

# 병렬 유전자 알고리즘을 이용한 차량경로문제에 관한 연구

유용석 · 노인규

한양대학교 산업공학과

## Vehicle Routing Problem Using Parallel Genetic Algorithm

Yoong-Seok Yoo · In-Kyu Ro

Vehicle routing problem(VRP) is known to be NP-hard problem, and good heuristic algorithm needs to be developed. To develop a heuristic algorithm for the VRP, this study suggests a parallel genetic algorithm(PGA), which determines each vehicle route in order to minimize the transportation costs. The PGA developed in this study uses two dimensional array chromosomes, which rows represent each vehicle route. The PGA uses new genetic operators. New mutation operator is composed of internal and external operators. internal mutation swaps customer locations within a vehicle routing, and external mutation swaps customer locations between vehicles. Ten problems were solved using this algorithm and showed good results in a relatively short time.

### 1. 서론

#### 1.1 연구의 배경 및 목적

현대 산업사회에서 기업의 규모가 커질수록 점점 다양한 제품의 공급지와 수요지를 보유하게 된다. 또한 점점 복잡해지는 도로사정과 차량의 증가 등 외부 환경요인으로 인하여 원하는 시간에 제품을 수요지에 배송하는 것이 점점 힘들어지고 있다. 따라서 기업은 고객과의 계약을 지키기 위해서 물류부분에 많은 비용을 투자하고 있으며, 물류비용의 감소를 위해 노력하고 있다.

물류비용의 감소를 위해서는 적절한 배송정책을 수립하여 운영하는 것이 중요하다. 적절한 배송정책이란 전체 차량의 운행거리를 최소화하고, 운행차량의 수를 줄이며, 차량의 적재용량을 최대한으로 활용하는 것이다. 따라서 각 차량은 동일한 수요지점을 중복 방문하지 않으며, 가능한 한 하나의 차량이 많은 수요지점을 방문하도록 배송계획을 수립해야 한다.

차량경로문제(vehicle routing problem)는 다수의 수요지점에서 발생하는 수요를 만족시키는 차량의 경로를 결정하는 문제이다. 일반적으로 차량경로문제는 차량의 적재 용량과 대수, 운행거리의 제약하에서 최소의 비용으로 모든 수요를 만족할 수 있는 차량의 경로를 구하는 것이다.

Danzig and Ramser(1959)의 연구에 의하면 차량경로문제는 일반적으로 NP-hard 문제라고 알려져 있다. 왜냐하면 수요지점

의 수가 늘어날수록 계산량이 증가하기 때문이다. 만일  $n$ 개의 수요지점이 있는 경우에 가능한 차량의 경로의 수는  $n!/2$ 이다. 예를 들어 수요지점이 15개인 경우 가능한 차량의 배송 경로는 653,837,184,000 가지나 된다. 따라서 이러한 경우를 모두 탐색하여 최적해를 구하는 최적화 해법은 사용이 불가능하므로 발견적 해법을 이용하여 배송차량의 경로를 구하는 방법을 이용해야 한다.

발견적 해법은 최적해를 얻을 수 있다는 보장은 없지만, 보다 빠른 시간 안에 시스템이 만족할 수 있는 수준의 해를 얻을 수 있다는 데 그 장점이 있다. 그러나 발견적 해법 역시 수요지점이 늘어날수록 계산량이 증가하므로 탐색의 종료시점을 정하는 것이 어려운 것이 단점이다. 따라서 계산 시간에 따라 목적하는 해에 수렴해 나가는 정도가 중요한 문제로 인식되고 있다.

본 연구에서는 발견적 해법의 하나인 유전자 알고리즘(genetic algorithm)을 차량경로문제에 맞게 수정한 병렬 유전자 알고리즘(parallel genetic algorithm)을 이용하여 보다 빠른 시간 안에 목적하는 수준의 해를 구할 수 있는 발견적 해법을 제시한다.

병렬 유전자 알고리즘은 유전인자를 2차원 배열형태로 저장하는 유전자를 사용한다. 각각의 배송차량별로 하나의 독립된 유전자가 생성되며, 이러한 유전자들이 모여 완전한 하나의 유전자를 생성하는 것이다. 이는 보다 현실상황에 맞게 유전인자를 표현하는 것이다. 따라서 세대가 진화되어도 각각의 차량에 대한 정보를 계속 유지하며 진화를 할 수 있다.

병렬 유전자 알고리즘은 기존의 유전 연산자를 사용할 수 없다. 따라서 새로운 형태의 유전 연산자를 사용하는데, 각각의 차량별로 수행하는 교배(crossover) 연산자와 역전(reverse) 연산자가 있으며, 돌연변이(mutation) 연산자는 내부 돌연변이(internal mutation)와 외부 돌연변이(external mutation) 연산자로 나누어 수행하게 된다.

1.2 기존연구 고찰

차량경로문제에 대한 기존연구는 최적화 해법(optimal algorithm)과 발견적 해법(heuristic algorithm)으로 나눌 수 있다. 그러나 수요지점의 수가 늘어날 경우 계산시간이 급속히 증가하므로 최적화 해법은 적용하기 곤란한 경우가 많다. 따라서 요즘에는 상대적으로 발견적 해법에 대한 연구가 활발히 진행되고 있다.

Christofides and Eilon(1969)은 Lagrangian 완화기법에 의한 분지한계법(branch and bound method)을 개발하여 기본적인 차량 경로문제에 차량의 최대 운행거리 제한에 대한 제약조건을 추가시킨 문제를 다루었다. 분지한계법은 최적해를 구할 수 있는 장점은 있으나, 수요지점이 늘어날 경우 계산시간이 많이 걸리거나 불가능하다는 단점을 가지고 있다.

Clarke and Wright(1964)는 두 차량이 서로 다른 두 지점을 방문하고 돌아오는 것보다 한 대의 차량이 두 지점을 방문하고 돌아오는 경우 발생하는 비용의 절약을 이용하는 절약기법(saving heuristic)을 제안하였다.

Gendreau, Hertz and Laporte(1994)는 tabu search 개념을 이용한 해법을 개발하였다. 이 해법은 다른 방법들에 비하여 우수한 해를 구할 수 있는 장점은 있으나 계산시간이 오래 걸린다는 단점을 가지고 있다.

Goldberg and Lingle은 처음으로 유전자 알고리즘을 순환판매원문제(TSP)에 적용하였으며, 신해웅(1994)은 혼합형 유전해법(hybrid genetic algorithm)을 이용하여 일회방문 및 다회방문이 모두 가능한 수송모델을 개발하였다.

Lee and Kim(1995)은 병렬 유전자 알고리즘(parallel genetic algorithm)을 처음으로 사용하였다. 이들은 단일 기계작업 일정 계획 문제(job scheduling problem for a single machine)에 이 알고리즘을 적용하였다. 그러나 이진수(binary) 형태의 유전인자를 사용하였으며, 유전자는 1차원 배열로 표시되었다.

Cheng and Gen(1995)은 병렬 기계작업 일정계획 문제(job scheduling problem in identical parallel machine systems)에 병렬 유전자 알고리즘을 적용하였다. 이들은 실수(float) 유전인자를 사용하였으나, 유전자 표현에 있어서는 여전히 1차원 배열을 사용하였다. 따라서 각 기계들을 분할하기 위해서 특수한 기호를 사용하였다.

2.1 모형의 정의 및 가정

차량경로문제에 대한 전통적인 모형은 Dantzig and Ramser (1959)에 의해 제시되었다. 이 모형에서는 단일의 본점( $i=0$ )과  $N$ 개의 수요지점, 그리고 차량용량이  $U$ 로 동일한  $M$ 대의 차량을 보유하고 있다. 또한 각각의 수요지점의 수요량은  $D_i$ 로 알려져 있고, 차량의 운행거리에 대한 제한이 있을 수도 있다.

이러한 차량경로문제에서는 다음과 같은 가정을 사용한다.

- 가정 1 : 모든 차량은 본점을 출발하여 다시 본점으로 돌아온다.
- 가정 2 : 본점을 제외한 모든 수요지점은 한 대의 차량에 의해서 한번만 방문된다.
- 가정 3 : 모든 수요지점의 수요량은 차량의 적재용량을 넘지 않는다.
- 가정 4 : 각각의 차량의 경로에 포함된 수요지점의 수요량의 합은 차량의 적재용량을 초과할 수 없다.
- 가정 5 : 각각의 차량의 경로에 포함된 수요지점의 경로거리의 합은 차량의 최대 운행거리를 초과할 수 없다.
- 가정 6 : 모든 수요지점의 수요량은 알려져 있다.

2.2 기호의 정의 및 모형

- $N$ : 수요지점의 수
- $D_i$ : 수요지점  $i$ 의 수요량 ( $i = 1, 2, \dots, N$ )
- $M$ : 차량 수
- $U$ : 차량의 적재 용량
- $C_{ijk}$ : 차량  $k$ 의 수요지점  $i$ 에서 수요지점  $j$ 까지의 운행거리 ( $i = 0, 1, \dots, N; j = 0, 1, \dots, N; k = 1, 2, \dots, M; i, j$ 가 0인 경우는 본점을 의미)
- $W$ : 차량의 최대 운행거리

이를 수리 모형으로 표현하면 식 (1) ~ 식 (9)와 같다. 본 수리모형의 목적식은 식 (1)로서, 차량의 총 운행거리를 최소화하는 것이다. 이때 차량의 운행거리는 비용과 일 대 일로 대응할 수 있으므로 식 (1)은 차량의 총 운행비용을 최소화하는 목적식이 될 수 있다.

유효한 차량의 경로를 결정하기 위한 제약식으로는 식 (2) ~ 식 (9)가 사용된다. 식 (2)와 식 (3)은 각 수요지점에는 한 대의 차량만이 방문한다는 일회방문의 제약의 의미이다. 식 (4)는 차량이 수요지점을 방문한 후에는 반드시 다른 수요지점이나 본점으로 돌아가야 한다는 제약이다. 식 (5)는 모든 차량이 각각의 차량의 경로에 포함된 수요지점을 방문하는 데 소요되는 총 운행거리가 차량의 최대 운행거리를 초과할 수 없다는 제약이며, 식 (6)은 모든 차량이 각각의 차량의 경로에 포함된 수요지점의 총 수요는 차량의 적재용량을 초과할 수 없다는 제약이다. 식 (7)과 식 (8)은 모든 수요지점을 방문하는 데 필요한

2. 수리모형

차량의 수는 주어진 최대 차량의 수를 넘을 수 없다는 제약이며, 식 (9)는 0, 1 정수 조건이다.

$$\text{Minimize } \sum_{k=1}^M \sum_{i=0}^N \sum_{j=0}^N c_{ijk} x_{ijk} \quad (1)$$

subject to

$$\sum_{k=1}^M \sum_{i=0}^N x_{ijk} = 1 \quad \forall j \quad (2)$$

$$\sum_{k=1}^M \sum_{j=0}^N x_{ijk} = 1 \quad \forall i \quad (3)$$

$$\sum_{j=0}^N x_{ijk} - \sum_{i=0}^N x_{ijk} = 0 \quad \forall i, k \quad (4)$$

$$\sum_{i=0}^N \sum_{j=0}^N c_{ijk} x_{ijk} \leq W \quad \forall k \quad (5)$$

$$\sum_{i=1}^N D_i \left( \sum_{j=0}^N x_{ijk} \right) \leq U \quad \forall k \quad (6)$$

$$\sum_{j=1}^N x_{0jk} \leq M \quad \forall k \quad (7)$$

$$\sum_{i=1}^N x_{ik} \leq M \quad \forall k \quad (8)$$

$$x_{ijk} = \{0, 1\} \quad \forall i, j, k \quad (9)$$

### 3. 병렬 유전자 알고리즘

#### 3.1 병렬 유전자 알고리즘 개념

병렬 유전자 알고리즘(parallel genetic algorithm)은 Lee와 Kim(1995), Cheng과 Gen(1995) 등에 의해서 연구되었다. 지금까지의 병렬 유전자 알고리즘은 차량경로문제를 위한 것이 아니고, 기계들간의 작업 일정계획 문제(job scheduling problem)에 국한되어 사용되었다.

Lee와 Kim은 이진수 형태의 유전인자를 사용하였으며 유전자는 1차원 배열로 표현되었다. Cheng과 Gen은 실수형태의 유전인자를 사용하였으나 유전자의 표현에 있어서는 역시 1차원 배열을 사용하였다. 또한 특수한 기호를 사용하여 기계를 구분하는 방법을 사용하였다. 따라서 이들의 연구는 시스템의 모형은 병렬이면서도 유전자는 직렬형태로 구성하였다는 한계를 가지고 있다.

본 연구에서 제시하는 병렬 유전자 알고리즘은 기존의 방법과는 다르게 실수 유전인자를 사용하며, 실제 시스템의 모형과 유사하게 2차원 배열형태로 유전자를 구성한다.

유전인자를 실수의 배열로 표현하면, 실제상황을 보다 잘 표현할 수 있으며, 유전인자를 다시 실수변수로 변환하는 과정을 거치지 않으므로 계산시간을 단축할 수 있다. 또한 수요지점의 수가 많은 경우에도 기억공간을 줄일 수 있으므로 대형의 문제에서도 효율적으로 해를 구할 수 있는 장점이 있다.

또한 유전자를 2차원 배열로 구성하게 되면, 다수의 차량이

다수의 지점을 일회방문하는 모형을 보다 현실성 있게 표현할 수 있으며, 차량의 적재용량을 최대한 활용할 수 있으므로 운행차량의 수를 최소화할 수 있다. 또한 각각의 유전 연산자를 차량단위로 적용함으로써 보다 효율적으로 해를 탐색해 나아갈 수 있다.

#### 3.2 전통적 유전자 알고리즘의 문제점

Zbigniew Michalewicz(1996)에 의하면 일반적으로 유전자 알고리즘은 다음과 같은 다섯 가지 요소를 고려하여야 한다.

- 해가 될 가능성이 있는 것의 유전자적 표현방법
- 초기 개체집단을 생성하는 방법
- 적합도에 의해 해를 평가하는 평가함수
- 자손의 합성을 변화시키는 유전 연산자
- 사용하는 여러 가지 매개변수의 값

따라서 전통적인 방법의 유전자 알고리즘을 차량경로문제에 적용할 경우에는 다음과 같은 문제점들이 발생한다. 본 연구에서 제시하는 병렬 유전자 알고리즘은 이러한 문제점들을 해결할 수 있는 해법을 제시한다.

첫째, 전통적 유전자 알고리즘은 해집합의 표현에 있어서 이진(binary)배열을 사용하였다. 이진배열을 사용할 경우  $N$ 개의 수요지점과  $K$ 대의 차량을 가진 경우에 대한 해를 표현하기 위해서는  $N \times N \times K$  차원의 배열이 필요하다. 따라서 수요지점의 수가 증가하면 보다 많은 메모리와 계산시간을 필요로 하는 단점이 있다. 따라서 문제의 크기가 커지는 경우에는 계산이 불가능하다.

따라서 본 연구에서 제시하는 알고리즘에서는 실수형(float) 배열을 사용하여 유전자를 구성한다. 만일  $N$ 개의 수요지점과  $K$ 대의 차량을 가진 경우에는 최대  $N \times K$  차원의 배열로 표현이 가능하다. 일반적으로 수요지점의 수에 비하여 차량의 수가 상대적으로 매우 적으므로 전통적 알고리즘에 비하여 많은 메모리의 절약과 계산시간의 단축할 수 있는 장점을 가지고 있다.

둘째, 전통적 유전자 알고리즘은 초기 해집합을 생성함에 있어서 무작위(random) 추출법을 사용한다. 이 경우 수요지점이 중복되거나 누락된 해집합이 발생할 수 있다. 따라서 일회 방문 제약조건을 위배하는 실현불가능 해(infeasible solution)들이 초기 해집합에 포함되게 되며, 불필요한 계산을 수행하게 된다. 이는 알고리즘의 수행속도를 떨어뜨리는 결과를 가져오게 된다.

본 연구에서 제시하는 알고리즘에서는 초기 해집합의 생성 과정에서 해를 검토하여 수요지점이 중복되지 않도록 구성한다. 그러므로 실현 불가능한 해들이 진화 과정의 초기에 제거됨으로써 보다 신속하게 목적하는 수준의 해로 수렴해 나갈 수 있는 장점을 가지고 있다.

셋째, 전통적 유전자 알고리즘은 이진배열형태의 유전자를 사용하므로, 해의 적합도(fitness) 평가를 위해서는 다시 실수로

변환하는 계산을 수행하여야 한다. 이 과정에서 실수값의 유효숫자 등에 의한 오차가 발생하고, 목적함수가 복잡한 경우 많은 계산시간이 소요되는 단점이 있다.

본 연구에서는 실수형 유전인자를 사용함으로써 해의 적합도 계산과정에서 발생할 수 있는 오차를 감소시키고, 계산시간을 단축할 수 있는 장점을 가지고 있다.

넷째, 전통적 알고리즘에서 사용하는 교배(crossover) 연산자와 돌연변이(mutation) 연산자, 역전(reverse) 연산자 등을 그대로 사용할 경우, 수요지점이 중복되거나 누락되어 일회방문 제약 조건을 위배하는 문제점을 가져온다. 그 동안 많은 연구에서 이러한 문제점을 해결하기 위해서 PMX, CX, OX(Cheng and Gen, 1996) 등의 연산자들이 개발되었다. 그러나 이러한 연산자들은 모두 1차원 배열형태의 유전자를 위한 것이다.

본 연구에서 사용되는 유전자는 2차원 배열형태로 구성되므로 기존의 연구에서 제시되는 유전 연산자를 그대로 사용할 수 없다. 따라서 2차원 배열의 형태로 구성된 유전자에 알맞은 새로운 유전 연산자를 개발하여 사용한다. 새로운 유전 연산자에 대하여는 다음절에서 보다 자세하게 설명하도록 한다.

다섯째, 유전자 알고리즘에서는 사용하는 매개변수의 값에 따라 해에 수렴하는 시간이나 정도에 많은 차이가 많이 발생한다. 따라서 많은 수준의 실험을 반복하여 최적의 매개변수를 결정하는 것이 중요하다. 그러나 모든 문제에 적용할 수 있는 매개변수를 구하는 것은 사실상 불가능하다. 따라서 본 연구에서도 다양한 수준실험을 통하여 본 알고리즘에 적합한 매개변수를 선정한다.

이상과 같이 기존의 전통적 유전자 알고리즘을 차량경로문제에 적용할 경우 발생할 수 있는 문제점들에 대하여 알아보았다. 본 연구에서 제시하는 병렬 유전자 알고리즘은 이러한 문제점들을 해결할 수 있는 효과적인 방법을 제시한다.

### 3.3 알고리즘의 개발

#### 3.3.1 순서도

병렬 유전자 알고리즘에 대한 개략적인 순서도는 <그림 1>과 같다. 기존의 유전자 알고리즘과의 차이점은 돌연변이 연산자를 내부와 외부로 분할하였으며, 해의 진화과정에서 발생할 수 있는 실현불가능 해의 제거를 위해서 해의 보정과정을 거치게 된다.

유전 연산자의 사용 후에 해의 보정과정을 수행함으로써 계산시간이 더 걸리게 되는 단점을 가지고 있으나, 다음 진화 단계에서 실현불가능 해를 제거할 수 있으므로 보다 빠른 해의 수렴속도를 가져올 수 있는 장점을 가지고 있다.

#### 3.3.2 유전자 구성

본 연구에서는 2차원 배열형태로 유전자를 구성한다. 유전자의 행은 각각의 차량을 의미하며, 열은 각 차량이 방문해야 할 수요지점의 고유번호가 된다. 또한 모든 차량은 본점(0)을

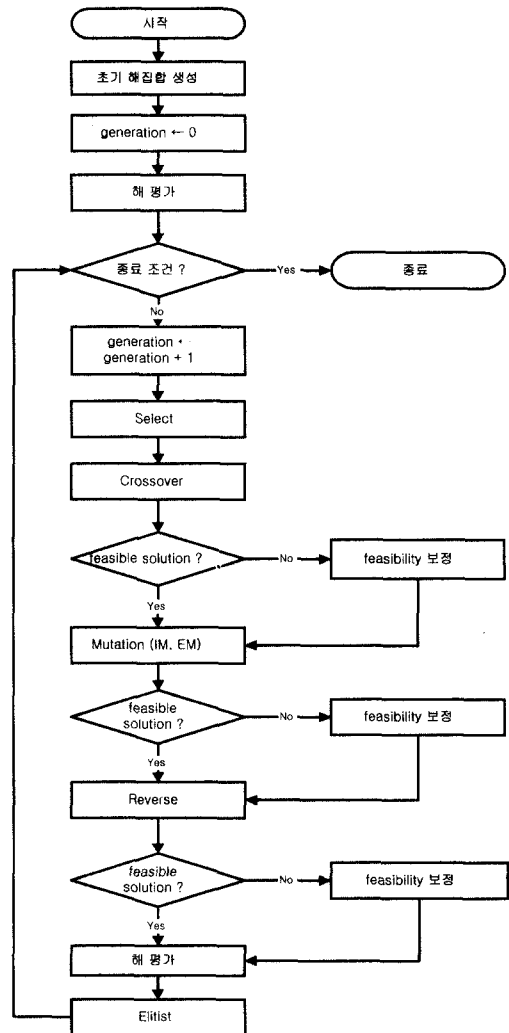


그림 1. 병렬 유전자 알고리즘의 순서도.

출발하여 다시 본점으로 돌아오는데, 이때 본점(0)은 표시하지 않는다.

다음 <그림 2>는 2차원 배열로 구성된 유전자의 예이다. 즉, 이 유전자는 3대의 차량을 보유하고 있으며, 1개의 본점과 12개의 수요지점을 가지고 있다. 예를 들면, 1번 차량은 0(본점) → 1 → 4 → 7 → 10 → 0(본점)의 배송경로를 가지게 되는 것이다.

차량의 경로는 차량의 최대 적재용량과 최대 운행거리 제약을 만족시켜야 하므로, 각 차량의 유전자의 길이는 서로 상이하다. 따라서 10로 표시된 유전인자는 차량의 적재용량이나 운행거리의 제약으로 인하여 더 이상 수요지점이 할당될 수 없다는 것을 의미한다.

1	4	7	10	
2	5	8	11	12
3	6	9		

그림 2. 2차원 배열의 유전자 예.

3.3.3 초기해 구성

초기 해집합은 수요지점들이 중복되지 않도록 구성한다. 먼저 각 수요지점이 중복되지 않도록 난수(random number)를 발생시켜서 1차원 배열에 저장한다. 다음으로 차량의 적재용량 제약이나 운행거리 제약에 위배되지 않도록 2차원 배열로 재구성한다.

2차원 배열로 변형하는 과정에서 각 차량의 적재용량과 운행거리 제약을 최대한 활용하도록 구성한다. 즉, 첫번째 차량의 적재용량과 운행거리 제약을 최대한 만족시킨 후, 다음 차량에 수요지점이 할당되게 된다. 따라서 최소의 차량으로 모든 수요지점의 수요량을 만족시킬 수 있는 유전자를 구성하게 된다.

이러한 방법으로 생성된 초기 해집합은 일회방문의 제약을 위배하지 않으며, 실현불가능한 해들이 초기 해집합에 포함되는 것을 방지할 수 있다. 따라서 불필요한 계산을 줄일 수 있으며, 해의 초기 탐색과정에서 보다 빠르게 목적하는 수준의 해에 수렴해 나아갈 수 있게 된다. 또한 운행차량의 수를 최소화 시킴으로써 잠재적인 비용의 절감을 가져올 수 있다.

3.3.4 적합도 평가

해의 적합도(fitness) 평가는 식 (1)처럼 나타낼 수 있다. 즉, 모든 수요지점의 수요를 만족하는 데 필요한 운행거리를 최소화하는 것이다. 다시 말하면 차량의 운행에 필요한 비용을 최소화하는 것이다.

3.3.5 교배 연산자

교배(crossover) 연산자는 두 부모해의 임의의 위치에서 유전자를 상호교환하여 새로운 자식 유전자를 생성해 나가는 과정이다. 이는 부모해보다 우수한 자식해를 만들어 낼 수 있다는 장점을 가지고 있다.

기존의 연구에서 차량경로문제에 적용가능한 교배 연산자로는 PMX, OX, CX(Cheng and Gen, 1996) 등의 연산자가 있다. 그러나 이들 연산자들은 모두 1차원 배열형태의 유전자에만 적용가능하다. 따라서 본 알고리즘에서는 2차원 배열형태의 유전자에 적용가능하도록 변형된 OX 연산자를 제시한다.

변형된 OX 연산자의 수행과정은 다음과 같으며, 간략히 도시하면 <그림 3>과 같다.

Step 1 : 교차율(Pc)에 의하여 두 부모해를 선발한 후, 각 부모해로부터 랜덤하게 하나씩의 차량(point 1, point 2)을 선택한다.

Step 2 : 첫번째 부모(parent 1)의 선택된 차량의 수요지점을 두 번째 부모(parent 2)의 유전자에서 제거한다.

Step 3 : 첫번째 부모의 선택된 차량을 제외한 다른 차량에 두 번째 부모의 남아 있는 수요지점을 차례대로 할당한다.

위의 방법으로 수행하면 <그림 3>의 child 1과 child 2가 생성되게 된다. 새로이 생성된 child 1과 child 2에서는 수요지점의 중복이나 누락이 발생하지 않는다.

parent 1

point 1 →

1	2	3	4	5
6	7	8	×	×
9	10	11	12	×

parent 2

point 2 →

1	4	7	10	×
2	5	8	11	12
3	6	9	×	×

child 1

1	4	10	2	5
6	7	8	×	×
11	12	3	9	×

child 2

1	4	7	10	×
2	3	5	6	8
9	11	12	×	×

그림 3. 변형된 OX 연산자.

본 알고리즘에서 사용한 변형된 OX 교배 연산자는 차량 단위로 교배 연산을 수행하게 된다. 따라서 일회방문의 제약을 위배하지 않으면서 선택된 차량의 경로가 다음 자식해에도 변형되지 않고 전달된다는 장점을 가지고 있다. 즉, 부모해의 우수한 형질을 다음 세대에 전달할 수 있는 것이다.

3.3.6 돌연변이 연산자

본 알고리즘에서는 상호교환 돌연변이(reciprocal exchange mutation)(Cheng and Gen, 1996)를 변형한 두 가지의 돌연변이(mutation) 연산자를 사용한다. 돌연변이 연산자는 기존의 부모해와 전혀 다른 해를 생성해냄으로써 유전 알고리즘이 국부 최적해(local optimum)에 빠지는 것을 방지하는 목적으로 사용된다.

본 연구에서 사용하는 돌연변이 연산자는 내부 돌연변이(internal mutation)와 외부 돌연변이(external mutation) 연산자로 나누어 수행한다. 유전자의 형태가 2차원 배열이므로, 하나의 행에서만 일어나는 연산을 내부 돌연변이라 부르고, 서로 다른 행 사이에서 일어나는 연산을 외부 돌연변이라 부른다. 다시 말하면 내부 돌연변이는 하나의 차량에서 발생하는 돌연변이이고, 외부 돌연변이는 서로 다른 차량 사이에서 발생하는 돌연변이이다.

내부 돌연변이의 과정을 살펴보면,

Step 1 : 돌연변이율(Pm)에 의하여 부모해(original)를 선발한 후, 랜덤하게 하나의 차량(point 1)을 선택한다.

Step 2 : 선택된 차량에서 2개의 수요지점(point 2, point 3)을 선택한다.

Step 3 : 선택된 두 수요지점을 서로 상호교환한다.

내부 돌연변이를 간략히 도시하면 <그림 4>와 같다.

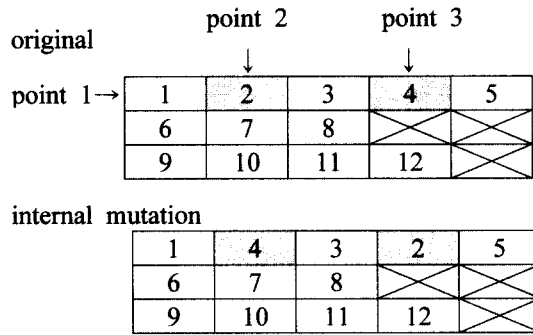


그림 4. 내부 돌연변이.

다음으로 외부 돌연변이의 과정을 살펴보면,

- Step 1 : 돌연변이율(Pm)에 의하여 부모해(original)를 선발한 후, 랜덤하게 두 대의 차량(point 1, point 2)을 선택한다.
  - Step 2 : 선택된 차량마다 각각 1개씩의 수요지점(point 3, point 4)을 선택한다.
  - Step 3 : 선택된 두 수요지점을 서로 상호교환한다.
- 외부 돌연변이를 간략히 도시하면 <그림 5>와 같다.

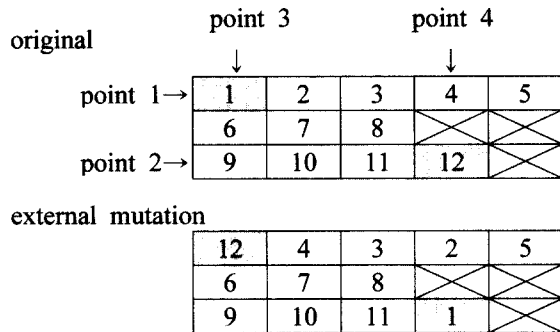


그림 5. 외부 돌연변이.

이상에서 살펴본 바와 같이 돌연변이 연산자를 두 가지로 나누어서 수행함으로써 국부 최적해(local optimum)에 빠지는 것을 최대한 방지할 수 있다. 내부 돌연변이는 교배(crossover) 연산자가 특정 차량의 경로를 다음 세대에 전달하는 과정에서 국부 최적해에 빠질 수 있는 가능성을 방지할 수 있으며, 외부 돌연변이는 유전자에 좀더 과감한 변화를 가할 수 있는 장점을 가지고 있다.

### 3.3.7 역전 연산자

본 알고리즘에서는 국부 최적해(local optimum)에서 좀더 수월하게 빠져 나오기 위해 역전(reverse) 연산자를 사용한다. 역전 연산자는 유전자 알고리즘이 목적하는 수준의 해에 대한 수렴속도가 세대가 진화할수록 느려지는 단점을 극복하기 위한 방법이다.

역전 연산자의 과정을 살펴보면,

- Step 1 : 역전율(Pr)에 의하여 부모해(original)를 선발한 후, 랜덤하게 한 대의 차량(point 1)을 선택한다.
  - Step 2 : 선택된 차량에서 1 개의 수요지점(point 2)을 선택한다.
  - Step 3 : 선택된 수요지점을 중심으로 전후의 수요지점을 교환한다.
- 역전 연산자를 간략히 도시하면 <그림 6>과 같다.

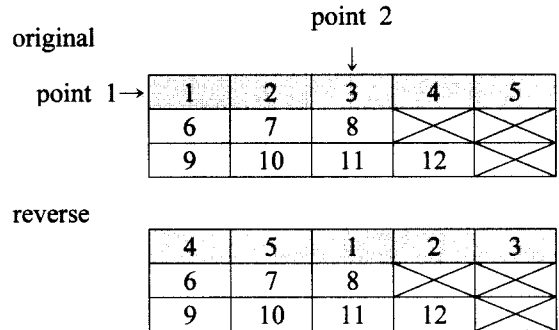


그림 6. 역전 연산자.

### 3.3.8 해의 보정

본 알고리즘에서 사용하는 유전 연산자를 사용할 경우에는 수요지점의 중복은 발생하지 않지만, 차량의 용량이나 운행거리 제약을 위반하는 경우가 발생할 수 있다. 따라서 각각의 연산자를 수행한 후 해의 보정과정을 거치게 된다.

해의 보정과정은 각 연산자의 수행이 끝난 후, 주어진 제약 조건을 검토하여 위반하는 해의 경우에는 초기해 생성방법과 동일한 방법으로 새로운 해를 생성하여 대체하는 방법을 사용한다.

## 4. 실험 및 분석

본 연구에서 제안하는 병렬 유전자 알고리즘을 평가하기 위해서 신(1994)의 연구에서 사용한 10개의 문제를 사용하였다. 이 문제들은 여러 명의 연구자들에 의해서 제안된 것인데 문제 1은 Hayes가, 문제 2는 Dantzig and Ramse, 문제 3, 4, 5, 7은 Gaskell, 문제 6은 Clarke and Wright, 문제 8, 9, 10 Christofides and Eilon이 최초로 제안하였다. 그 동안 이 문제들은 많은 연구자들에 의해서 우수한 해를 찾기 위한 다양한 알고리즘이 제시되어 왔다.

본 실험에서는 동일한 문제들에 대하여 각각 전통적 유전 연산자를 사용하는 알고리즘과 본 연구에서 제시하는 병렬 유전 연산자를 사용하는 알고리즘에 적용하여 본 알고리즘이 보다 우수함을 보여 준다.

알고리즘은 Visual C++ 5.0을 사용하여 구현하였으며, 실험은 Pentium 120MHz CPU, RAM 32MB를 장착하고 OS로 Windows95를 사용하는 PC에서 수행하였다.

표 1. 기준문제 특성

문제 번호	수요 지점수	소요 차량수	수요량 범위	차량 용량	운행시간 제한	하역시간
1	6	2	1	1	-	-
2	12	4	1100 ~ 1900	6000	-	-
3	21	4	100 ~ 2500	6000	200	10
4	22	5 ~ 6	60 ~ 4100	4500	240	10
5	29	4 ~ 5	100 ~ 3100	4500	240	10
6	30	8 ~ 9	1 ~ 123	140	-	-
7	32	4 ~ 5	40 ~ 4000	8000	240	10
8	50	5 ~ 6	3 ~ 41	160	-	-
9	75	10	1 ~ 37	140	-	-
10	100	8	1 ~ 41	200	-	-

※ 여기에서 사용된 소요 차량수는 기존연구(Elion, 1971)의 결과를 인용하였음.

#### 4.1 기준문제

실험에 사용한 기준문제는 <표 1>과 같은 특성을 가지고 있다. 이 문제들은 두 가지로 분류할 수 있는데, 문제 1, 2, 6, 8, 9, 10은 차량용량에만 제약을 가지고 있으며, 나머지 문제 3, 4, 5, 7은 차량용량과 운행시간, 하역시간의 제약을 모두 가지고 있는 경우이다.

기준문제에서 수요지점의 수는 최소 6개에서 최대 100개까지 다양하게 존재하고 있으며, 소요차량의 수도 다양하다. 따라서 다양한 환경의 문제에 대하여 본 알고리즘의 수행도를 측정할 수 있다. <표 1>에 나타난 소요차량수는 기존의 연구(Elion, 1971)에서 우수한 해로 밝혀진 경우에 사용된 차량수를 제시하였다.

#### 4.2 매개변수 최적화

유전자 알고리즘에서는 사용하는 매개변수값이 해의 수렴 속도에 많은 영향을 미친다. 또한 모든 유전자 알고리즘에 공통적으로 적용될 수 있는 매개변수 값은 존재하지 않는다. 따라서 본 알고리즘에서도 새로운 유전 연산자를 사용하므로 이에 맞는 매개변수값의 결정이 필요하다. 본 알고리즘에서 사용하는 매개변수로는 초기 해집합의 크기, 교배율, 내부 돌연변이율, 외부 돌연변이율, 역전율 등 다섯 가지의 매개변수를 사용한다. 초기 해집합의 크기는 진화과정 동안 보유할 해집합의 크기로서, 큰 경우에는 다양한 개체를 탐색할 수 있다는 장점이 있으나 계산시간이 많이 소요된다는 단점이 있으며, 작은 경우에는 그 반대의 장단점을 가지고 있다. 따라서 두 가지 요소를 모두 만족시키는 적정 수준의 해집합의 크기를 유지하여야 한다. 교배율, 내부 돌연변이율, 외부 돌연변이율, 역전율은 각각의 개체에 대하여 교배 연산자, 내부 돌연변이 연산자, 외부 돌연변이 연산자, 역전 연산자를 수행할 확률이다.

우수한 해를 찾기 위해서는 최적의 진화환경을 구성하여야 한다. 그러기 위해서는 다양한 수준의 매개변수들을 선택한 후 이들을 조합하고 실험을 수행하여 가장 우수한 결과를 가져오는 조합의 수준을 선택하면 된다. 그러나 수준의 수를 높일 경우 실험의 횟수가 증가하게 됨으로써 시간이 많이 소요된다는 단점을 가지고 있다. 만일  $n$ 개의 수준에 대하여  $k$ 회 반복 실험을 할 경우 본 알고리즘에서는 다섯 가지의 매개변수를 사용하므로 문제마다  $k \cdot (n^5)$ 회 실험을 해야 한다. 따라서 모든 실험을 전부 수행할 수는 없으므로, 본 연구에서는 하나의 문제에 대하여 실험을 수행하여 두 가지의 예비수준을 선택하고, 이 두 가지의 예비수준을 모든 문제에 적용하여 최적수준을 결정하는 과정을 거치도록 하였다.

본 연구에서는 먼저 수요지점의 수가 22개인 문제 4를 가지고 실험을 하였다. 세대 반복수를 5,000회로 고정시키고 한 번에 한 가지씩의 매개변수만을 변경하면서 수렴속도와 수행시간을 측정하였다. 초기 해집합의 크기는 100부터 400까지 50씩 변경을 시키고, 교배율은 0.7부터 1.0까지 0.1 단위로, 내부 돌연변이율과 외부 돌연변이율은 0.01부터 0.2까지 0.05단위로, 역전율은 0.3부터 0.6까지 0.1단위로 변경을 시키면서 측정하였다.

문제 4에 사용한 매개변수 중 우수한 수렴속도를 보이는 두 가지의 매개변수<표 2>를 이용하여 모든 문제에 2회씩 반복 실험하여 가장 우수한 수렴속도를 보이는 수준을 최적 진화환경으로 결정하였다.

본 실험에서 최적 환경으로 결정된 매개변수는 <표 2>와 같다.

#### 4.3 실험결과 분석

앞에서 구한 최적 진화환경요인으로 선정된 매개변수들을 이용하여 기존의 유전자 알고리즘과 본 연구에서 제안하는 병렬

표 2. 매개변수 최적화 실험결과

수준	초기해	교배율	내부 돌연변이	외부 돌연변이	역전을
1	150	0.8*	0.05	0.05	0.3*
2	300*	0.9	0.1*	0.1*	0.5

\* 최적 진화환경요인으로 선택된 매개변수

유전자 알고리즘을 10개의 기준문제에 대하여 비교 실험하였다. 세대 반복수는 5,000 세대로 제한하고, 동일한 해가 2,000회 이상 연속하여 나오게 되면 더 이상 해의 진화가 없는 것으로 판단하고 실험을 종료한다.

4.3.1 계산시간 분석

10개의 문제의 계산시간을 살펴보면 <표 3>과 같다. 여기에서는 두 가지 방법을 사용하여 계산시간을 비교하였다. 하나는 두 알고리즘의 수렴속도를 비교하는 계산시간이고, 다른 하나는 알고리즘의 전체 수행속도를 비교하는 계산시간이다. 이와 같이 두 가지로 나눈 이유는, 두 알고리즘이 종료조건에 도달한 후에 구한 해가 차이가 나기 때문에 동일한 조건에서 분석을 하기 위해서이다.

첫째, 계산시간 A는 두 알고리즘의 수렴 속도를 비교하기 위한 것이다. 계산시간 A는 두 알고리즘이 종료조건까지 해를 탐색한 후, 열등한 해를 보이는 알고리즘이 구한 해를 보다 우수한 해를 보여 주는 알고리즘에서 구할 때 걸리는 시간이다. 예를 들면 열등한 알고리즘이 5,000 세대 진화과정에서 가장 우수한 해가 4,000 세대에서 나타나기 시작한다면 그때까지의 계산시간과, 이와 동일한 수준의 해를 우수한 알고리즘이 얻을 때까지 걸리는 계산시간을 보여 주는 것이다.

<표 3>에서 보는 바와 같이, 문제 2와 8을 제외한 8개의 문

제에서 병렬 유전자 알고리즘이 보다 빠르게 해를 구해낼 수 있다. 문제 1에서 두 알고리즘 모두 계산시간이 0이 나온 이유는 수요지점이 적으므로 인해서 초기 해집합에 최적해가 포함되기 때문이다. 특히 수요지점이 많은 문제 9와 10의 경우에 있어서도 병렬 유전자 알고리즘이 각각 약 24분과 29분 정도 해를 빨리 구하고 있음을 알 수 있다. 따라서 본 연구에서 제안하는 병렬 유전자 알고리즘이 대형의 문제에서 보다 우수하게 사용될 수 있다는 것을 보여 주고 있다.

둘째, 계산시간 B는 두 알고리즘이 모두 5,000 세대를 반복하는 경우 걸리는 시간이다. 이 경우에는 모든 문제에 걸쳐서 병렬 유전자 알고리즘이 열등한 것으로 보여지고 있다. 그러나 두 알고리즘이 구한 해의 정도에 차이가 있으므로 이를 단순히 수치적으로 비교할 수는 없다. 이는 단지 두 알고리즘을 5,000세대 정도 수행하였을 경우에 걸리는 계산시간을 보여 주기 위함이다. <표 3>에서 보는 바와 같이, 수요지점이 많은 문제 9와 10의 경우 계산시간이 약 38분과 53분이 걸리게 된다. 본 연구에서 사용하는 알고리즘이 동일한 세대조건에서 계산시간이 더 걸리는 것은 각각의 유전 연산자를 사용한 후 실현 불가능 해를 제거하는 해의 보정과정을 거치기 때문이다.

4.3.2 진화 과정 분석

기존 유전자 알고리즘의 결과와 본 연구의 결과를 비교하여 본 알고리즘의 적합도가 우수한 진화세대 구간을 보면 <표 4>와 같다.

<표 4>에서 보는 바와 같이 문제 2의 경우에는 본 알고리즘이 초기에는 조금 빠른 수렴속도를 보이지만 456세대 이후부터 1402세대까지는 기존의 알고리즘이 보다 우수한 적합도를 보여 주고 있다. 1402세대 이후부터는 두 알고리즘이 모두 동일한 적합도를 보여 주고 있다(<그림 7>). 문제 8의 경우에는 부분적으로 적합도가 좋은 구간이 있기는 하지만, 3639세대 이

표 3. 두 알고리즘의 계산시간 비교

문제 번호	수요 지점	계산 시간 A <sup>(1)</sup> (초)			계산 시간 B <sup>(2)</sup> (초)		
		기존 GA	병렬 GA	차 이	기존 GA	병렬 GA	차 이
1	6	0*	0*	0	183	184	-1
2	12	33*	38	-5	315	433	-118
3	21	659	76*	583	977	1122	-145
4	22	910	239*	671	1007	1077	-70
5	29	1154	178*	976	1232	1492	-260
6	30	598	364*	234	1150	1436	-286
7	32	672	430*	242	1220	1581	-361
8	50	1302*	2043	-1011	1936	2320	-384
9	75	2150	689*	1461	2506	3220	-714
10	100	2768	983*	1785	3419	4567	-1148

(1) 우수한 해가 나타나기 시작한 시간

(2) 5,000 세대 진화 후 알고리즘이 종료된 시간

\* 우수한 해를 구하는 알고리즘



표 4. 본 알고리즘이 우수한 진화세대 구간

문제 번호	수요 지점	소요 차량수		본 알고리즘이 동일하거나 우수한 진화세대 구간
		기존 GA	병렬 GA	
1	6	2	2	ALL*
2	12	4	4	29-455, 1402 -
3	21	5	4	12 -**
4	22	6	6	ALL
5	29	5	5	ALL
6	30	7	7	11 -
7	32	5	5	1057 -
8	50	5	5	11-18, 35-100, 115-698, 1812-3638
9	75	11	10	89 -
10	100	8	8	55 -

ALL\* : 전 진화세대 구간에서 우수한 적합도를 보임  
 XX-\*\* : XX 진화세대 이후부터 적합도가 우수

후에서는 계속해서 기존의 알고리즘의 적합도가 더 우수하게 나왔다(<그림 8>).

그러나 문제 1, 4, 5에서는 진화세대 전 구간에 걸쳐서 본 알고리즘의 적합도가 우수하거나 동일하게 나왔으며(<그림 9>), 문제 3, 6, 9, 10의 경우에는 초기 100세대 이전부터 우수한 적합도를 보임으로써 해의 수렴속도가 우수함을 알 수 있다(<그림 10>).

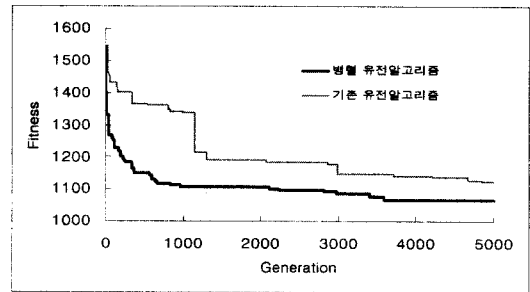


그림 9. 지점수 29인 경우(문제 5).

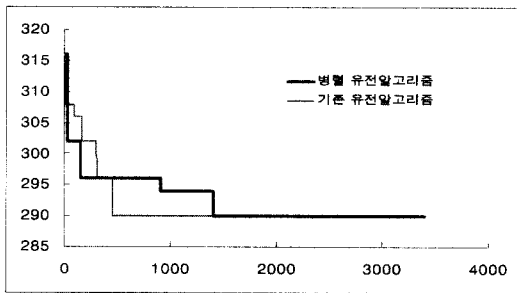


그림 7. 지점수 12인 경우(문제 2).

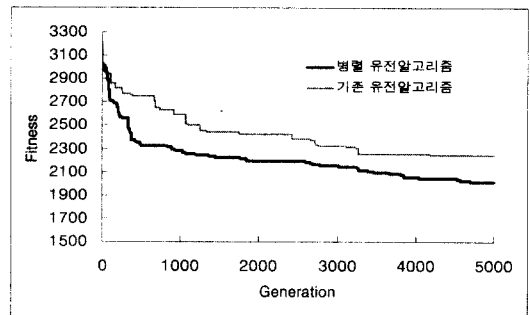


그림 10. 지점수 100인 경우(문제 10).

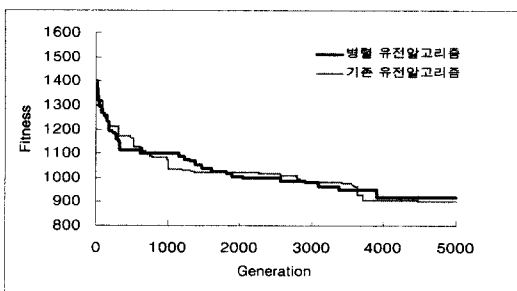


그림 8. 지점수 50인 경우(문제 8).

특히 수요지점의 수가 많은 문제 9, 10의 경우에서도 우수한 수렴속도를 얻을 수 있으므로 본 알고리즘을 대형의 문제에 적용할 경우에는 보다 빠른 시간 안에 만족하는 수준의 해를 얻을 수 있다.

본 알고리즘과 기존의 알고리즘을 비교 실험한 결과, 전체 10개의 문제 중에서 8개의 문제에서 기존의 알고리즘보다 우

수한 수렴속도를 보여 주고 있으며, 2개의 문제에서는 기존의 알고리즘과 유사하거나 동일한 수렴속도를 보여 주고 있다. 따라서 본 알고리즘은 보다 빠른 시간 안에 시스템에서 만족하는 수준의 해를 얻기에 유용하다고 할 수 있다.

### 5. 결론

본 연구에서는 병렬 유전자 알고리즘을 이용하여 차량경로문제에 대한 발견적 해법을 제시하였다. 발견적 해법은 좋은 해를 얻기 위해서는 많은 계산시간이 필요하다는 단점이 있기 때문에, 시스템이 만족하는 해의 수준에서 탐색을 마쳐야 한다. 따라서 보다 빠르게 만족하는 수준의 해에 도달하는 것이 발견적 해법에서 고려해야 할 중요한 요소이다.

본 연구에서 제시하는 병렬 유전자 알고리즘은 이러한 탐색적 해법의 중요요소인 해에 대한 수렴속도 부분을 개선한 것으로, 기존의 유전자 알고리즘보다 신속하게 보다 우수한 해로 수렴해 나간다. 특히 수요지점이 75개 이상인 대형의 문제에서 효율적으로 적용될 수 있다.

본 연구에서 사용한 병렬 유전자 알고리즘은 실수 유전인자를 사용하며, 유전자를 배송차량별로 나누어서 2차원 배열형태로 구성한다. 유전자에서 행은 수요지점을 방문하는 배송차량을 의미하고, 열은 각각의 차량이 방문해야 할 수요지점을 의미한다. 따라서 기존의 유전자 알고리즘보다 실제 현실 상황을 보다 잘 표현할 수 있으며, 기억장소의 절약과 수행시간의 단축을 가져올 수 있다.

또한 유전 알고리즘의 수행에 중요한 영향을 미치는 교배, 돌연변이, 역전 유전 연산자를 차량단위로 적용함으로써, 보다 빠르게 적합도가 수렴해 나아가게 된다. 특히, 교배(crossover) 연산자는 기존차량의 정보를 다음 세대에 전달함으로써 보다 신속하게 해의 수렴을 유도할 수 있는 장점을 가지고 있으며, 돌연변이 연산자의 경우에는 내부 돌연변이(internal mutation)와 외부 돌연변이(external mutation)로 구분하여 실행함으로써 보다 효율적으로 국부 최적해로부터 벗어날 수 있다.

본 연구에서 제시하는 병렬 유전자 알고리즘과 기존의 유전자 알고리즘을 N. Christofides 등이 제시한 10개의 문제에 대하여 비교 실험한 결과 10개의 문제 중 8개의 문제에서 보다 우수

한 해의 수렴속도를 얻을 수 있었으며, 2개의 문제에서는 기존 알고리즘과 유사하거나 동일한 해의 수렴속도를 얻을 수 있었다. 특히 수요지점이 많은 문제의 경우에서도 초기 100세대 이전부터 보다 우수한 해를 얻음으로써 빠른 해의 수렴속도를 얻을 수 있다.

추후의 연구내용으로는, 유전자 알고리즘에서 중요한 매개변수에 대한 보다 많은 수준실험을 통하여 최적의 진화 환경을 구축할 필요가 있다. 또한 적절한 시기에 알고리즘의 수행을 마치기 위해서는 시스템에서 만족하는 해의 수준을 어떻게 정할 것인가에 대한 연구가 필요하다고 생각된다. 또한 유전자 알고리즘의 특성상 계산시간이 많이 걸리는 단점을 극복할 수 있는 방법에 대한 연구가 필요하다고 생각된다.

### 참고문헌

- 신해웅 (1994), 혼합형 유전해법을 이용한 배송차량의 경로결정, 박사학위 논문, 한양대학교 대학원.
- Cheng, R. and Gen, M. (1996), *Genetic Algorithm and Engineering Design*, John Willy & Sons.
- Cheng, R. and Gen, M. (1995), Minmax earliness / tardiness scheduling in identical parallel machine system using genetic algorithm, *International Journal of Computers and Industrial Engineering*, 29(1-4), 513-517.
- Christofides, N. and Eilon, S. (1969), An algorithm for the vehicle dispatching problem, *Operational Research Quarterly*, 20(3), 309-318.
- Clarke, G. and Wright, J. (1964), Scheduling of vehicles from a central depot to a number of delivery points, *Operations Research*, 12(4), 568-581.
- Dantzig, G. B. and Ramser, J. H. (1959), The Truck Dispatching Problem, *Management Science*, 6(1), 80-91.
- Davis, L. (1991), *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York.
- Eilon, S., Watson-Gandy, C.D.T. and Christofides, N. (1971), *Distribution Management : Mathematical Modelling and Practical Analysis*, Hafner, New York.
- Gendreau, M., Hertz, A. and Laporte, G. (1994), A tabu search heuristic for the vehicle routing problem, *Management Science*, 40(10), 1276-1290.
- Lee, C. and Kim, S. (1995), Parallel genetic algorithm for the tardiness job scheduling problem with general penalty weights, *International Journal of Computers and Industrial Engineering*, 28, 231-243.
- Zbigniew Michalewicz (1996), *Genetic Algorithm + Data Structure = Evolution Programs*, Springer-Verlag, Berlin Heidelberg.