

지역 교통망 관리를 위한 최적 신호순서에 관한 연구*

성기석**

A Study on the Optimal Signal Timing for Area Traffic Control*

Kiseok Sung**

■ Abstract ■

A genetic algorithm to determine the optimal signal sequence and double cycle pattern is described. The signal sequence and double cycle pattern are used as the input for TRANSYT to find optimal signal timing at each junction in the area traffic networks. In the genetic process, the partially matched crossover and simple crossover operators are used for evolution of signal sequence and double cycle pattern respectively. A special conversion algorithm is devised to convert the signal sequence into the link-stage assignment for TRANSYT. Results from tests using data from an area traffic network in Leicester region R are given.

1. 서론

지역 교통망(Area Traffic Network)이란 주로 도시의 가로 망과 같이 일정한 지역 내에서 도로와 교차로들로 서로 연결되어 이루어진 도로망을 가리킨다. 각 교차로에서는 신호등을 이용하여 각 방향별 차량의 진입을 통제하게 된다. 교차로에서의 신호는, 신호주기(Cycle)와, 각 방향 별로 정해진 신호시간(Green time) 및 신호순서(Signal sequence) 등에 따라서 차량의 흐름을 통제한다. 독립된 교차

로에서의 차량의 정체는 주로 신호주기와 신호시간의 설정과 관련이 있다. 그러나 지역 교통망에서는 독립된 하나의 교차로만을 관리하는 경우와는 달리, 지역 내에 있는 다수의 교차로들을 동시에 고려하여 지역 전체의 차량 흐름이 원활해지도록 해야 한다. 이때 지역 내에서 인접한 교차로 상호간의 신호순서가 전체 지역내의 차량의 정체에 많은 영향을 미친다. 따라서 각 교차로에서의 신호주기와 신호시간 뿐만 아니라, 신호순서도 최적화하여야 한다.

* 이 논문은 1997년 한국학술진흥재단의 공모과제 연구비에 의하여 연구되었음.

** 강릉대학교 산업공학과

Allsop[1]는 신호 국면(signal phase), 신호 단계(signal stage), 신호주기(signal cycle), 신호국면간의 호환성(compatibility) 및 신호시간 결정에 있어서의 제약 조건(constraints) 등을 정의하였다. 신호 국면이란 교차로 내에서 하나 또는 그 이상의 차량 흐름의 방향에 대하여 주어지는, 한번의 연속되는 진행 신호가 주어지는 시간 구간을 말한다. 신호 단계는 특정한 방향의 차량 흐름들이 진행 신호를 받게되는, 신호주기상의 시간 구간들이고, 이러한 신호 단계들이 순서대로 완전히 한번 순환하면 하나의 신호주기를 형성하게 된다. 즉, 신호 국면은 각 진행 방향별로 주어지는 진행 신호의 구간을 가리키고, 신호 단계는 해당 교차로의 주기시간을 임의의 간격으로 분할한 시간 구간을 가리킨다.

교차로에서의 신호시간 결정을 위한 모델은 크게 신호 단계를 기본으로 한 것과(Stage based signal control) 신호 국면을 기본으로 한 것(Phase based signal control)의 두 가지로 구분할 수 있다. 앞의 것은 고전적인 신호시간 결정 모델로서, 신호 단계의 길이만을 결정하기 위하여 세워진 것으로서 단순하다. 즉, 이 모델은 신호의 순서(Signal sequence)는 전혀 고려하지 못한다. TRANSYT (TRAffic Network Study Tool)은 이 모델을 적용하고 있다[11]. 뒤의 모델은 앞의 모델보다 개선된 것이며, 실제 적용에 있어서 융통성이 크다. 그러나 이 모델은 수학적으로 모형화하고 최적해를 찾기가 더 어려운 단점이 있다. 즉, 신호순서를 최적화할 수 있도록 수리 모형을 세울 수는 있지만 그것을 풀어서 최적해를 구하기는 매우 어렵다. 이러한 모델로서는 Silcock[7] 등이 제안한 SIGSIGN와, Simmonite[8] 등이 제안한 LINSIGII 등이 있는데, 이들 모델은 지역교통망 전체의 교차로에 대해서 최적해를 구하지 못하고 단지 하나의 교차로에 대해서만 최적해를 구한다.

Heydecker[4] 등은 지역 교통망에서의 신호시간

을 결정하는 모델을 이단수리계획 문제(Bi-level programming)로 모델링하였다. 즉, 그들은 각 교차로에서의 신호순서에 대해서는 정수 변수를, 신호시간에 대해서는 실수 변수를 사용하여 표현하고, 이단수리계획 모형을 적용하여 신호시간, 신호주기 및 신호순서를 최적화하려고 하였다. 이 때 그들은 신호순서의 조합수가 너무 많아서 계산상의 어려움이 있자, 동등 순서(Equivalent sequence)와 완전 망(Clique)을 이용하여 계산량을 줄이려고 시도했으나 그리 효과적이지 못했다.

Hadil[6] 등은 TRANSYT-7F와 Simulated Annealing을 이용하여 지역 교통망에서 신호순서를 결정하는 모델을 세웠다. 그들은 여러 개의 신호순서 패턴을 미리 정하여 놓고 각 패턴들을 하나의 정수 값에 대응시킨 후 그 정수 변수들에 대하여 Simulated Annealing을 수행하여 가장 좋은 수행도 지수 값을 주는 패턴을 골라내는 방식을 제안하였다. 그들의 접근 방법에서는 Simulated Annealing의 수렴 속도가 너무 느려서 계산량이 많아지고, 고려 대상이 되는 신호순서의 패턴이 제한적일 수밖에 없는 문제가 있었다.

지금까지 국내외의 연구 현황을 보면, 각 교차로에서 각 방향별 신호순서를 결정하는 문제의 어려움 때문에 지역 교통망에서의 신호시간을 최적으로 결정하지 못하고 있음을 알 수 있다. 본 연구에서는 유전 알고리즘(Genetic algorithm)을 TRANSYT 프로그램에 적용하여 이중주기 패턴과 각 교차로에서의 신호순서의 조합수가 크기 때문에 발생하는 계산상의 어려움을 극복하고 교차로에서의 신호순서와 신호주기의 형태를 최적화하는 방안을 제시한다.

2. 연구 배경

TRANSYT은 지역 교통망에서 고정식 신호주기를 설정해주는 프로그램이다. 현재 영국에서 온-

라인, 가변식, 또는 감응식 신호주기 설정 시스템이라 할 수 있는 SCOOT 시스템이 개발되어 사용되고 있는데, 이 시스템은 TRANSYT을 기초로 하여 감응식으로 발전된 시스템이다. 최초의 TRANSYT version은 1969년 처음으로 개발되었으며, 그 이후 계속하여 그 기능이 개선되어서 현재 version 9까지 개선되었다[2]. 본 연구에서는 TRANSYT version 8을 사용하였다.

TRANSYT은 교차로 자료, 연결 도로의 차선별 자료, 도로별 교통량 자료, 적용하고자 하는 공통주기 시간과 이중주기 패턴 및 각 교차로에서의 각 방향별 신호순서 등을 그 입력 자료로서 받아들인 후 Pattern Matching 기법의 하나인 Hill Climb Method를 사용하여 각 교차로에서의 각 방향별 신호시간과 오프셋을 구해준다. 이때 TRANSYT은 정해진 임의의 각 방향별 신호시간에 대해서 수행도 지수(Performance Index)를 계산하는데, 이 지수는 지역 내에서 각 차량의 정지 수(Stop number), 대기 시간 및 연료 소모량 등으로부터 산출된다. Hill Climb Method는 이러한 수행도 지수를 최소로 하는 각 방향별 신호시간과 오프셋(Offset)을 찾는 발견적 방법(Heuristic Method)이다. 여기서 신호시간은 임의의 방향의 신호가 진행 가능 상태로 유지되는 시간(Green time)을 말하고, 오프셋은 지역의 모든 교차로에 대하여 적용하는 공통주기의 시작 시점을 기준 시점으로 하여 임의의 교차로에서 주기 시간이 시작하는 시점과의 차이를 말한다.

TRANSYT은 지역 교통망 안에 있는 다수의 교차로에 대하여, 신호주기, 각 방향별 신호시간 및 신호순서를 정하는 프로그램으로서, 지금까지 가장 효율적이라고 알려져 있다. 이 프로그램은 일종의 시뮬레이션 기법과, 발견적 기법인 Hill Climb 방법을 이용하여 지역내 각 교차로에서의 신호주기와 각 방향별 신호시간을 적절하게 결정하여 준다. 그러나 TRANSYT은 각 교차로에서

의 신호순서에 대해서 최적해를 스스로 구하지 못하고 단지 설정에 필요한 가이드만 제시해 준다. 따라서 사용자가 그 가이드를 보고 경험에 의거하여 각 교차로에서의 신호순서를 입력시켜 주면 TRANSYT은 그 입력에 따라 신호시간과 신호주기를 정하게 된다. 그런데 실제 지역 교통망에서는 인접한 교차로들 간의 신호순서에 따라서 차량 지체의 정도가 많은 영향을 받으므로 신호순서를 최적화하는 것이 지역내의 교통 흐름을 원활히 하는데 중요한 역할을 한다. 그러나 지역 내에서 관리하여야 할 교차로의 수가 증가함에 따라 교차로들 간의 신호순서를 최적으로 결정하기가 매우 어려워진다. 왜냐하면 각 교차로에서의 가능한 신호순서의 조합 수가 지수 승으로 늘어나기 때문이다.

또한 신호주기를 정함에 있어서는 지역 내의 모든 교차로가 공통적으로 사용하는 신호주기를 공통주기라고 하는데, 이 공통주기의 길이를 얼마로 정하느냐에 따라서 각 교차로에서의 차량의 정체가 영향을 받는다. TRANSYT에서는 이 공통주기의 길이를 최적화한다. 한편으로 임의의 교차로에 대해서 공통주기 동안에 두 번의 신호주기가 진행 되도록 하는, 즉 이중주기를 적용하도록 지정할 수가 있다. 이러한 이중주기는 일반적으로 교통량이 적은 교차로에서 차량의 정지 수를 줄이기 위해서 적용하는데, 실제로 어느 교차로에 대하여 이중주기를 적용함으로써 신호의 수행도 지수를 높일 수 있을 것인가 하는 것이 문제이다. 이러한 각 교차로에서 적용해야 할 신호주기의 형태도 역시 TRANSYT에서 최적화하지 못하고 단지 가이드만 제시해준다. 따라서 사용자가 경험에 의하여 각 교차로에 대하여 이중주기를 적용할 것인지 아닌지를 TRANSYT에 입력시켜 주어야 한다. 교차로의 수가 증가함에 따라 이중주기를 적용하는 패턴의 조합 수가 지수 승으로 늘어나므로, 최적의 이중주기 패턴을 구하는 것도 어려운 문제이다.

유전 알고리즘은 자연계에서 생물이 유전적으로 진화해 가는 방식을 모방하여 최적해를 찾는 해법이다[3]. 유전알고리즘에서는 시스템의 상태나 개별적인 구조들을 염색체(Chromosome)의 형태로 표현하고, 염색체들이 서로 선택(Selection), 교배(Crossover), 복제(Reproduction) 및 돌연변이(Mutation)의 과정을 통하여 시스템의 상태를 최적화하거나 최적인 개체의 구조를 찾아낸다. 이러한 생물 집단의 진화를 모방한 수리적 진화 과정에서 개체의 염색체의 집합인 세대(Generation)가 다음 세대로 진화해 간다. 즉, 개별 구조의 집합체가 더 나은 개별 구조의 집합체로 진화해 간다.

본 논문에서는 지역내 각 교차로의 이중주기 패턴과 각 교차로에서의 각 방향별 신호순서를 염색체의 형태로 표현하고 이것을 선택, 교배, 복제 및 돌연변이 하는 과정을 거치면서 빠른 시간 내에 수행도 지수를 최소화하는 이중주기 패턴과 각 교차로에서의 각 방향별 신호순서를 가지는 세대로 수렴되도록 하였다.

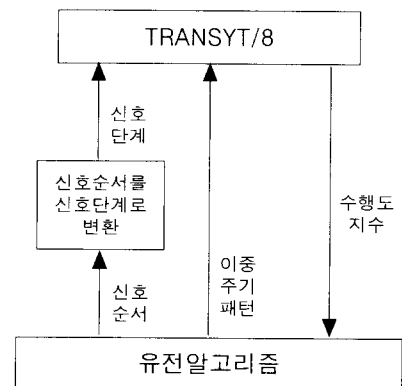
각 교차로에서의 각 방향별 신호순서를 염색체 형태로 표현할 때, 하나의 지역 내의 신호순서의 패턴에 대한 충분한 정보가 그에 대응되는 하나의 염색체 안에 내포되어야 한다. 또 각 염색체들이 교배하여 생성된 자손이 역시 그 자체로써 가능한 지역 내의 신호 패턴중의 가능한(Feasible) 하나를 표현하여야 한다. 그렇게 되기 위해서는 교배 방법 또한 연구되어야 한다. 그리고 복제 과정에서는 적합도(Fitness) 값을 필요로 하는데, 이를 위하여 TRANSYT에서 구해지는 수행도 지수를 사용했다. 즉, 복제 과정 중에 임의의 각 방향별 신호순서가 가질 수 있는 최적의 수행도 지수를 구하기 위해서 TRANSYT을 연결하여 사용했다.

3. 계산 모델

여기서 제시하는 이중주기 패턴과 최적 신호순

서를 찾는 계산 모델은 다음 [그림 1]과 같다. 유전 알고리즘에서 생성된 이중주기 패턴과 신호순서들은 TRANSYT의 입력에 적합한 형태로 변환되어 TRANSYT에 입력된다. TRANSYT에서는 입력된 이중주기 패턴과 신호순서를 그대로 유지하면서 신호시간과 신호주기를 최적화한다. 그리고 그때의 수행도 지수를 유전알고리즘에서 사용할 수 있도록 한다. 유전알고리즘에서는 다음 세대의 신호순서들을 생성하기 위한 복제, 선택 및 교배 연산에 수행도 지수를 이용한다.

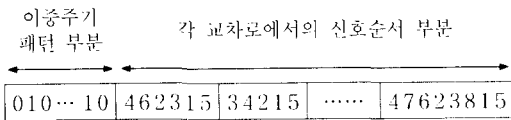
한편, 유전알고리즘 부분에 있어서, 일반적인 염색체 표현 방식을 이용할 경우 신호순서를 나타낼 수 없고, 또 염색체들을 교배한 후에 생성되는 다음 세대의 염색체가 적용 가능한(Feasible) 신호순서를 나타낸다는 보장이 없다. 따라서 이러한 염색체의 표현 방식 문제를 해결하기 위해서 특별한 염색체 표현 방식과 교배 방법을 적용하였다. 즉, 일반적인 염색체 표현 방법인 0과 1의 나열이 아니라, 순서(Sequence)를 표현 할 수 있도록 a-c-f-d-b-e 등과 같은 형태로 염색체를 표현하였다. 또 교배 방법에 있어서는 부분대응 교배(Partially Matched Crossover) 방법을 사용하여 교배 후에 생성된 염색체가 표현하는 신호순서의 실행가능성(Feasibility)을 유지하도록 하였다.



[그림 1] 계산 모델

3.1 이중주기 패턴과 신호순서를 나타내는 염색체 표현

염색체 표현은 이중주기 패턴을 나타내는 부분과 신호순서를 나타내는 두 부분으로 나뉜다. 이중주기 패턴은 지역내의 교차로의 수만큼의 길이를 갖는 0과 1의 조합으로 된 숫자 열을 사용한다. 신호순서는 각 교차로 별로 각 신호 방향을 나타내는 숫자를 신호순서대로 나열한 숫자 열을 사용하는데, 그 길이는 각 교차로 내에서의 신호 방향의 수만큼이 된다.



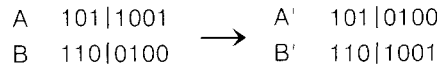
[그림 2] 해의 염색체 표현

3.2 교배 연산자(Crossover Operator)

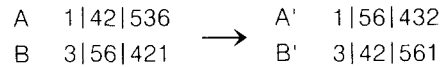
여기서 사용한 교배 연산자는 두 가지이다. 즉, 이중주기패턴 부분에 대해서는 단순 교차 교배(One Point Crossover) 방법을 적용하고, 신호순서 부분에 대해서는 부분 대응 교배(Partially Matched Crossover) 방법을 적용한다. 이중주기패턴 부분에 대해서는 단순 교차교배 방법을 적용하더라도 주기패턴의 실행가능성(Feasibility)이 그대로 유지되지만, 신호순서 부분의 경우에는 단순 교차 교배 방법을 사용하면 교배 후에 생성되는 염색체가 표현하는 신호순서의 실행가능성이 유지되지 않는다. 따라서 부분 대응 교배 방법을 적용하여 교배함으로써, 교배 후에 생성된 염색체가 표현하는 신호순서의 실행가능성이 유지되도록 하였다.

단순 교차 교배는 [그림 3]에서와 같이 두 염색체의 염색 인자 열에서 임의의 한 위치를 정하고 그 점을 기준으로 서로의 염색 인자 열을 서로 교환하는 것이다. 그러나 [그림 4]에서와 같이 염색 인자들 사이에 배열 순서가 고려되어야 하는 경우에는 교

배 후에 생성된 염색체의 염색 인자들의 순서가 의미를 가질 수 있도록 하여야 한다. 우선 두 개의 교차점을 임의로 정하는데, 이 두 교차점 사이에 있는 염색 인자들을 대응부분으로 하여 각 염색체에서 염색 인자들 사이의 위치를 교환한다. 즉, [그림 4]의 경우, 염색체 A에서 4와 5, 2와 6의 위치가 교환되었고, 염색체 B에서 마찬가지로 5와 4, 6과 2의 위치가 교환되었다. 결과적으로 A'과 B'의 염색체가 생성되었는데, 이들은 어느 한 인자도 중복됨이 없이 순서만 교환되어서 실행가능성이 유지되고 있다.



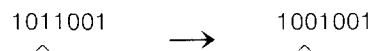
[그림 3] 단순 교차 교배



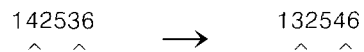
[그림 4] 부분 대응 교배

3.3 돌연변이 연산자(Mutation Operator)

돌연변이 연산자도 역시 두 가지를 사용한다. 즉, 이중주기 패턴 부분에 대해서는 역변환(Inversion) 방법을 적용하고, 신호순서 부분에 대해서는 순서 교환(Order Exchange) 방법을 적용한다. 이들 두 돌연변이 연산자들을 각각 이중주기 패턴과 신호순서 부분에 대해서 적용하였을 때, 항상 실행 가능성이 유지되는 돌연변이의 결과를 가져다준다. 역변환 돌연변이와 순서 교환 돌연변이를 설명하면 다음 [그림 5], [그림 6]과 같다.



[그림 5] 역변환 돌연변이



[그림 6] 순서 교환 돌연변이

3.4 복제(Reproduction)

교배연산과 돌연변이 연산 후에 생성된 새로운 세대의 염색체들의 적합도에 따라 각 염색체들이 다음 세대를 교배 생성하기 위한 부모로 선택될 확률이 달라지도록 한다. 즉, 적합도가 높은 염색체들은 될수록 많이 복제하여 다음 세대의 부모로서 선택될 확률이 높아지도록 하고, 적합도가 낮은 염색체는 적게 복제하여 그 선택 확률을 낮춘다. 물론 실제로 프로그램 상에서 구현할 때에는 똑같은 염색체를 복제하여 만드는 것은 아니며, 부모를 선택할 때, 룰렛과 같은 확률적 선택 방법을 사용함으로써, 마치 여러 개가 복제된 것과 같은 효과를 나타내도록 한다. 중요한 것은 각 염색체의 적합도를 어떻게 구할 것이며, 그 적합도에 따라서 선택될 확률을 어떻게 정할 것인가 하는 것이다.

3.4.1 적합도(Fitness)

우선 적합도는 각 염색체가 내포하는 신호주기 패턴과 신호순서를 TRANSYT의 입력 값으로 하고, TRANSYT을 수행하여 얻어지는 수행도 지수(Performance Index)의 역수를 적합도로 삼았다. 수행도지수란 지역내의 모든 교차로에서 발생하는 차량의 지체와 정지수로부터 산출되는 수치로서 이 수치가 적을수록 지역 교통망의 흐름이 원활하다고 볼 수 있다. 즉, 주어진 신호주기 패턴과 신호순서를 사용하여 TRANSYT이 최적의 신호 방법을 구하고, 그 신호 방법을 적용하였을 때 나타나는 지역 교통망의 흐름의 원활한 정도를 수행도지수라고 할 수 있다. 한편 TRANSYT에서 수행도지수를 산출하는 식은 다음과 같다.

$$\text{수행도지수}(P.I.) = \sum_{i=1}^{i=n} (d_i + K c_i)$$

단, d_i 는 교통망에서 i 번째 Link에서의 시간당 평균 지체수

c_i 는 i 번째 Link에서의 시간당 평균 정지수

K 는 가중치

이렇게 구해진 수행도지수는 그 값이 적을수록 좋으므로 그 역수를 적합도로 삼는다.

$$\text{적합도}(Fitness) = \frac{1}{\text{수행도지수}(P.I.)}$$

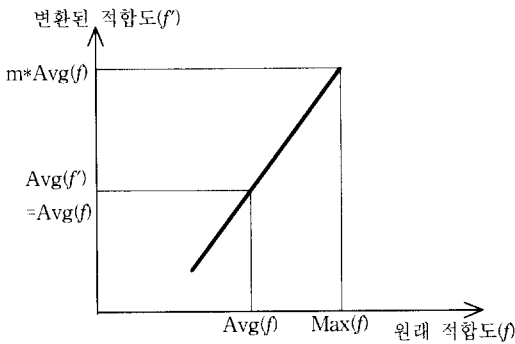
3.4.2 적합도의 변환(Fitness Scaling)

생성된 하나의 세대를 구성하는 각 염색체들의 적합도가 구해지고 난 뒤, 그 적합도에 따라 선택 확률을 정하게 된다. 그런데, 세대가 다르면 그에 따라 세대를 구성하는 염색체들의 적합도 편차가 달라진다. 이 적합도 편차가 너무 크면 각 염색체의 선택 확률의 편차가 너무 커져서 우성의 염색체들이 대표하는 부분 최적으로 수렴하기 쉬워진다. 적합도 편차가 작으면 선택 확률의 편차가 작아져서 염색체들 간의 우열을 구분하기가 힘들어진다. 따라서 이러한 적합도의 편차를 적정하게 되도록 각 염색체의 적합도를 변환해야 한다. 변환하는 방법에는 여러 가지가 있으나, 여기에서는 선형비례척도(Linear Scaling)를 사용한다.

선형비례척도에 의한 적합도의 변환 방법을 설명하면 다음과 같다. 우선 원래의 적합도를 변환하기 위한 직선의 식을 구하는데, 그 직선을 이용하여 변환하였을 때, 변환된 적합도들 중의 최대값이 원래 적합도의 평균의 m 배로 되고, 변환된 적합도들의 평균은 원래 적합도의 평균과 같아지도록 하는 직선의 식을 구한다. 그러한 직선의 식은 [그림 7]에 나타낸 것과 같다. 이러한 직선의 식에 의해서 적합도를 변환한다. 이때 사용한 평균의 배수 m 을 척도요소(Scale factor)라고 하는데, 보통 이 값은 1.5에서 2 사이의 값을 사용한다. 한편 여기서 사용한 변환을 위한 식이 직선의 형태이므로 선형비례척도라고 한다.

3.4.3 선택 확률의 결정

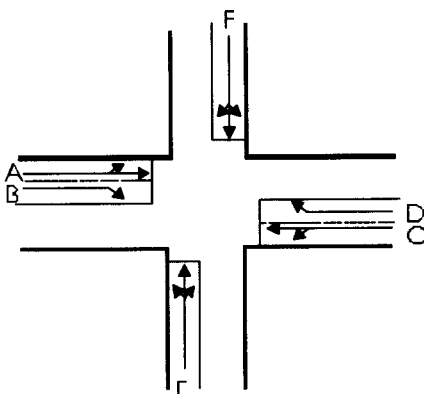
세대를 구성하는 각 염색체를 위와 같이 변환된 적합도만큼 복제한 것으로 하고, 룰렛에 의한 선택 방식을 적용하면 각 염색체의 선택 확률이 결정되게 된다.



[그림 7] 선형비례척도에 의한 적합도 변환

3.5 각 염색체가 내포하는 신호순서를 TRANSYT의 입력 형태로 변환

위의 염색체 표현에서 사용한 신호순서는 각 방향별 신호순서(Phase Sequence)를 나타낸다. 그런데 TRANSYT에서는 전체 신호주기를 몇 개의 단계(Stage)로 나누고, 각 신호 방향별로 진행 신호(Green Time)가 시작하는 단계와 끝나는 단계를 정해줘야 한다. 따라서 위의 각 방향별 신호순서를 TRANSYT의 입력으로 사용할 수 있도록 각 신호 단계에 할당된 신호 패턴으로 변환하여야 한다. 즉, 각 방향별로 동시에 진행 신호가 될 수 없는(Incompatible) 신호가 동일한 단계에서 진행 신호를 받을 수 없도록 하면서 각 방향 별 신호순서를 만족하도록 각 단계에 각 방향별 신호의 시작과



[그림 8] 교차로 예제

끝을 할당한다. 한편 서로 동시에 진행 신호가 될 수 없는 신호방향들을 비호환방향이라 부르고 교차로내의 가능한 모든 신호 방향들 간의 비호환성을 나타낸 행렬을 비호환방향 행렬이라 하자. <표 1>은 예로 든 [그림 8]의 좌측통행 방식을 따르는 교차로에서 나타나는 비호환방향 행렬이다.

<표 1> 비호환방향 행렬

	A	B	C	D	E	F
A				1	1	1
B			1		1	1
C		1			1	1
D	1				1	1
E	1	1	1	1		
F	1	1	1	1		

주어진 하나의 신호순서 목록에 대해서 다음과 같이 각 방향별 진행 신호의 시작과 끝을 각 단계에 할당한다.

단계 1.0 : I=1로 놓는다.

단계 1.1 : 신호순서 목록에서 I 번째 신호방향에 대해서, 그 목록에서 바로 뒤에 있는 첫 번째 비호환방향을 찾고 그것의 순서를 J라고 둔다.

단계 1.2 : I 번째 신호방향의 진행신호의 끝과 J 번째 신호방향의 진행신호의 시작을 J 번째 단계에 할당한다.

단계 1.3 : 신호순서 목록의 모든 신호방향에 대해서 진행신호 끝의 단계가 할당되었으면 단계 2.0으로 간다. 아니면 I=I+1로 두고 단계 1.1로 간다.

단계 2.0 : 신호순서 목록에서 시작 단계를 할당받지 못한 신호방향을 하나 찾고 그것의 순서를 I라고 둔다.

단계 2.1 : 신호순서 목록에서 I 번째 신호방향에 대해서, 그 목록에서 바로 앞에 있는 첫 번째 비호환방향을 찾고 그것의 순서를 J라고 둔다.

단계 2.2 : I 번째 신호방향의 진행신호의 시작을, J 번째 신호방향의 진행신호의 끝으로 할당된 단계에 할당한다.

단계 2.3 : 신호순서 목록에서 모든 신호방향에 진행 신호 시작의 단계가 할당되었으면 단계 3.0으로 간다. 아니면 단계 2.0으로 간다.

단계 3.0 : 각 단계에 대해서 어떠한 신호 방향의 신호 시작이나 끝으로도 할당되지 않은 단계를 제거한다.

단계 3.1 : 제거된 단계를 빼고 각 단계의 번호를 다시 정렬한다.

<표 2>와 <표 3>은 [그림 8]의 교차로에서 임의의 신호순서 목록 A-D-B-E-C-F에 대해서 각 신호방향의 진행신호 시작과 끝의 단계를 할당한 예를 보여준다.

<표 2> 각 방향별 신호단계 할당(Link-stage assignment)

	1	2	3	4	5	6
A	←→					
D		←→				
B	←→					
E				←→		
C					←→	
F						←→

<표 3> 신호순서 목록 A-D-B-E-C-F에 대해서 할당된 각 단계

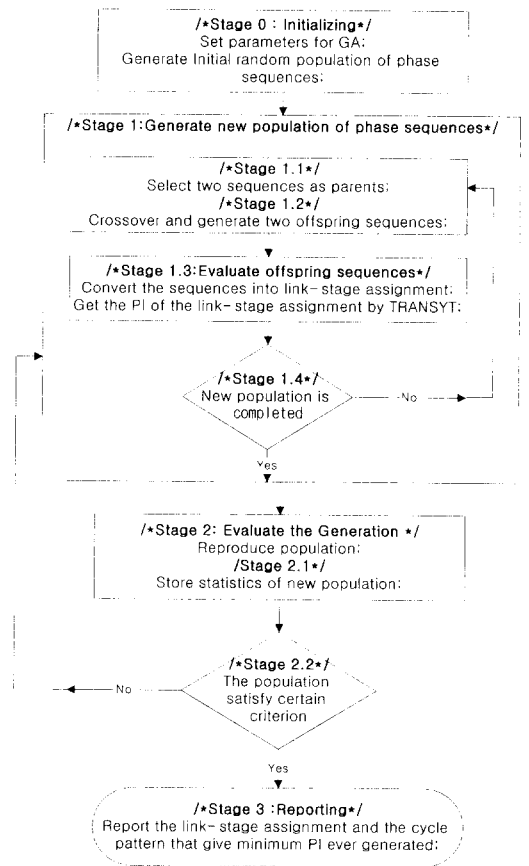
신호방향	시작 단계	끝 단계
A	1	2
B	1	4
C	5	6
D	2	4
E	4	5
F	6	1

4. 계산 과정

이와 같은 계산 모델에 따른 신호순서 결정 방

법을 프로그램으로 작성하였다. 프로그램의 계산 단계는 다음의 [그림 9]와 같다.

초기화 단계에서는 유전알고리즘을 수행하는데 필요한 모수(Parameter)들을 입력한다. 그 모수들은 세대의 개체수(Population size), 진화를 진행시킬 세대의 최대 수(Number of generation), 신호순서에 대한 교배와 돌연변이 확률(Junction crossover and mutation probability), 이중주기 패턴에 대한 교배와 돌연변이 확률(Double cycle pattern crossover and mutation probability), 선행비례척도의 승수(Scale multiplier), 최초의 난수(Random number seed) 등이다. 이러한 모수들을 입력한 뒤 임의의 초기 세대를 생성한다.



[그림 9] 신호순서 결정 과정

단계 1.1에서는 각 개체의 적합도를 그 개체가 선택될 확률의 비율로 삼아서 두 개의 부모 개체를 선택한다. 단계 1.2에서 신호순서 부분과 이중주기 패턴 부분 각각에 대해서 교배와 돌연변이 확률에 따라서 교배연산과 돌연변이 연산을 수행함으로써 두 개의 자손 개체를 생성한다. 단계 1.3에서는 새로 구해진 개체들이 나타내는 각 신호방향별 신호순서를 TRANSYT에서 사용할 수 있는 각 방향별 신호단계 할당으로 변환한다. 그리고 이중주기 패턴과 함께 입력으로 사용하여 TRANSYT으로부터 최소의 수행도 지수를 구한다. 단계 1.2~3의 과정을 주어진 세대의 개체수만큼 자손 개체들이 생성될 때까지 반복한다.

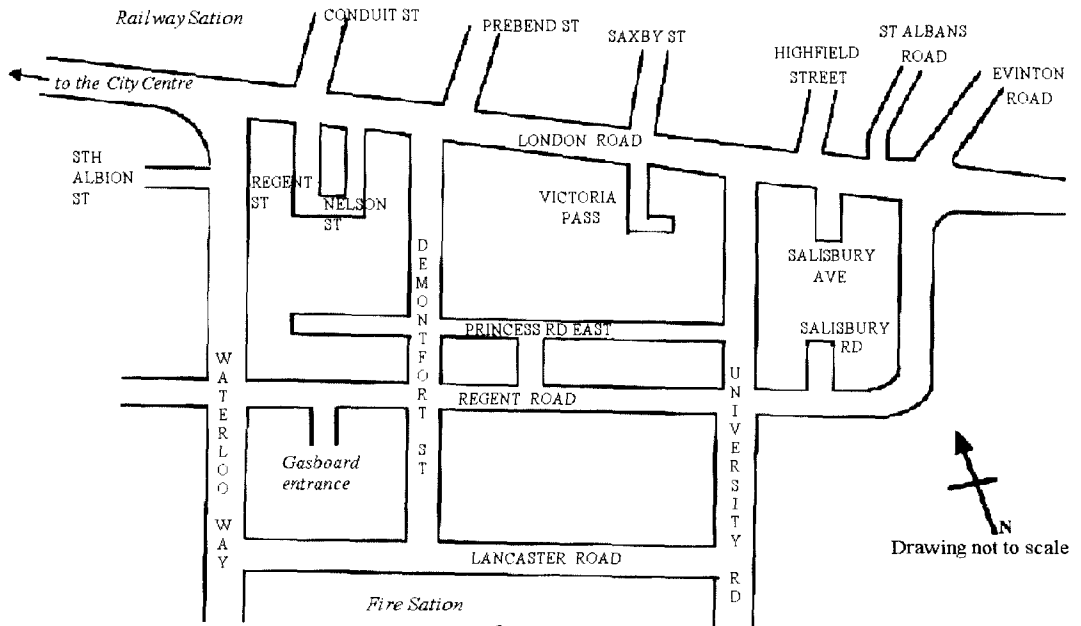
단계 2.1에서는 새로운 세대의 각 개체의 수행도 지수의 역수로서 구해진 적합도를 선형비례척도에 의해서 변환한다. 변환된 적합도는 단계 1.1에서 사용한다. 단계 2.2에서는 새로 구해진 세대가 일정한 판정기준을 만족하는지 살펴본다. 예를 들어, 수행도 지수의 개선된 정도가 8% 이하인 경우가 몇 세

대 이상 계속되었다던가, 그때까지 진행된 세대수가 미리 정해둔 세대의 최대 수를 넘었다던가 하는 기준들이다. 이러한 판정 기준들을 만족하면 진화 과정을 끝낸다. 단계 3에서는 그때까지 구해진 각 방향별 신호단계 할당과 이중주기 패턴들 가운데 수행도지수가 최소인 것을 골라 출력하고 끝낸다.

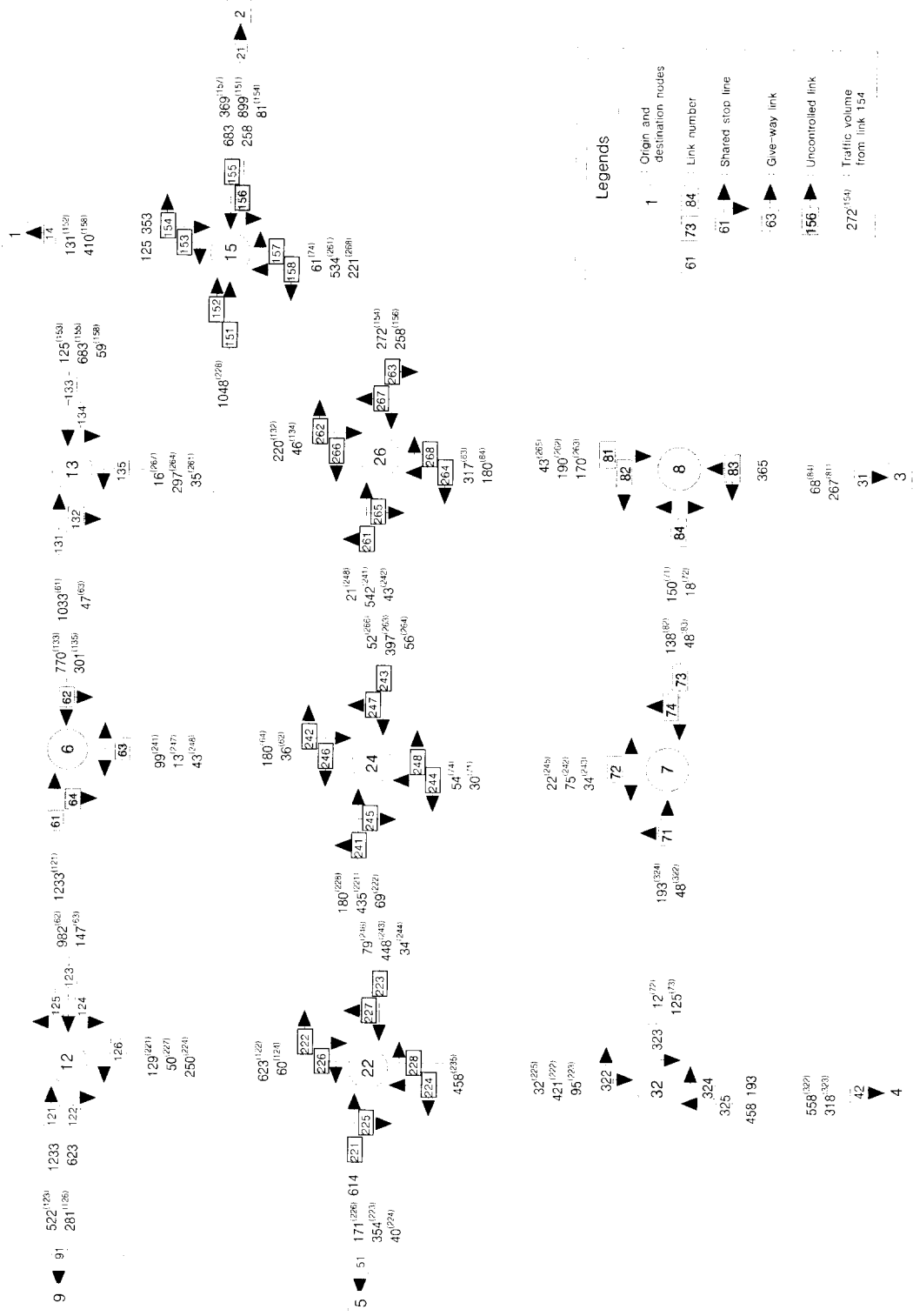
5. 실험 예

위와 같은 유전알고리즘을 FORTRAN으로 프로그래밍하고 그 효율성을 검토하기 위하여 영국의 Leicester의 일부 지역 교통망에서 수집된 교통량 자료를 사용하여 테스트하였다. [그림 10]은 교통량 자료 수집 대상 지역의 도로배치 현황을 보여준다. [그림 11]은 대상 지역의 도로 네트워크를 개념적으로 나타내었는데, 각 교차로에서 각 진행 방향별로 교통량을 표시하였다.

[그림 12]는 프로그램을 수행한 결과, 세대수의 증가에 따라 수행도지수가 수렴되는 것을 보여 준



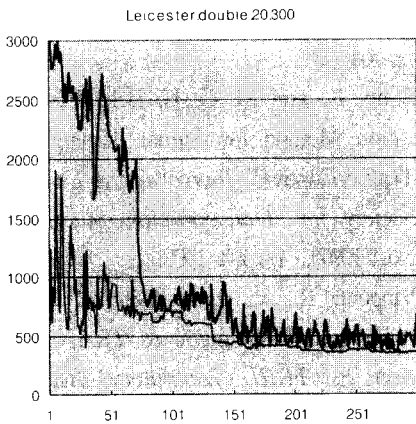
[그림 10] Leicester Region R의 도로 배치



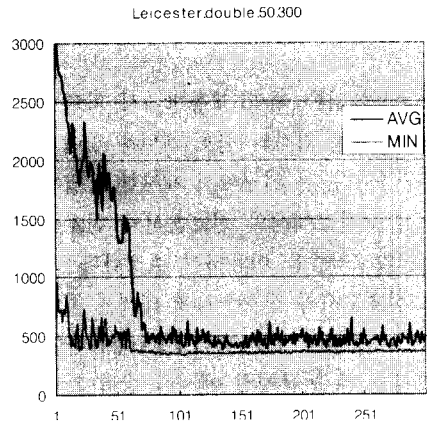
[그림 11] Leicester Region R의 도로네트

다. 그림에서 각 경우에 대해서 300세대까지 계산을 수행했다. (a)와 (c)의 경우에는 세대의 개체수를 20으로 하였고, (b)와 (d)의 경우에는 세대의 개체수를 50으로 하였다. 이때, 신호순서에 대한 교배와 돌연변이 확률은 각각 0.8과 0.2로, 이중주기 패턴에 대한 교배와 돌연변이 확률은 각각 0.9와 0.1, 선형비례척도의 승수는 1.5로 두었다. 한편 (a)와 (b)의 경우 각 교차로에서의 신호순서와 이중주기 패턴 두 가지 모두를 유전알고리즘에 의해서 최적화시키는 과정을 적용하였다. 반면에 (c)와 (d)의 경우에는 신호순서만 유전알고리즘에 의해서 최적화시키고, 이중주기 패턴은 TRANSYT에서 추천하는 것을 그대로 사용하였다.

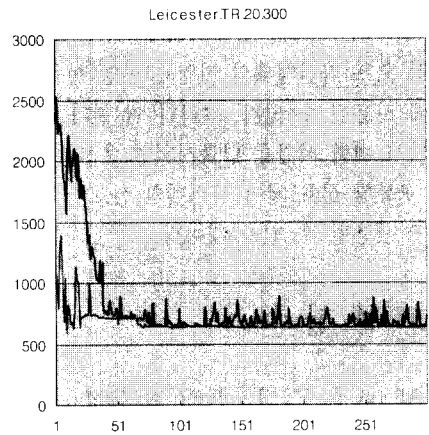
세대의 개체수를 많게 한 경우가 적게 한 경우보다 더 빨리 수렴되었으며, 300세대까지 실행한 후 최종적으로 구해진 수행도 지수도 더 좋은 결과를 얻었다. 한편 신호순서와 이중주기 패턴 두 가지 모두를 유전알고리즘에 의해서 최적화시키는 과정을 적용한 경우에 최종적으로 구해진 수행도 지수가 신호순서만 유전알고리즘에 의해서 최적화시킨 경우보다 낮았다. 즉, 신호순서와 이중주기 패턴 두 가지 모두를 유전알고리즘에 의해서 최적화시키는 과정을 적용한 경우에 더 좋은 결과를 얻을 수 있었다.



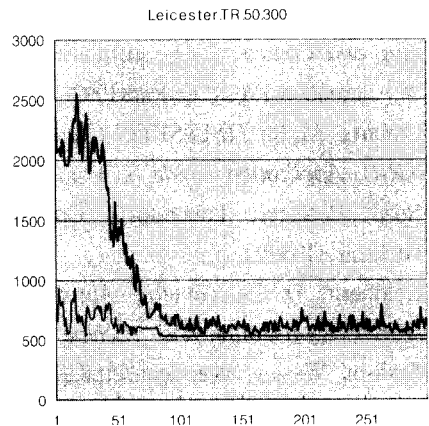
(a)



(b)



(c)



(d)

[그림 12] 세대수에 따른 수행도 지수의 수렴

6. 결 론

본 논문에서는 지역 교통망에서 차량흐름을 최적화하기 위한 최적의 신호 패턴을 찾아내는 방법을 연구하였다. 본 연구에서 제시한 방법에 따르면, 먼저 각 교차로에서의 신호순서와 신호주기 형태를 유전알고리즘을 이용하여 결정한다. 그와 동시에 그 신호순서와 신호주기에 대해서 최적인 신호 시간과 오프셋을 TRANSYT을 이용하여 계산한다. 그리고 본 논문에서 제시한 방법에 따라 프로그램을 작성하고 일부 지역에서 수집된 교통량 자료를 사용하여 효율성을 테스트하였다.

본 연구의 결과는 원래 TRANSYT 프로그램의 특성상 오프 라인형태의 지역 교통망 신호 제어 시스템에 사용될 수 있다. 그러나 이 결과를 좀더 발전시켜서 계산 시간을 단축시키면 온-라인, 감응식 지역 교통망 신호 제어 시스템의 신호주기 결정 모형에도 사용될 수 있을 것이다.

참 고 문 헌

- [1] Allsop, R. E., "Evolving application of mathematical optimization in design and operation of individual signal-controlled road junctions," *Mathematics in Transport Planning and Control*, J. D. Griffiths(Ed), pp.1-24, Clarendon Press, Oxford(1992).
- [2] Crabtree, M. R., *TRANSYT/9 Users Manual, TRRL Application Guide 8*, Transport and Road Research Laboratory, Crowthorne, Berkshire(1988).
- [3] Goldberg, D. E., *Genetic Algorithms in search, optimization and machine learning*, Addison-Wesley, Massachusetts(1989).
- [4] Heydecker, B. G., "Sequencing of traffic signals," *Mathematics in Transport Planning and Control*, J. D. Griffiths(Ed), pp.57-67, Clarendon Press, Oxford(1992).
- [5] Hunt, P. B., D. I. Robertson, R. D. Bretherton and R. I. Winton, *SCOOT - a traffic responsive method of co-ordinating signals, TRRL Laboratory Report 1014*, Transport and Road Research Laboratory, Crowthorne (1981).
- [6] Hadi, M. A. and C. E. Wallace, "Optimization of Signal Phasing and Timing Using Chauchy Simulated Annealing," *Presentation to The Transportation Research Board 3rd Annual Meeting*, Washington, D. C.(1994).
- [7] Silcock, J. P. and A. Sang, "SIGSIGN: A phase-based optimization program for individual signal-controlled junctions," *Traffic Engineering and Control*, Vol.31, No.5(1990), pp.291-298.
- [8] Simmonite, B. F.(1994) "LINSIGII: The next generation. A computer program to aid traffic signal design and assessment," *Traffic Engineering and Control*, Vol.35, No.4(1994), pp.236-242.
- [9] Sung, K., J. Anderson and M. G. H. Bell, "A Genetic Algorithm to Optimize Signal Phasing in Area Traffic Control," *IEE Conference Publication Number 446*, Second International Conference on GALESIA(1997), pp.404-408.
- [10] Sung, K. and M. G. H. Bell, "An Optimization Method for Signal Timing in Area Traffic Control," *Journal of the Eastern Asia Society for Transportation Studies*, Vol.2, No.4(1997), pp.993-1002.
- [11] Vincent, R. A., A. I. Mitchell and D. I. Robertson, *User guide to TRANSYT version 8, TRRL Laboratory Report 888*, Transport and Road Research Laboratory, Crowthorne, Berkshire(1980).