 경우에는 운행시격이 길기 때문에 하루 동안의 버스운행 회수가 많지 않은 경우에 해당된다. 이는 버스 스케줄을 작성할 때 한번에 고려해야 할 운행수가 많지 않는 경우로 스케줄링 문제의 크기 역시 그렇게 크지 않는 것이 일반적이라고 할 수 있다. 스케줄링 문제가 크지 않은 경우에는 일반적인 스케줄링 방법에 의해서 최적의 버스 스케줄을 구하는데 큰 어려움이 없기 때문에 본 논문의 시간분할에 의한 다시간대 스케줄링 방법을 적용할 필요는 없다.

2. 버스 스케줄링 과정

1) 단일노선 버스 스케줄링

단일차고지 단일노선에서 버스가 공차운행 할 수 있는 경우는 차고지로의 입·출고와 반대 기점으로 이동하는 경우라고 할 수 있다. 그런데 단일노선 운행방식은 운행시격이 짧고, 운행회수가 많은 경우에 주로 적용되기 때문에 운행시격이 반대 기점으로의 공차이동시간 보다 짧은 경우가 대부분이다. 이런 경우에는 반대 기점으로 공차이동하는 비용이 동일 지점에서 후속 운행을 연결할 때의 비용보다 크기 때문에 반대 기점에서 출발하는 운행과의 연결은 최적의 버스 스케줄 대안이 될 수 없다. 그리고 차고지로의 입·출고는 버스 혹은 운전자의 교대 및 장시간 휴무 등이 필요할 때 발생되며, 버스 운행과 관련된 비수입 운행비용을 줄이는 데는 도움이 되지 않는다.

결국 단일차고지 단일노선 운행방식에서의 버스 스케줄링 문제는 각 운행의 기점에서의 연결운행시간을 최소화시키면서 차고지로의 입·출고 회수 역시 최소화시키는 것이다. 다음과 같은 방법으로 각각의 버스에 대한 하루 동안의 운행들을 선입선출법으로 연결하여 최적의 버스 스케줄을 구할 수 있다.

- 동일한 버스대수가 운행되는 시간대에는 조건(연결운행 준비시간)을 만족시키는 운행들 중에서 최초로 출발하는 운행을 연결한다.
- 운행대수에 차이가 나는 시간대에는 주어진 운행 i에 대하여,
  - $t_j^1 - t_i^2 \geq 0$, $t_j^1 - t_{i+1}^2 \geq 0$ 이면 i를 운행한 버스는 차고지로 입고한다.
  - $t_j^1 - t_i^2 \geq 0$, $t_{j+1}^1 - t_{i+1}^2 < 0$이면 j는 i와 연결운행하고, j+1 운행은 차고지에서 출고한 새로운 버스에 배정한다.

2) 복수노선 버스 스케줄링

하나의 버스가 여러 노선을 교차 운행하는 경우는 단일노선의 경우에서 보다 노선간 이동을 위한 공차운행이 추가된다. 우선 버스운행시간표의 모든 운행들을 각 노선의 기점별로 운행특성이 동일한 시간대로 분할한다. 그리고 시간대별로 분할된 각 기점별 운행들을 출발시간 순서에 따라 정리하고, 노선별 운행시격이 반대 기점으로의 공차운행시간 보다 짧으면, 노선별 기점 중에서 차고지에서 가장 멀리 떨어진 기점에서 출발하는 운행들부터 먼저 선입선출법으로 연결시킨 확장운행을 작성한다. 그 다음에 또 다른 기점(그 다음 원거리에 위치한 기점)에서 출발하는 운행들을 앞에서와 동일한 방법으로 연결시킨다. 2개 이상의 운행이 연결된 확장운행은 최초 운행(i)의 출발지와 출발시간, 최종 운행(j)의 도착지와 도착시간을 속성으로 한다.

이렇게 작성된 확장운행들과 차고지를 교점으로 하는 하루 전체에 대한 새로운 네트워크를 작성하고, 교통배정문제 알고리즘에 의해 버스 스케줄을 구한다. 이 과정에 대한 절차는 〈그림 3〉과 같다.

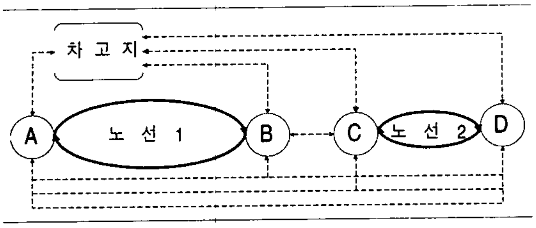
단계 1)	각 노선별(r)로 주어진 버스운행시간표의 1회 운행을 운행시격 및 운행소요시간이 동일한 시간대별로 분할한 노선(r)별 지점(p)별 시간대(t)별 부분운행집합 I_{rpt}(r=1, 2, ⋯, R, p = 1, 2, ⋯, R, t = 1, 2, ⋯, T)를 작성한다.
단계 2)	부분운행집합 I_{rpt}에서 각 노선의 지점별(p)로 연결운행이 가능한 운행들을 선입선출법으로 연결한 t시간대 r노선에 대한 확장운행 (ET_{rt})들을 구하고, 이들을 버스 스케줄의 기본단위로 한다.
단계 3)	확장운행(ET_{rt})들과 차고지(n+1)를 교점으로 하는 새로운 전체 네트워크 G′=(V′, A′)를 작성한다. 여기서 교점 V′={ ET_{rt}, n+1}이고, 호 A′는 ET_{rt}중에서 연결이 가능한 확장운행들을 연결한 호 및 차고지(n+1)를 연결하는 호들의 집합이다.
단계 4)	새로운 전 시간대 네트워크 G′에 교통배정문제 알고리즘을 적용하여 최적의 버스 스케줄을 구하고 종료한다.

〈그림 3〉 복수노선 버스 스케줄링 과정

Ⅳ. 예제 및 모형의 평가

1. 예제

〈그림 4〉와 같이 하나의 차고지에서 2개 노선(노선 1, 2)에 버스를 배차하는 경우를 가정한다. 1번 노선은 A와 B를 기점으로 하고, 2번 노선은 C와 D를 기점으로 하는 노선으로 구성된다. 1번 노선을 운행한 버스는 2번 노선의 C 혹은 D지점으로 공차



〈그림 4〉 예제 노선의 구성

〈표 1〉 각 기점간 공차운행시간 (단위:분)

출발지 \ 종착지	차고지 (G)	A	B	C	D
차고지(G)	0	20	35	45	70
A	20	0	35	45	70
B	35	35	0	12	35
C	45	45	12	0	25
D	70	70	35	25	0

이동한 후 2번 노선을 운행할 수 있으며, 2번 노선을 운행한 버스 역시 1번 노선을 운행할 수 있다. 그림에서 점선은 공차운행을 뜻하며, 실선은 수입운행을 나타낸다. 그리고 차고지와 각 노선의 기·종점간 이동에 필요한 공차운행시간은 〈표 1〉과 같고, 각 노선의 버스운행시간표는 첨부와 같다고 가정한다.

〈그림 3〉의 단계 1)에 의한 노선별 시간대별 부분 운행집합(I_{rpt})은 첨부의 버스 운행시간표와 같이 작성될 수 있다. 1번 노선의 경우 이른 아침시간(6:00~7:00)대에는 101번부터 106번까지 6개를 운행하고 출근 시간대에는 26개 운행을, 평시에는 64개를 운행하는 경우이다.

그리고 단계 2)에 의해 각 시간대별로 차고지에서 원거리에 위치한 노선 1의 B지점과 노선 2의 D지점에서 연결 가능한 운행들을 선입선출법으로 연결하면, 노선 1의 출근 시간대에는 107번과 126번 운행을 연결할 수 있고, 108번과 127번 역시 연결운행이 가능하다. 이와 같은 방법에 의한 결과는 〈표 2〉와 같다. B지점에서 출발하는 운행들을 연결하면, 노선 1에서는 원래 96개의 운행을 고려해야 하나 61개의 확장운행(ET_{rt})으로 변환되었다. 그리고 노선 2 역시 120개의 운행에서 71개의 확장운행으로 변환되었다. 앞에서와 동일한 방법으로 A와 C지점에서 한번 더 연결하면 〈표 3〉과 같은 결과를 얻을 수 있다. 결국 노선 1의 확장운행은 26개이고, 노선 2는 22개이다. 이들에 번호를 부여하면 101번에서 126번까지의 확장운행이 노선 1에 속하며, 127번부터 148번까지는 노선 2를 운행하는 확장운행들이다.

그리고 확장운행을 구성하는 맨 처음 운행의 출발시간과 출발지, 맨 마지막 운행의 도착시간과 도착지를 확장운행의 출발시간, 출발지, 도착시간, 도착지로 한다. 결과적으로 스케줄링의 기본단위가 216개의 운행에서 48개의 확장운행으로 변환된 것이다. 이들 확장운행들과 차고지(차고지 번호를 150으로 함)를 교점으로 하는 새로운 네트워크 G′을 구할 수 있고, 식(1)을 이용하여 하루(계획기간)동안에 운행할 최적의 버스 스케줄을 구할 수 있다. 예제에 대한 최적의 버스 스케줄은 〈표 4〉와 같고, 'LINDO' 프로그램의 0-1 정수계획법에 의해 해를 구하였다.

〈표 2〉 예제의 운행 수 및 확장운행

구 분		노선 1	노선 2
시간대 별 운행수	6:00~ 7:00	101~106(6개 운행)	501~508(8개 운행)
	7:00~ 9:00	107~132(26개 운행)	509~544(36개 운행)
	9:00~17:00	133~196(64개 운행)	545~620(76개 통행)
B, D에서 출발하는 운행의 연결(확장운행)	6:00~ 7:00	-	(501, 507), (502, 508): 2개 운행연결
	7:00~ 9:00	(107, 126), (108, 127), …, (113,132): 7개 운행 연결	(509, 533), (510, 534), …, (520, 544): 12개 운행 연결
	9:00~17:00	(133, 169), (134, 170), …, (160, 196): 28개 운행 연결	(545, 586), (546, 587), …, (579, 620): 35개 운행 연결

〈표 3〉 선입선출법에 의한 확장운행 구축 결과

노선 1			노선 2		
확장운행 번호	확장운행에 속한 운행번호	출발지 시작시간 도착지 종료시간	확장운행 번호	확장운행에 속한 운행번호	출발지 시작시간 도착지 종료시간
101	104	B 6:00 A 6:45	127	505, 503	D 6:00 D 7:00
102	105	B 6:20 A 7:05	128	506, 504	D 6:15 D 7:15
103	106	B 6:40 A 7:25	129	501, 507	C 6:00 C 7:00
104	101	A 6:00 B 6:45	130	502, 508	C 6:15 C 7:15
105	102	A 6:20 B 7:05	131	527, 515, 539	D 7:00 C 9:04
106	103	A 6:40 B 7:25	132	528, 516, 540	D 7:07 C 9:11
107	107, 126, 119	A 7:00 B 10:00	133	529, 517, 541	D 7:14 C 9:18
108	108, 127	A 7:10 A 9:10	134	530, 518, 542	D 7:21 C 9:25
109	109, 128	A 7:20 A 9:20	135	531, 519, 543	D 7:28 C 9:32
110	110, 129	A 7:30 A 9:30	136	532, 520, 544	D 7:35 C 9:39
111	111, 130	A 7:40 A 9:40	137	509, 533, 521	C 7:00 D 9:04
112	112, 131	A 7:50 A 9:50	138	510, 534, 522	C 7:07 D 9:11
113	120, 113, 132	B 7:00 A 10:00	139	511, 535, 523	C 7:14 D 9:18
114	121, 114	B 7:10 B 9:10	140	512, 536, 524	C 7:21 D 9:25
115	122, 115	B 7:20 B 9:20	141	513, 537, 525	C 7:28 D 9:32
116	123, 116	B 7:30 B 9:30	142	514, 538, 526	C 7:35 D 9:39
117	124, 117	B 7:40 B 9:40	143	583, 548, 589, 554, 595, 560, 601, 566, 607, 572, 613, 578,, 619	D 9:10 C 17:33
118	125, 118	B 7:50 B 9:50	144	584, 549, 590, 555, 596, 561, 602, 567, 608, 573, 614, 579, 620	D 9:23 C 17:46
119	165, 137, 173, 145, 181, 153, 189, 161	B 9:15 B 17:05	145	585, 550, 591, 556, 597, 562, 603, 568, 609, 574, 615, 580	D 9:36 D 17:20
120	166, 138, 174, 146, 182, 154, 190, 162	B 9:30 B 17:20	146	545, 586, 551, 592, 557, 598, 563, 604, 569, 610, 575, 616, 581	C 9:10 D 17:33
121	167, 139, 175, 147, 183, 155, 191, 163	B 9:45 B 17:35	147	546, 587, 552, 593, 558, 599, 564, 605, 570, 611, 576, 617, 582	C 9:23 D 17:46
122	168, 140, 176, 148, 184, 156, 192, 164	B 10:00 B 17:50	148	547, 588, 553, 594, 559, 600, 565, 606, 571, 612, 577, 618	C 9:36 C 17:20
123	169, 133, 141, 177, 149, 185, 157, 193	A 9:15 A 17:05			
124	170, 134, 142, 178, 150, 186, 158, 194	A 9:30 A 17:20			
125	171, 135, 143, 179, 151, 187, 159, 195	A 9:45 A 17:35			
126	172, 136, 144, 180, 152, 188, 160, 196	A 10:00 A 17:50			

2. 개발 모형의 평가

〈표 4〉의 버스 스케줄에서 보는 것처럼 2개 노선 운행에 필요한 버스대수는 24대이고, 이중 5대(3, 9, 14, 16, 17번)는 2개 노선을 교차 운행하는 스케줄이며, 11대의 버스는 노선 1을, 8대는 노선 2 만을 운행하는 스케줄이다. 연결운행시간을 포함한 총 공차운행시간은 2,174분이고 각각의 노선을 독립적으로 운행했다면 3,052분(노선 1이 1,302분, 노선 2가 1,750분)의 공차운행시간이 필요했을 것이다.

동일한 운행특성을 갖는 시간대별로 각각 몇 개씩의 운행들이 포함되어 있는가에 따라 다시간대 버스 스케줄링 방법에 의해 스케줄링의 기본단위의 개수를 어느 정도로 줄일 수 있는 가를 결정지을 수 있다. 그러나 이러한 특성들은 버스운행시간표에 따라 각각 다르기 때문에 본 논문에 의한 방법으로 버스 스케줄링 문제의 크기를 어느 정도 줄일 수 있을 것인가를 수식으로 나타내기는 곤란하다.

그런데 앞의 예제의 경우를 통해 살펴보면, 2개 노선에 대한 1회 운행은 총 216개였다. 이를 이른 아

〈표 4〉 예제에 대한 버스 스케줄링 결과

스케줄 번호	노선 1 (확장운행 번호)	노선 2 (확장운행 번호)
1	150→101→107→150	
2	150→103→110→124→150	
3	150→104→	137→143→150
4	150→105→114→119→150	
5	150→106→116→120→150	
6	150→109→150	
7	150→111→125→150	
8	150→112→150	
9	150←118	←128←150
10	150→113→150	
11	150→115→150	
12	150→117→150	
13	150→102→108→123→150	
14	150←121	←135←127←150
15		150→129→131→146→150
16	150←122	←140←150
17	150←126	←136←150
18		150→130→134→150
19		150→132→150
20		150→133→147→150
21		150→138→148→150
22		150→139→144→150
23		150→141→145→150
24		150→142→150

침과 출근 시간대 및 평시인 3개의 시간대로 분할하고, 선입선출법으로 각 노선의 종착지(B, D)에서 84개의 운행을 서로 연결시켜 새로운 스케줄의 기본단위를 132개로 줄였다. 이를 다시 A, C지점에서 출발하는 운행들을 선입선출법으로 연결시킨 결과 노선 1에서 26개의 확장운행을 얻었고, 노선 2에 대해서는 22개의 확장운행을 얻었다. 결국 48개의 확장운행을 교점으로 하는 새로운 네트워크가 작성된 것이다.

결과적으로 버스 스케줄링의 기본단위가 216개에서 48개로 줄어든 것이다. 즉 버스 스케줄을 구하기 위한 교통배정배정문제 알고리즘의 계산량이 $n^3=216^3$에서 48^3으로 줄어든 것이다. 그러나 버스운행시간표를 시간대별로 분할하고, 선입선출법으로 각 운행을 연결시키는 계산량과 이들 자료를 다시 재정리하는 계산량은 추가로 발생되었다. 그리고 시간대별로 작성한 확장운행들은 전체네트워크를 중복 없이 분할한 시간대별 부분네트워크의 최단경로들로서 전체네트워크의 최단경로를 구성하는 부분경로임으로, 216개의 운행들을 기본 단위로 하는 스케줄링의 결과와 시간분할에 의한 스케줄링의 결과는 동일하게 된다. 단, 다른 노선으로의 공차이동시간 보다 주어진 노선의 운행시격이 짧은 경우에 한한다.

V. 결론 및 향후 연구과제

1. 결론

본 연구는 선진 외국에서 많은 연구가 이루어지고 있고, 전산시스템이 개발되어 상용화되고 있는 버스 스케줄링 문제를 우리나라 현실에 쉽게 적용하기 위한 시도라고 할 수 있다. 그래서 본 연구에서는 버스 스케줄링 문제의 개요 및 해를 구하기 위한 기존의 방법들을 설명하였다. 기존의 방법에 의할 경우 버스 스케줄링 문제의 크기는 하루에 운행할 운행회수에 비례하기 때문에 고려해야할 운행회수가 많을 경우, 스케줄링 문제의 크기가 커져 수리적 모형으로 해를 구하는데 어려움이 뒤따른다.

그런데 우리나라에서 운행되는 많은 버스들은 시간대별로 동일한 운행시격과 운행소요시간을 기준으로 배차되고 있다. 이런 상황에서는 하루의 운행시간을 운행특성이 동일한 시간대로 분할하고, 선입선출법에 의해 원래의 1회 운행을 확장운행으로 전환함으로써, 스케줄링의 기본단위의 개수를 줄이고, 스케줄링 문제의 크기 역시 작아지기 때문에, 원래의 문제 보다 크기가 훨씬 작은 문제에서 최적의 버스 스케줄을 구할 수 있다. 본 논문은 이와 같은 과정에 의해 버스 스케줄을 구하는 시간 분할에 의한 다시간대 버스 스케줄링 방법을 제시하였으며, 예제를 통하여 다시간대 버스 스케줄링의 방법과 문제의 크기가 줄어드는 효과를 설명하였다.

2. 향후 연구과제

대중교통에 대한 스케줄링 문제는 크게 차량과 운전자 부문으로 구분된다. 그런데 버스회사의 운영비는 차량운행비 대비 운전비가 약 2배에 달하고 있다(한국산업관계연구원, 1997). 이는 버스 스케줄링에 의해 얻을 수 있는 효과 보다 운전자 스케줄링에 의

한 효과가 훨씬 클 수 있다는 반증으로 해석된다. 물론 버스 스케줄이 운전비의 많은 부분을 개선하고, 운전자 스케줄링의 입력자료가 되기 때문에 운전자 스케줄링을 위해서도 버스 스케줄링 시스템은 필요하다. 버스대중교통을 보다 효율적으로 운영하기 위해서는 버스 스케줄링뿐만 아니라 운전자 스케줄링에 대한 연구가 필요함을 강조하는 것이다.

본 논문은 노선별 버스의 시간대별 운행시간이 일정하다는 가정 하에서 버스 스케줄을 작성하는 방법을 제시하였다. 그러나 최근에는 노면교통의 혼잡 등으로 인해 버스의 지연운행이 종종 발생하고 있다. 사전에 많은 노력을 들여 효율적인 버스 스케줄을 작성한다고 할 지라도, 예정된 시간에 차량이 목적지에 도착하지 못한다면, 후속 운행은 지연 출발할 수밖에 없다. 이를 방지하기 위해서는 자동차량위치측정기법(AVL)을 이용한 버스 스케줄의 실시간 조정 및 관리 등에 대한 연구가 이루어져야 할 것이다.

참고문헌

1. (재)한국산업관계연구원(1997), " '96 서울시내버스 요금 운송원가 검증 및 경영분석(I)", 1997. 2, p.136.
2. Bertosi, A., Carraresi, P. and Gallo, G. (1987), "On Some Matching Problems Arising in Vehicle Scheduling Models", Networks, Vol. 17, pp.271~281.
3. Carpaneto, G., Dell'Amico, M., Fischetti, M., Toth, P.(1989), "A Branch and Bound Algorithm for the Multiple Depot Vehicle Scheduling Problem", Networks, Vol. 19, pp.531~548.
4. Daduna, J. R., Pinto Paixão, J. M.(1995), "Vehicle Scheduling for Public Mass Transit -An Overview", Computer-Aided Transit Scheduling, Proceedings of the Sixth International Workshop on Computer-Aided Scheduling of Public Transport, Joachim R. Daouna Isabel Brancó, Jose M. Pinto Paixão(ed.), Springer, pp.76~90.
5. EL-AZM, A.(1985), "The Minimum Fleet Size Problem and its Applications to Bus Scheduling", Computer Scheduling of Public Transport 2, J.-M. Rousseau(ed.), © Elsevier Science Publishers B. V. (North-Holland), pp.493~512.
6. Linus Schrage(1986), "User's Manual : Linear, Integer and Quadratic Programming with LINDO, third edition", The Scientific Press, 1986, pp.185~206.
7. Orloff, C. S.(1976), "Route constraint fleet scheduling", Transportation Science, Vol. 10, No. 2, pp.149~168.
8. Stern, H. I. and Ceder, A.(1981), " A Deficit Function Approch for Bus Scheduling", Computer Scheduling of Public Transport, Wern, A. (ed.), © North-Holland Publishing Company, pp.85~96.

첨부자료 : 버스운행시간표 및 시간대별 부분운행집합

노선 1

번호	출발지	출발시간	도착지	도착시간	번호	출발지	출발시간	도착지	도착시간
$I_{1,p,1}$(6:00 ~7:00, h = 20분, d = 45분)									
101	A	6:00	B	6:45	104	B	6:00	A	6:45
102	A	6:20	B	7:05	105	B	6:20	A	7:05
103	A	6:40	B	7:25	106	B	6:40	A	7:25
$I_{1,p,2}$(7:00 ~ 9:00, h = 10분, d = 60분)									
107	A	7:00	B	8:00	120	B	7:00	A	8:00
108	A	7:10	B	8:10	121	B	7:10	A	8:10
109	A	7:20	B	8:20	122	B	7:20	A	8:20
110	A	7:30	B	8:30	123	B	7:30	A	8:30
111	A	7:40	B	8:40	124	B	7:40	A	8:40
112	A	7:50	B	8:50	125	B	7:50	A	8:50
113	A	8:00	B	9:00	126	B	8:00	A	9:00
114	A	8:10	B	9:10	127	B	8:10	A	9:10
115	A	8:20	B	9:20	128	B	8:20	A	9:20
116	A	8:30	B	9:30	129	B	8:30	A	9:30
117	A	8:40	B	9:40	130	B	8:40	A	9:40
118	A	8:50	B	9:50	131	B	8:50	A	9:50
119	A	9:00	B	10:00	132	B	9:00	A	10:00
$I_{1,p,3}$(9:00 ~ 17:00, h = 15분, d = 50분)									
133	A	9:15	B	10:05	165	B	9:15	A	10:05
134	A	9:30	B	10:20	166	B	9:30	A	10:20
135	A	9:45	B	10:35	167	B	9:45	A	10:35
136	A	10:00	B	10:50	168	B	10:00	A	10:50
137	A	10:15	B	11:05	169	B	10:15	A	11:05
138	A	10:30	B	11:20	170	B	10:30	A	11:20
139	A	10:45	B	11:35	171	B	10:45	A	11:35
140	A	11:00	B	11:50	172	B	11:00	A	11:50
141	A	11:15	B	12:05	173	B	11:15	A	12:05
142	A	11:30	B	12:20	174	B	11:30	A	12:20
143	A	11:45	B	12:35	175	B	11:45	A	12:35
144	A	12:00	B	12:50	176	B	12:00	A	12:50
145	A	12:15	B	13:05	177	B	12:15	A	13:05
146	A	12:30	B	13:20	178	B	12:30	A	13:20
147	A	12:45	B	13:35	179	B	12:45	A	13:35
148	A	13:00	B	13:50	180	B	13:00	A	13:50
149	A	13:15	B	14:05	181	B	13:15	A	14:05
150	A	13:30	B	14:20	182	B	13:30	A	14:20
151	A	13:45	B	14:35	183	B	13:45	A	14:35
152	A	14:00	B	14:50	184	B	14:00	A	14:50
153	A	14:15	B	15:05	185	B	14:15	A	15:05
154	A	14:30	B	15:20	186	B	14:30	A	15:20
155	A	14:45	B	15:35	187	B	14:45	A	15:35
156	A	15:00	B	15:50	188	B	15:00	A	15:50
157	A	15:15	B	16:05	189	B	15:15	A	16:05
158	A	15:30	B	16:20	190	B	15:30	A	16:20
159	A	15:45	B	16:35	191	B	15:45	A	16:35
160	A	16:00	B	16:50	192	B	16:00	A	16:50
161	A	16:15	B	17:05	193	B	16:15	A	17:05
162	A	16:30	B	17:20	194	B	16:30	A	17:20
163	A	16:45	B	17:35	195	B	16:45	A	17:35
164	A	17:00	B	17:50	196	B	17:00	A	17:50

노선 2

번호	출발지	출발시간	도착지	도착시간	번호	출발지	출발시간	도착지	도착시간
$I_{2,p,1}$(6:00 ~ 7:00, h = 15분, d = 30분)									
501	C	6:00	D	6:30	505	D	6:00	C	6:30
502	C	6:15	D	6:45	506	D	6:15	C	6:45
503	C	6:30	D	7:00	507	D	6:30	C	7:00
504	C	6:45	D	7:15	508	D	6:45	C	7:15
$I_{2,p,2}$(7:00 ~ 9:00, h = 7분, d = 35분)									
509	C	7:00	D	7:40	527	D	7:00	C	7:40
510	C	7:07	D	7:47	528	D	7:07	C	7:47
511	C	7:14	D	7:54	529	D	7:14	C	7:54
512	C	7:21	D	8:01	530	D	7:21	C	8:01
513	C	7:28	D	8:08	531	D	7:28	C	8:08
514	C	7:35	D	8:15	532	D	7:35	C	8:15
515	C	7:42	D	8:22	533	D	7:42	C	8:22
516	C	7:49	D	8:29	534	D	7:49	C	8:29
517	C	7:56	D	8:36	535	D	7:56	C	8:36
518	C	8:03	D	8:43	536	D	8:03	C	8:43
519	C	8:10	D	8:50	537	D	8:10	C	8:50
520	C	8:17	D	8:57	538	D	8:17	C	8:57
521	C	8:24	D	9:04	539	D	8:24	C	9:04
522	C	8:31	D	9:11	540	D	8:31	C	9:11
523	C	8:38	D	9:18	541	D	8:38	C	9:18
524	C	8:45	D	9:25	542	D	8:45	C	9:25
525	C	8:52	D	9:32	543	D	8:52	C	9:32
526	C	8:59	D	9:39	544	D	8:59	C	9:39
$I_{2,p,3}$(9:00 ~ 17:00, h = 13분, d = 35분)									
545	C	9:10	D	9:45	583	D	9:10	C	9:45
546	C	9:23	D	9:58	584	D	9:23	C	9:58
547	C	9:36	D	10:11	585	D	9:36	C	10:11
548	C	9:49	D	10:24	586	D	9:49	C	10:24
549	C	10:02	D	10:37	587	D	10:02	C	10:37
550	C	10:15	D	10:50	588	D	10:15	C	10:50
551	C	10:28	D	11:03	589	D	10:28	C	11:03
552	C	10:41	D	11:16	590	D	10:41	C	11:16
553	C	10:54	D	11:29	591	D	10:54	C	11:29
554	C	11:07	D	11:42	592	D	11:07	C	11:42
555	C	11:20	D	11:55	593	D	11:20	C	11:55
556	C	11:33	D	12:08	594	D	11:33	C	12:08
557	C	11:46	D	12:21	595	D	11:46	C	12:21
558	C	11:59	D	12:34	596	D	11:59	C	12:34
559	C	12:12	D	12:47	597	D	12:12	C	12:47
560	C	12:25	D	13:00	598	D	12:25	C	13:00
561	C	12:38	D	13:13	599	D	12:38	C	13:13
562	C	12:51	D	13:26	600	D	12:51	C	13:26
563	C	13:04	D	13:39	601	D	13:04	C	13:39
564	C	13:17	D	13:52	602	D	13:17	C	13:52
565	C	13:30	D	14:05	603	D	13:30	C	14:05
566	C	13:43	D	14:18	604	D	13:43	C	14:18
567	C	13:56	D	14:31	605	D	13:56	C	14:31
568	C	14:09	D	14:44	606	D	14:09	C	14:44
569	C	14:22	D	14:57	607	D	14:22	C	14:57
570	C	14:35	D	15:10	608	D	14:35	C	15:10
571	C	14:48	D	15:23	609	D	14:48	C	15:23
572	C	15:01	D	15:36	610	D	15:01	C	15:36
573	C	15:14	D	15:49	611	D	15:14	C	15:49
574	C	15:27	D	16:02	612	D	15:27	C	16:02
575	C	15:40	D	16:15	613	D	15:40	C	16:15
576	C	15:53	D	16:28	614	D	15:53	C	16:28
577	C	16:06	D	16:41	615	D	16:06	C	16:41
578	C	16:19	D	16:54	616	D	16:19	C	16:54
579	C	16:32	D	17:07	617	D	16:32	C	17:07
580	C	16:45	D	17:20	618	D	16:45	C	17:20
581	C	16:58	D	17:33	619	D	16:58	C	17:33
582	C	17:11	D	17:46	620	D	17:11	C	17:46