

다목적 환경에서의 ATIS 운영을 위한 A^* 탐색 알고리즘과 유전자 알고리즘의 혼합모형

장인성

승실대학교 산업·정보시스템공학과

A Hybrid Model of A^* Search and Genetic Algorithms for ATIS under Multiple Objective Environment

In-Seong Chang

This paper presents a new approach which uses A^* search and genetic algorithms for solving large scale multi-objective shortest path problem. The focus of this paper is motivated by the problem of finding Pareto optimal paths for an advanced traveler information system(ATIS) in the context of intelligent transportation system(ITS) application. The individual description, the decoding rule, the selection strategy and the operations of crossover and mutation are proposed for this problem. The keynote points of the algorithm are how to represent individuals and how to calculate the fitness of each individual. The high performance of the proposed algorithm is demonstrated by computer simulations.

1. 서론

네트워크 모델 중에서 자주 언급되어지는 중요한 문제 중의 하나가 최단경로문제(Shortest Path Problem)이다. 최근에 지능형 교통체계(Intelligent Transportation System : ITS)에 대한 연구가 세계적으로 활발해지면서 ITS 분야의 하나인 첨단 여행자 정보체계(Advanced Traveler Information System : ATIS)의 운영을 위해 도로망의 최단경로탐색에 대한 관심이 높아지고 있다. ATIS 하에서 최적경로에 대한 정보를 여행자에게 효율적으로 제공하기 위하여 최단경로탐색 알고리즘은 매우 중요한 핵심 기술이다.

오늘날의 도로망은 규모가 매우 커서 소규모 도로망을 대상으로 개발된 기존의 최단경로탐색 알고리즘을 이용해서는 단시간에 최단경로를 산출하지 못한다. 또한 교차로에서의 좌회전금지, U-turn 및 P-turn 허용 등과 같이 도로망의 도로행태가 복잡하기 때문에 기존에 개발된 최단경로탐색 알고리즘을 이용하는 경우 현실적인 최단경로와 다른 비현실적인 경로를 찾게 된다. 따라서, 대규모 도로망의 최단경로를 빠른시간내에 산출하거나 실제적인 도로행태를 정확하게 반영한 최단경로 탐색 알고리즘의 개발은 ATIS의 운영을 위해 매우 중요하며

이에 대한 연구가 절실히 요구되어지고 있다.

Dijkstra(1959) 알고리즘은 최단경로문제의 대표적인 탐색 알고리즘으로 널리 알려져 있다. Dijkstra 알고리즘은 네트워크의 규모가 복잡하거나 커지면 많은 계산량을 소요한다. 이러한 단점을 극복하기 위해 인공지능(Artificial Intelligence : AI) 분야에서 사용되는 발견적 탐색기법인 A^* 알고리즘(Winston, 1992)이 대규모 네트워크의 최단경로를 탐색하기 위해 적용되어져 왔다. 또한 Dijkstra 알고리즘은 좌회전 금지, U-turn 및 P-turn 허용 등의 현실적인 도로행태를 정확하게 반영한 최단경로를 탐색하지 못한다. 이와 같은 복잡한 도로행태를 반영하기 위해 링크탐색 알고리즘(이승환 외, 1996)과 수정형 덩굴망 알고리즘(김익기, 1998)이 개발되어졌다.

하지만 이러한 탐색 알고리즘들은 통행거리, 통행시간 또는 통행비용 등과 같이 최단경로문제에서 발생할 수 있는 다양한 목적들 중에서 오직 하나의 목적만을 고려 대상으로 하는 단일목적 최단경로문제(Single Objective Shortest Path Problem)를 해결하기 위해 개발되어졌다. 그러나 실제로는 도로를 이용하는 여행자가 단일목적만을 기준으로 경로를 선택하는 경우는 드물며, 경로선택은 상충적인 관계를 지닌 다양한 목적들을 종합적으로 고려해서 결정되어진다. 그러므로 단일목적 최단경로문제는 실제의 교통모형을 잘 반영하지 못하며 여행자의

다양한 목적을 동시에 고려하여 최적경로를 선택하고자하는 다목적 최단경로문제(Multi-objective Shortest Path Problem)가 더욱 현실적이다.

다목적 최단경로문제에서의 경로탐색은 이들 목적 간의 상충적인 관계로 인해 간단하지 않다. 예를 들면, 도로망의 기점과 종점을 잇는 경로의 통행비용과 통행시간은 비용이 적게 들수록 시간이 오래 걸리고, 시간이 적게 걸릴수록 비용이 커지는 상충적인 관계가 있다고 할 수 있다. 이러한 경우에 주어진 목적 모두를 만족시키는 최적경로는 존재치 않기 때문에, 이들 목적간의 절충(trade-off)을 고려한 파레토 최적경로(Pareto optimal path)가 탐색되어야 한다. 파레토 최적경로는 다른 경로들에 의하여 지배되지 않는 경로(nondominated path)를 의미한다. 주어진 도로망의 파레토 최적경로는 다수 존재하며, 탐색되어진 이들 파레토 최적경로가 여행자를 위한 다양한 정보로써 제공되어질 수 있다. 그러나 기존의 다목적 최단경로탐색 알고리즘은 하나 또는 일부의 파레토 최적경로만을 탐색하기 때문에 다양한 정보를 제공하지 못한다. 따라서 가능한 한 많은 파레토 최적경로를 탐색할 수 있는 새로운 최단경로탐색 알고리즘의 개발이 요구되어지며, 이에 대한 연구는 장래 ATIS의 운영을 위해 매우 필요하다.

본 연구에서는 조합최적화 문제의 효율적인 탐색기법인 유전자 알고리즘(Genetic Algorithm : GA)과 단일목적 최단경로문제의 AI적 탐색기법인 A* 알고리즘을 이용하여 대규모인 다목적 최단경로문제의 파레토 최적경로를 가능한 한 모두 효율적으로 탐색할 수 있는 새로운 알고리즘을 개발하고자 한다. 본 연구에서 개발한 새로운 다목적 최단경로탐색 알고리즘은 현실성이 반영된 다목적 환경 하에서 도로망의 다양한 경로정보를 효율적으로 여행자에게 제공하기 위한 ATIS의 분석에 유용하게 활용될 수 있는 기법이라고 고려된다.

2. 다목적 최단경로문제

본 연구에서는 다음과 같은 두 개의 상충된 목적함수를 다룬다.

- z_1 : 기점과 종점을 잇는 경로의 총 통행시간의 최소화
- z_2 : 기점과 종점을 잇는 경로의 총 통행비용의 최소화

노드 i 와 j 를 연결하는 링크를 (i, j) 로 표현하고 링크 (i, j) 의 통행시간을 t_{ij} , 통행비용을 c_{ij} , 주어진 네트워크의 기점과 종점을 연결하는 모든 가능한 경로의 집합을 P , 그리고 P 안에 있는 어떤 임의의 경로를 p 라하면, 두 개의 목적을 수반하는 최단경로문제는 다음과 같이 정식화 할 수 있다.

$$\begin{aligned} \min z_1(p) &= \sum_{(i,j) \in p} t_{ij} \\ \min z_2(p) &= \sum_{(i,j) \in p} c_{ij} \\ \text{subject to} \quad & p \in P. \end{aligned}$$

위 문제의 이상적인 해(ideal solution)는 각 목적함수를 독립적으로 고려해서 해를 구함으로써 얻어지며, 목적함수 영역(objective function space)에서 다음과 같이 나타낼 수 있다.

$$I = (z_1^*(p), z_2^*(p))$$

여기서 $z_k^*(p) = \min_{p \in P} z_k(p)$, $k=1,2$. 목적함수 $z_1(p)$, $z_2(p)$ 는 서로 상충되어지기 때문에 두 목적함수를 모두 최적화하는 이상적인 값 I 에 대응되는 최적경로 p 는 존재하지 않는다. 이러한 경우에 최적경로(최적해)의 개념은 파레토 최적경로(파레토 최적해)의 개념으로 바뀌어져야 한다 (Steuer, 1986).

[정의 1] 두 경로 $p_1, p_2 \in P$ 에 대해 아래의 조건들 중 어느 하나가 성립되면 경로 p_1 은 경로 p_2 를 지배한다.

- 조건 1. $z_1(p_1) < z_1(p_2)$, $z_2(p_1) \leq z_2(p_2)$
- 조건 2. $z_2(p_1) < z_2(p_2)$, $z_1(p_1) \leq z_1(p_2)$

[정의 2] 어떤 특정 경로 $p_0 \in P$ 를 지배하는 어떠한 경로 $p_k \in P$ 도 존재치 않으면 p_0 는 파레토 최적경로이다.

파레토 최적경로는 유일하게 존재하는 것이 아니라 다수 존재하며 이들은 목적함수 영역에서 파레토 최적경계선(Pareto optimal frontier)을 형성한다. <그림 1>은 목적함수 영역에서 파레토 최적경로들을 나타내고 있다. 점선은 파레토 경계선(Pareto frontier)을 나타내며 특정 파레토 경계선상에 놓여있는 경로들은 서로를 지배하지 않는다. 이상적인 해 I 에 가장 가까이 위치하고 있는 파레토 경계선이 파레토 최적경계선을 나타내며, 이 경계선상에 놓여 있는 경로들이 다른 경로들에 의하여 지배되지 않는 파레토 최적경로에 해당된다. 주어진 파레토 최적경로들을 이용하면 여행자는 두 목적 간의 절충을 평가할 수 있기 때문에 경로의 다양한 선택이 가능하다. 따라서 도로망을 효율적으로 이용하고 여행자의 편리성을 향상시킬 수 있다.

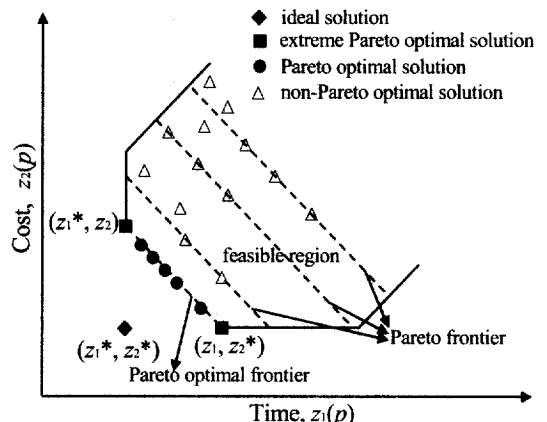


그림 1. 파레토 최적해.

다목적 최단경로문제는 조합최적화 문제에 해당하기 때문에 파레토 최적경로를 탐색하는 데 계산시간이 많이 소요되어진다. 또한 주어진 네트워크의 노드 수가 증가하면 파레토 최적경로의 수는 지수적으로 증가하기 때문에 모든 파레토 최적경로를 탐색하는 것은 불가능하다(Hansen, 1980). 이러한 이유로 기존의 알고리즘들은 파레토 최적경로들의 오직 하나 또는 일부분만을 탐색하기 위해 개발되어졌다. 그러나 이러한 소수의 파레토 최적경로들은 여행자에게 제공되는 정보로서 충분치 못하다. 따라서 다양한 경로정보를 효율적으로 여행자에게 제공할 수 있는 알고리즘의 개발이 다목적 환경을 고려한 장래 ATIS의 운영을 위해 필수적이다.

3. 다목적 최적화기법

다목적 최단경로문제는 현실성이 잘 반영된 다목적 최적화 문제의 한 예로써 조합최적화 문제에 해당하기 때문에 파레토 최적경로들의 탐색은 쉽지 않으며 많은 계산시간을 요구한다. 다목적 최적화 문제의 일반적인 기법은 각각의 목적에 가중치를 부여하여 다목적을 단일목적으로 변형하는 것이다. 수리적인 계산법에 의해 파레토 최적해가 쉽게 얻어지나 단지 하나의 파레토 최적해가 얻어지므로 본 문제에 적용하는 경우 여행자를 위한 정보로써 충분치 못하다.

Climaco and Martins(1982)는 두 개의 목적함수 z_1 (총통행시간), z_2 (총통행비용)을 수반하는 최단경로문제에 대하여 파레토 최적경로의 일부를 얻기위해 K 최단경로 알고리즘(Lawler, 1972)을 도입하였다. K 최단경로 알고리즘은 단일목적 최단경로문제에서 목적함수 z 를 최소화하는 하나의 경로 p_1 을 찾는 것이 아니라 $z(p_1) < z(p_2) < \dots < z(p_k)$ 인 k개의 경로를 구하는 것이다. K 최단경로 알고리즘은 도로망이 일시적으로 두절 상태에 처할 경우 효과적으로 대안 경로를 선택할 수 있도록 해준다. Climaco and Martins는 K 최단경로 알고리즘을 이용하여 목적함수의 어느 한쪽을 최소화하는 k 순위개의 경로를 구하고 이것들에 대하여 파레토 최적경로를 탐색하였다. Climaco and Martins의 알고리즘은 파레토 최적해가 목적들 중의 어느 한쪽에 집중되는 문제점을 갖고 있다. 본 연구에서 제안되어지는 알고리즘과 비교분석 되어질 Climaco and Martins의 파레토 최적경로탐색 알고리즘의 절차는 아래와 같다.

Procedure : Climaco and Martins의 Algorithm

- Step 0: 1. 파레토 최적경로들의 집합 $\Phi = \emptyset$.
- 가장 빠른 경로 p_1^c 를 찾고 $\hat{c} = z_2(p_1^c)$ 를 계산한다.
 - K 최단경로 알고리즘을 이용하여 $z_2(p_1^c) < z_2(p_2^c) < \dots < z_2(p_k^c)$ 인 k개의 경로를 구한다.
 - 가장 저렴한 경로 p_1^c 에 대하여 $z_1(p_1^c) = z_1(p_1^c)$ 이라면 $\hat{c} = z_2(p_1^c)$ 를 재계산한다.
 - $\Psi = \{p_1^c\}$, $f_1 = z_2(p_1^c)$, $f_2 = z_1(p_1^c)$ 를 계산한다.

6. $j = 2$.

- Step 1: 1. j번째로 저렴한 경로 p_j^c 를 찾는다.
 2. 만일 $z_1(p_j^c) = z_1(p_1^c)$ 이라면 $\hat{c} = \min\{z_2(p_j^c), \hat{c}\}$ 를 재계산한다.
- Step 2: 만일 $z_2(p_j^c) = f_1$ 이라면 Step 4로 이동한다.
- Step 3: 만일 $z_1(p_j^c) \geq f_2$ 이라면 Step 5로 이동한다. 그렇지 않으면 $\Phi = \Phi \cup \Psi$, $\Psi = \{p_j^c\}$, $f_1 = z_2(p_j^c)$, $f_2 = z_1(p_j^c)$ 를 재계산하고 Step 5로 이동한다.
- Step 4: 1. 만일 $z_1(p_j^c) > f_2$ 이라면 $j = j + 1$ 로 하고 Step 1로 재이동한다.
 2. 만일 $z_1(p_j^c) < f_2$ 이라면 $\Psi = \{p_j^c\}$, $f_2 = z_1(p_j^c)$ 를 재계산하고 Step 5로 이동한다.
 3. 만일 $z_1(p_j^c) = f_2$ 이라면 $\Psi = \Psi \cup \{p_j^c\}$, $j = j + 1$ 로하고 Step 1로 재이동한다.
- Step 5: 1. 만일 $z_2(p_j^c) \leq \hat{c}$ 이라면 $j = j + 1$ 로 하고 Step 1로 재이동한다.
 2. 만일 $z_2(p_j^c) > \hat{c}$ 이거나 $j = k + 1$ 이라면 $\Phi = \Phi \cup \Psi$ 로하고 알고리즘을 종결한다.

최근에 조합최적화 문제의 효율적인 탐색기법인 유전자 알고리즘을 각종의 다목적 최적화문제에 적용하여 다수의 파레토 최적해를 탐색하고자 하는 연구들이 활발히 수행되어지고 있으며 적용상의 우수성이 입증되었다(Horn et al., 1994; Kim et al., 1996; Murata et al., 1996; Osyczka and Kundu, 1996; Schaffer, 1985; Zhou and Gen, 1999). 이들 방법의 기본적인 개념은 비 파레토 최적해(non-Pareto optimal solution)들로 구성된 초기 가능해들의 집합을 자연세계의 진화과정을 모방하여 파레토 최적 경계선 상에 놓여있는 파레토 최적해들로 갱신하고자 하는 것이다(<그림 2>).

유전자 알고리즘을 다목적 최단경로문제에 적용함으로써 다수의 파레토 최적경로의 효율적인 탐색이 또한 가능하나 현재까지 보고된 연구사례는 없다. 그 이유는 다목적 최단경로 문제에 적합한 유전자 알고리즘의 구성에 많은 어려움이 존재

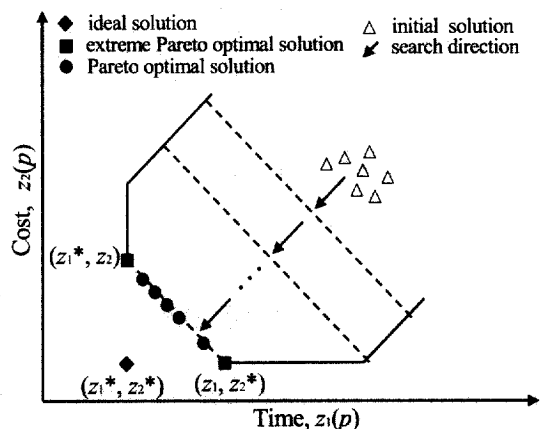


그림 2. GA의 파레토 최적해 탐색 방향.

하기 때문이다. 유전자 알고리즘을 다목적 최단경로문제에 적용하기 위해서는 우선 도로망의 기점과 종점을 연결하는 임의의 경로를 유전자형으로 표현할 수 있어야 한다. 또한, 유전 연산자가 수행된 뒤 초래되는 결과는 통행 가능한 경로를 나타내어야 한다. 그러나 이러한 조건을 충족시키는 유전자형의 설계는 쉽지 않다. 최근에 장인성과 이승재(1999)는 다목적 최단경로문제에 대한 파레토 최적경로를 모두 탐색할 수 있는 유전자 알고리즘을 제안하였으나 네트워크의 규모가 커지면 적용하기가 곤란해지는 문제점을 갖고 있다. 본 연구에서는 현실적인 대규모의 네트워크에도 적용할 수 있도록 조합최적화 문제의 효율적인 탐색기법인 유전자 알고리즘과 대규모 네트워크의 최단경로를 효율적으로 탐색하는 A^* 알고리즘을 적용하여 파레토 최적경로를 가능한 한 모두 신속하게 탐색할 수 있는 효과적인 해법을 개발하고자 한다.

4. A^* 탐색 알고리즘

A^* 알고리즘은 인공지능분야에서 사용되는 발전적 탐색기법 중의 하나로 단일목적 최단경로문제에 적용하는 경우에 최적해를 보장해준다. Dijkstra 알고리즘은 최단경로를 찾기 위해 네트워크의 모든 노드를 탐색하므로 탐색속도가 느리고 네트워크가 커지면 적용하기가 곤란해지는 문제점이 있다. 이에 반해서 A^* 알고리즘은 문제에 대한 특별한 정보를 이용해서 탐색영역을 제한하기 때문에 탐색작업이 효율적이며 탐색속도가 빠르다.

A^* 알고리즘은 최단경로를 구축하는 과정에서 다음과 같은 비용함수를 최소로 하는 노드 n 을 탐색한다.

$$f(n) = g(n) + \hat{h}(n)$$

여기서, $g(n)$ 은 기점으로부터 현 노드 n 까지의 경로비용을 나타내며 $\hat{h}(n)$ 은 현 노드 n 에서 종점까지의 잔여 경로비용인 $h(n)$ 의 추정치를 나타낸다. A^* 알고리즘에서는 $\hat{h}(n) \leq h(n)$ 이 되도록 $\hat{h}(n)$ 을 설정하면 대상문제의 최적해를 탐색하는 것이 보장된다. $\hat{h}(n)$ 의 값은 주어진 네트워크의 모든 노드에 따라 다르며 $\hat{h}(n)$ 이 실제의 $h(n)$ 에 얼마만큼 가깝게 추정되는가에 따라 탐색범위가 줄어든다. 따라서 $\hat{h}(n)$ 의 설정이 A^* 알고리즘의 탐색효율을 결정하며, $\hat{h}(n)$ 은 주어진 네트워크의 정보를 나타내는 휴리스틱 함수를 의미한다.

5. 파레토 최적경로의 탐색을 위한 유전자 알고리즘

유전자 알고리즘은 자연세계의 진화과정을 공학적으로 모방한 확률적 계산모형으로써 전역적(global)인 최적해를 병렬적으로 탐색하는 기법이다. 유전자 알고리즘에서는 대상이 되는

문제의 후보해들을 문자나 기호의 배열인 염색체(chromosome)로 표현하며, 이 절차를 코드화(encoding)라 한다. 염색체를 구성하는 문자나 기호들을 유전자(gene)라고 부르며, 염색체를 문제의 대응되는 후보해로 변형하는 역의 과정을 디코드화(decoding)라 한다. 염색체를 개체(individual)라 언급하며, 이들 개체들의 적당한 크기(population size)로 구성되는 모집단(population)을 생성하여 이들을 점차적으로 갱신함으로써 전역적인 최적해를 탐색하게 된다. 모집단의 생성과정을 초기화(initialization)라 하며 모집단의 갱신은 유전 연산자(genetic operator)에 해당되는 선택(selection), 교배(crossover), 돌연변이(mutation)의 세 가지 과정을 통해서 이루어지고 이를 1 세대(generation)라 한다. 해의 탐색과정에서 모집단을 이용한 병렬처리를 수행하기 때문에 탐색 소요시간이 매우 짧고, 모집단의 갱신과정에서 개체의 수를 유지하므로 다목적 최적화 문제의 다양한 파레토 최적해들을 탐색하기에 매우 적절한 기법이다.

5.1 모집단의 생성

유전자 알고리즘을 적용하는 첫 단계이며 가장 중요한 것은 대상이 되는 문제의 후보해를 염색체로 표현(encoding)하기 위한 방법을 설계하는 과정이다. 본 연구에서 다루는 문제는 도로망의 기점과 종점을 잇는 경로를 탐색하는 것이므로 네트워크상의 모든 경로를 염색체에 대응시킬 수 있어야 한다.

유전자 알고리즘이 네트워크 모델에 적용된 대표적인 예는 순회판매원문제(Traveling Salesman Problem)에서 찾을 수 있으며, 유전자의 배열을 노드의 조합으로 나열하여 염색체를 구성함으로써 후보해를 표현하고 있다(Michalewicz, 1994). Vignaux and Michalewicz(1991)은 수송문제(Transportation Problem)에 유전자 알고리즘을 적용하였으며, 염색체를 구성하는 유전자의 배열을 행렬의 형태나 노드의 조합으로 나열하여 후보해를 표현하고 있다. 최근에 Zhou and Gen(1999)은 최소연결나무문제(Minimal Spanning Tree Problem)에 염색체를 구성하는 유전자의 배열을 노드들의 조합인 Prüfer number(Dossey et al., 1993)로 표현하여 유전자 알고리즘을 구성하였다.

그러나 위와 같이 후보해를 행렬의 형태나 노드의 조합에 의하여 염색체로 encoding 하는 방법을 최단경로문제에 적용하는 경우에는 후보해로의 변형이 불가능하거나 유전 연산자가 수행된 뒤 존재하지 않는 경로가 발생한다. 이와 같은 문제점을 효율적으로 해결하기 위하여 본 연구에서는 행렬의 형태나 노드의 조합대신에 링크에 기초로 해서 염색체를 표현하는 방법(장인성, 김현철, 1999)을 사용하였다. 이 방법은 염색체의 유전자를 네트워크의 링크 수만큼 확보하고 유전자를 링크의 이용값으로 표현한다. 만약 링크가 n 개일 경우 염색체는 다음과 같이 n 개의 유전자들로 구성된다.

$$C = (c[1]c[2] \cdots c[n-1]c[n])$$

여기서 i 번째 유전자 $c[i](i=1, 2, \dots, n)$ 는 링크 i 의 이용

값을 나타내며 0과 1 사이의 실수로 표현되어진다. 구체적으로는 아래와 같은 초기화 절차에 의해 모집단을 생성하였다.

Procedure : Initialization

```

begin
  l=1;
  while(l ≤ population_size) do
    begin
      i=1;
      while(i ≤ number_of_links) do
        begin
          generate_random_number_c[i];
          if(c[i] < 0.5) then
            link[i]_value_(time, cost)
              = infinite_value_(time, cost);
          else
            link[i]_value_(time, cost)
              = original_value_(time, cost);
          i=i+1;
        end
      end
      l=l+1;
    end
  end
end
    
```

*l*은 염색체(개체)의 번호, *population_size*는 모집단의 크기, *i*는 링크 번호를 나타내며 *number_of_links*는 주어진 네트워크를 구성하는 링크들의 수를 나타낸다.

*generate_random_number_c[i]*에서 각 링크의 이용값 *c[i]*를 0과 1사이의 임의의 실수값으로 초기화하고, 그 값이 미리 지정된 값(0.5)보다 작으면 대응되는 *i*번째 링크는 매우 큰 값의 통행시간 및 통행비용을 갖는 가상링크(dummy link)로 대체되어진다. 반면에 *c[i]*의 값이 0.5보다 크거나 같으면 대응되는 *i*번째 링크는 본래의 통행시간 및 통행비용을 갖는 실링크(real link)가 부여되어진다. 따라서, 염색체는 네트워크의 링크들을 가상링크와 실링크로 재구성하여 원래의 네트워크와는 다른 구조를 갖는 네트워크를 형성시킨다. 또한 상기의 절차에 의하여 생성된 염색체들은 서로 구별되는 네트워크의 형상을 갖기 때문에 각각의 염색체들은 독립된 개체에 대응되어진다.

5.2 염색체의 변환 및 평가

다음 단계는 각 개체의 염색체를 기점과 종점을 있는 어떤 경로로 변환(decoding)하고, 해로서의 우수성을 평가하는 것이다. 본 연구에서 사용된 염색체의 구조는 실링크와 가상링크로 구성된 네트워크 자체를 나타내므로 decoding 절차가 간단하게 수행될 수 있다.

<그림 3>은 24개의 링크와 16개의 노드로 구성된 격자형

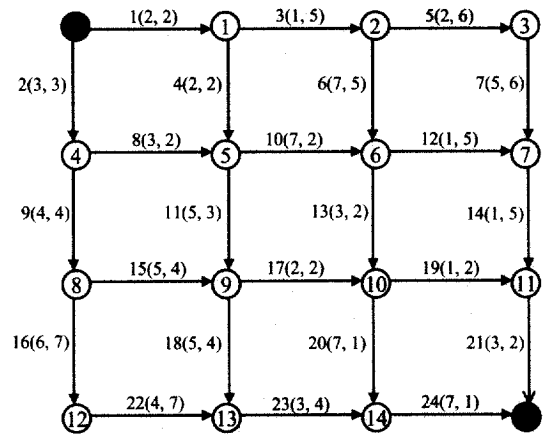


그림 3. 주어진 네트워크.

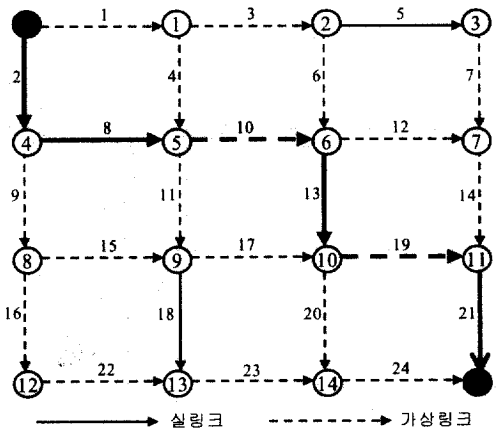


그림 4. 염색체 C에 의하여 재구성된 네트워크.

네트워크(노드×노드= 4×4)를 보여준다. 링크 상에 존재하는 숫자는 대응되는 링크 번호이며, 괄호 안의 숫자는 링크를 통과하는데 소요되는 시간과 비용을 나타낸다. 노드 *o*와 노드 *d*는 각각 기점과 종점을 나타낸다.

<그림 4>는 염색체 $C = (0.001, 0.535, \dots, 0.113, 0.004)$ 에 의해 재구성된 네트워크를 보여준다. 점선으로 표시되어진 링크들은 링크 이용값이 0.5보다 작은 염색체 C의 유전자에 대응되며 매우 큰 값의 통행시간 및 통행비용이 부여된다. 반면에 실선으로 표시된 링크들은 링크 이용값이 0.5이상인 염색체 C의 유전자에 대응되며, 링크 본래의 통행시간 및 통행비용이 부여된다. 각각의 염색체에 대하여 경로로 변환하는 과정은 아래의 절차에 의해 수행되어진다.

Procedure : Decoding

```

begin
  l=1;
  while(l ≤ population_size) do
    begin
      i=1;
    end
  end
end
    
```

```

while (i ≤ number_of_links) do
begin
    link[i]_value = time + cost ;
    i = i + 1 ;
end
l = l + 1 ;
end
l = 1 ;
while (l ≤ population_size) do
begin
    calculate_path(by A* algorithm) ;
    fitness[l] = total_link_value_(time, cost) ;
    l = l + 1 ;
end
end
end
    
```

link[i]_value에서 탐색체에 의해 재구성된 네트워크의 각 링크와 관련된 통행시간과 통행비용의 합을 각 링크의 값으로 설정한다. calculate_path에서 재구성된 네트워크로부터 링크의 총값이 최소가 되는 경로를 도출한다. 이때 경로를 도출하기 위해 모든 노드를 탐색하는 Dijkstra 알고리즘을 이용할 수 있지만 많은 계산량이 수반되어진다. 게다가 GA의 디코드화 절차가 매 세대마다 개체들에 대하여 수행되어지므로 <그림 3>과 같은 소규모 네트워크에서도 GA를 수행하는데 많은 계산 시간이 요구되어진다. 따라서 Dijkstra 알고리즘이 적용된 디코드화 절차를 이용해서는 GA를 대규모 네트워크에 적용 불가능하다. 본 연구에서는 제안하는 GA의 수행속도를 향상시키기 위해 제한된 영역만을 탐색하면서도 최적경로를 찾을수 있는 A* 알고리즘을 적용 하였다. 먼저, 각 링크의 통행시간과 통행비용의 합인 link[i]_value를 이용해서 각 노드를 좌표상의 점으로 나타내고, 네트워크의 노드 n에서 종점노드 d까지의 잔여 경로비용의 추정치를 나타내는 함수 $\hat{h}(n)$ 로써 두 노드와의 직선거리를 사용하였다. $xcor(I)$, $ycor(I)$ 를 각각 노드 I의 X, Y좌표라 한다면 $\hat{h}(n)$ 은 다음과 같다.

$$\hat{h}(n) = \sqrt{(xcor(n) - xcor(d))^2 + (ycor(n) - ycor(d))^2}$$

calculate_path에서 A* 알고리즘에 의해 가능한 한 많은 실선의 링크로 구성된 경로가 도출되어지며, 탐색체 C의 경우는 4개의 실링크(2, 8, 13, 21)와 2개의 가상링크(10, 19)로 구성되는 다음과 같은 경로이다(<그림 4>).

$$p = 2 - 8 - 10 - 13 - 19 - 21$$

fitness[l]에서 도출된 경로를 구성하는 링크들의 총 소요시간 z_1 과 총 소요비용 z_2 를 계산해서 개체의 우수성을 평가한다. 유전 알고리즘의 진화과정을 통해 결국은 <그림 5>와 같이 실선의 링크만으로 구성된 경로를 갖고 있는 탐색체들로 갱신

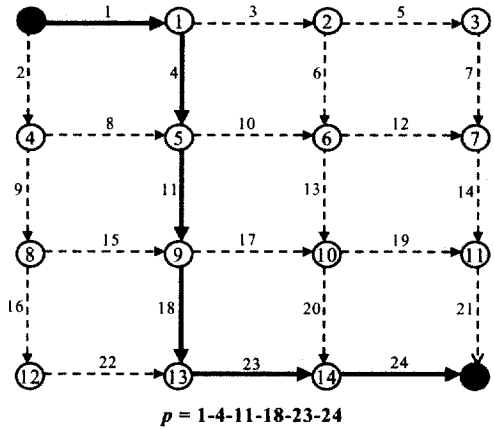


그림 5. 갱신된 탐색체 C'

되어진다.

5.3 선택

선택이란 현재의 모집단을 갱신시키기 위한 준비단계로써 지정된 전략에 따라 우수한 개체를 다음세대로 복제(reproduction)시키거나 교배를 위한 개체의 쌍을 만들기 위해 현재의 모집단을 재편성하는 과정이다. 대표적인 선택전략으로써 엘리트 보존전략과 roulette wheel 전략 등이 있다(Michalewicz, 1994). 이들 선택전략은 단일 목적함수만을 고려하여 개체의 적응도를 평가한다. 따라서 다목적 최단경로 문제와 같이 다수의 목적을 동반하는 경우에는 이들 모두를 반영하는 새로운 선택전략이 적용되어져야 한다.

다목적 최적화 문제에 유전자 알고리즘을 적용하기 위한 최초의 연구는 병렬탐색 전략을 개발한 Schaffer(1985)에 의해 수행되어졌다. Schaffer의 병렬탐색 전략은 모집단을 두 목적함수 z_1 과 z_2 에 대해 균등하게 분배하고, 각각의 z_1 과 z_2 에 대해 독립적으로 선택을 행하는 것이다. 파레토 최적해의 일부가 얻어지나 각 목적함수의 최적화를 도모하기 때문에 목적함수 영역에서 양 극점(extreme Pareto optimal solution)에 치우치는 파레토 최적해들이 나타난다. 따라서 모집단의 다양성을 상실하게 되므로 다양한 파레토 최적해의 탐색이 불가능하다.

본 연구에서는 선택전략으로 단일 목적의 최적화를 위한 엘리트 보존전략과 다목적 최적화를 위한 Schaffer의 병렬탐색 전략을 결합한 혼합전략을 사용하였다. 엘리트 보존전략은 개체의 적응도가 높은 것은 다음 세대에 그대로 복제하는 것으로 다목적 최단경로문제에서는 파레토 최적개체의 후보들을 교배, 돌연변이의 수행 없이 다음세대로 복제하는 개념으로 확장될 수 있다. 두 선택전략을 결합한 혼합전략은 다음과 같은 단계로 수행된다.

Procedure : Selection

Step 1 : decoding 절차의 fitness[l]에서 계산된 각 개체의(z_1 ,

z_2)를 비교해서 파레토 최적성의 정의에 따라 현 모집단으로부터 파레토 최적개체의 후보들을 추출한다.

Step 2: 추출된 파레토 최적개체 후보들의 수가 모집단의 수와 같다면 현 후보개체들에 대해 Schaffer의 병렬 탐색 전략을 수행한다. 그렇지 않으면, Step 3을 수행한다.

Step 3: 추출된 파레토 최적개체 후보 전부를 다음 세대로 복제하고, 나머지 수만큼의 개체를 현 모집단에 대해 Schaffer의 병렬탐색 전략을 수행함으로써 추출한다.

5.4 교배

교배는 재편성된 모집단으로부터 일정한 교배확률에 따라 교배를 위한 개체의 쌍을 만들어 각 개체의 유전자 교환을 통해 모체보다 우수한 개체를 생성시키기 위한 단계이다. 본 연구에서는 표현되어진 염색체 구조의 특이성으로부터 1점 가중합 교배(one-point weighted sum crossover) 라는 새로운 교배 연산법을 제안한다. 이 새로운 교배 연산법은 먼저, 교배를 위해 선택된 한 쌍의 개체의 염색체에 대해 1점 교배를 적용해서 교배 위치를 정하고 두 염색체의 유전자의 가중합을 교환함으로써 새로운 개체의 염색체를 생성하는 것이다.

교배확률(p_c)에 따라서 교배를 위해 선택된 한 쌍의 개체의 염색체 A, B가

$$A = (a[1] \ a[2] \ \dots \ a[n-1] \ a[n])$$

$$B = (b[1] \ b[2] \ \dots \ b[n-1] \ b[n])$$

인 경우, 교배를 통해 생성되는 새로운 두 개체의 염색체는 주어진 네트워크의 링크 수 n 이 짝수라면 다음과 같다.

$$A' = a'[i] = \alpha \times a[i] + (1 - \alpha) \times b[i+k], \text{ if } i \leq k,$$

$$= a[i], \text{ if } i > k,$$

$$B' = b'[i] = \alpha \times b[i] + (1 - \alpha) \times a[i+k], \text{ if } i \leq k,$$

$$= b[i], \text{ if } i > k.$$

여기서, k 는 $n/2$ 의 정수 값으로써 교배 위치를 나타내며 α 는 0과 1사이의 임의의 값이다. n 이 홀수인 경우에는 다음과 같은 두 개체의 염색체가 생성된다.

$$A' = a'[i] = \alpha \times a[i] + (1 - \alpha) \times b[i+k+1], \text{ if } i \leq k,$$

$$= a[i], \text{ if } i > k,$$

$$B' = b'[i] = \alpha \times b[i] + (1 - \alpha) \times a[i+k+1], \text{ if } i \leq k,$$

$$= b[i], \text{ if } i > k.$$

5.5 돌연변이

돌연변이란 각 개체에 대해 일정한 돌연변이 확률로 모체의

성질과는 전혀 다른 새로운 개체를 발생시키는 과정이다. 돌연변이 확률이 높을수록 광범위한 영역을 탐색할 수 있어 국지해(local solution)에 수렴할 가능성을 낮출 수 있지만 수렴속도는 늦어지게 된다. 본 연구에서는 모집단의 다양성을 확보하기 위해 돌연변이 확률(p_m)에 의해서 돌연변이가 실시되는 개체의 염색체의 모든 유전자를 0과 1사이의 값을 이용하여 완전히 교체한다.

5.6 GA 절차

제안된 알고리즘의 전체적인 절차는 다음과 같다.

Procedure : GA

begin

$t = 0$;

initialize the population of solutions $P(t)$ by initialization procedure ;

evaluate $P(t)$ by decoding procedure ;

determine the set of Pareto optimal solutions $E(t)$;

while (not termination condition) do

begin

select $P(t+1)$ from $P(t)$ by selection procedure ;

recombine $P(t+1)$;

evaluate $P(t+1)$ by decoding procedure ;

update $E(t+1)$;

end

$t = t + 1$;

end

6. 성능평가 및 분석

제안된 알고리즘의 성능을 검증 및 평가하기 위해서는 주어진 네트워크에 존재하는 파레토 최적경로들을 모두 알고 있어야 한다. 그러나 주어진 네트워크의 파레토 최적경로들을 모두 탐색할 수 있는 알고리즘이 존재하지 않으므로 파레토 최적경로들은 네트워크의 통행 가능한 경로를 모두 열거하여 비교함으로써 얻어질 수 있다.

네트워크가 소규모인 경우에는 통행 가능한 경로를 모두 나열하는 것이 가능하지만, 대규모인 경우에는 불가능하다. 따라서, 본 연구에서는 먼저 통행 가능한 경로를 모두 나열할 수 있는 소규모 네트워크(문제 1)에 대하여 제안된 알고리즘의 성능을 검증·평가하였으며, 또한 제안된 알고리즘에 사용되는 매개변수들의 적당한 값을 설정하였다. 설정된 매개변수들의 값을 이용해서 제안된 알고리즘이 중규모 네트워크(문제 2)와 대규모 네트워크(문제 3)에 적용되었으며 우수성을 평가하기 위해 다목적 최단경로문제에 적용될 수 있는 그 외의 알고리

들들과 비교·분석하였다. 제안된 알고리즘과 비교분석을 위해 사용된 알고리즘은 모두 Visual C++ 6.0을 사용하여 구현하였으며, Pentium 200MHz CPU, 64MB RAM를 장착한 IBM-PC에서 수행되었다.

6.1 성능평가 및 매개변수 설정

5.2절의 <그림 3>에 예시된 격자형 네트워크(문제 1)를 이용하여 알고리즘을 검증하였다. 총 통행시간 z_1 을 최소화하는 경로는 $p=1-3-5-7-14-21$ 이고 대응되는 z_1^* 는 14이다. 총 통행비용 z_2 를 최소화하는 경로는 $p=1-4-10-13-20-24$ 이고 대응되는 z_2^* 는 10이다. 이들 각 경로는 2절 <그림 1>의 목적함수 영역에서 파레토 최적경계선상에 있는 양 극점인, $(z_1^*, z_2^*)=(14, 26)$, $(z_1, z_2)=(28, 10)$ 에 대응된다. 이때 점 $(z_1^*, z_2^*)=(14, 10)$ 은 <그림 1>의 목적함수 영역에서 이상적인 해 I 에 대응된다.

네트워크에 존재하는 모든 경로는 20가지이며 이들 경로를 모두 열거하여 비교함으로써 쉽게 파레토 최적경로를 구할 수 있다. 양 극점에 대응되는 두 경로를 포함해서 5개의 파레토 최적경로가 얻어지며 얻어진 파레토 최적경로와 대응되는 목적값은 <표 1>과 같다. 위 문제에 유전자 알고리즘을 적용하기에 앞서서 교배확률(p_c), 돌연변이 확률(p_m), 최대 세대수(max_gen)를 실험적으로 각각 $p_c=0.5$, $p_m=0.05$, max_gen=300으로 설정하였다. 본 연구에서 제시된 1점 가중합 교배의 가중치 α 는 알고리즘의 성능에 커다란 영향을 미치며, 그 값은 모집단의 크기(population_size)에 영향을 받는 것으로 나타났다.

<표 2>는 population_size를 비교적 작게 20~30으로 설정하고, α 값을 0.1, 0.2, ..., 0.9의 9단계로 변화시켜 알고리즘을

표 1. 파레토 최적경로와 목적값 $Z=(z_1, z_2)$

파레토 최적경로	$Z=(z_1, z_2)$
$p_1=1-3-5-7-14-21$	(14, 26)
$p_2=1-4-11-17-19-21$	(15, 13)
$p_3=1-4-10-13-19-21$	(18, 12)
$p_4=1-4-11-17-20-24$	(25, 11)
$p_5=1-4-10-13-20-24$	(28, 10)

표 2. α 값의 변화에 따른 파레토 최적해의 집합 $E(t)$

α	파레토 최적해의 집합 $E(t)$					
	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6
0.1	(15, 13) (18, 12)					
0.2	(15, 24) (16, 18) (17, 14) (18, 12) (25, 11) (28, 10)					
0.3	(15, 24) (16, 18) (17, 14) (18, 12) (25, 11) (28, 10)					
0.4 ~	(14, 26) (15, 13) (18, 12) (25, 11) (28, 10)					

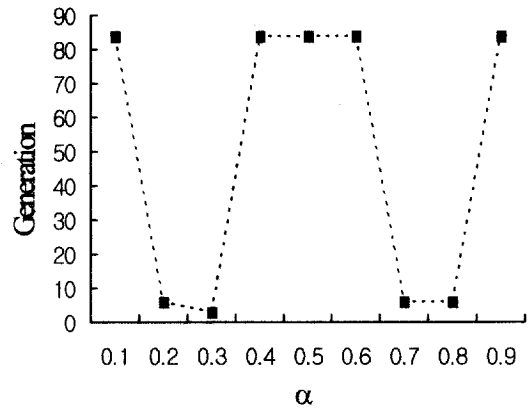


그림 6. α 값의 변화에 따른 수렴정도.

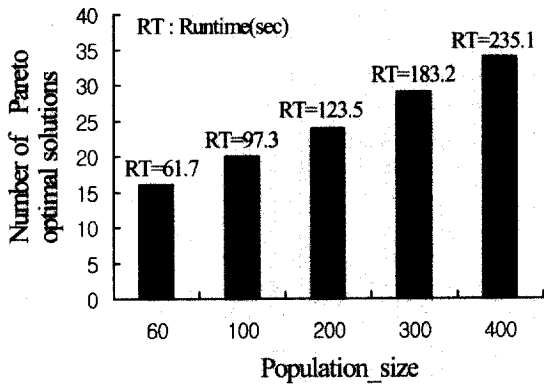
300세대만큼 수행한 결과이다. $\alpha=0.1$ 인 경우에 알고리즘은 2개의 파레토 최적해만을 탐색하고 조기에 수렴하였다. α 값이 0.2 또는 0.3인 경우는 파레토 최적해인 (15, 13), (14, 26)을 못 찾고 비 파레토 최적해인 (15, 24), (16, 18), (17, 14)을 찾아서 조기에 수렴하였다. α 값이 0.4이상인 경우에는 270~300세대에서 모든 파레토 최적해를 탐색하였다. population_size를 비교적 크게 50~60으로 설정한 경우에는 α 값에 관계없이 모든 파레토 최적해를 탐색하였다. <그림 6>은 population_size=60에 대하여 α 값의 변화에 따른 알고리즘의 수렴정도를 나타낸다. $\alpha=0.2, 0.3, 0.7, 0.8$ 인 경우에는 파레토 최적해를 조기에 탐색하고 수렴하는 경향이 있다. 따라서 모집단의 크기와 관계없이 $\alpha=0.4\sim 0.6, 0.9$ 에서 좋은 결과를 보여주는 것을 알 수 있다.

6.2 비교분석

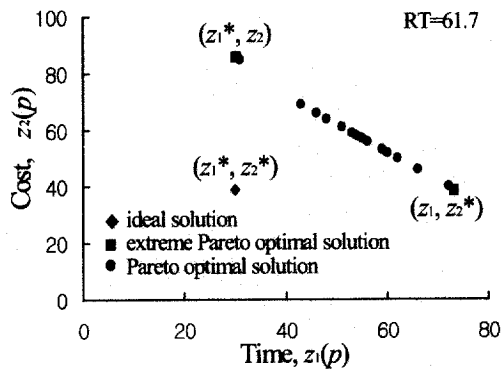
제안된 알고리즘의 우수성을 K 최단경로 알고리즘을 이용하는 Climaco and Martins의 방법과 Schaffer의 병렬탐색전략에 의한 GA와 비교분석하였다. 비교분석을 위해 사용된 예제는 64개의 노드와 112개의 링크로 구성된 8×8 의 격자형 네트워크(문제 2)와 400개의 노드와 760개의 링크로 구성된 20×20 의 격자형 네트워크(문제 3)이다. 각 링크의 통행시간 및 통행비용은 1과 10사이의 임의의 값으로 발생시켰다. 앞에서의 결과를 배경으로 $\alpha=0.5$ 로 설정하였으며, 그 외의 매개변수는 앞에서와 동일하게 $p_c=0.5$, $p_m=0.05$, max_gen=300으로 설정하여 수행하였다.

<그림 7>은 8×8 의 중규모 네트워크에 대하여 제안된 알고리즘을 적용한 결과이다. <그림 7(a)>로부터 모집단의 크기가 증가함에 따라 파레토 최적해의 수는 증가하지만 계산시간은 비례적으로 증가하는 것을 알 수 있다. <그림 7(b)>로부터 비교적 모집단의 크기를 작게 설정한 경우에도 양 극점의 파레토 최적해에 넓게 분포하는 다양한 파레토 최적해들이 얻어지는 것을 알 수 있다. 특히 모집단의 크기를 400으로 설정하여 얻어진 파레토 최적해들은 여행자를 위한 정보로서 충분하다

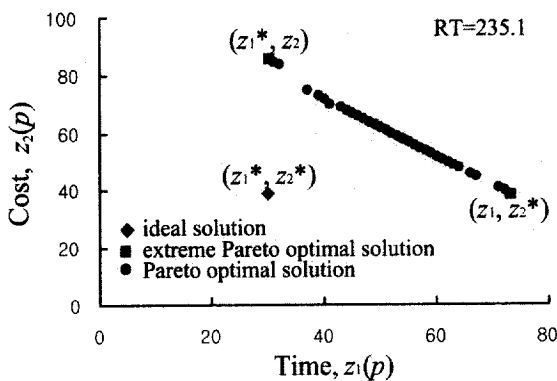
(<그림 7(c)>). <그림 8>은 문제 2에 대하여 본 연구에서 제안된 혼합전략 대신에 Schaffer의 병렬탐색전략을 이용한 경우의 GA 결과를 보여준다. 알고리즘의 수행시간에는 별차이가 없으나 양 극점에 치우친 파레토 최적해들을 조기에 탐색하고 수렴한다. 따라서 다양성이 상실되었으며 여행자를 위한 정보로서 충분하지 못하다.



(a) 모집단크기의 변화에 따른 파레토 최적해의 분포



(b) population_size = 60



(c) population_size = 400

그림 7. 문제 2의 제안된 알고리즘에 의한 결과.

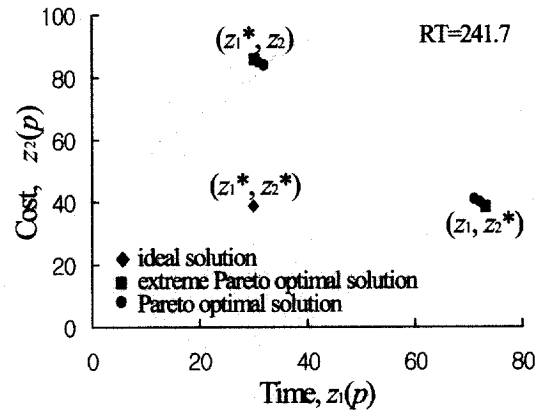


그림 8. 문제 2의 병렬탐색전략에 의한 GA 결과. (population_size = 400)

표 3. K 최단경로 알고리즘의 적용결과

문제 번호	파레토 최적해의 수			계산시간(초)		
	k=10	k=100	k=200	k=10	k=100	k=200
2	3	7	11	13.3	141.2	204.8
3	2	5	9	23.7	193.3	332.1

<표 3>은 문제 2와 문제 3에 대하여 K 최단경로 알고리즘을 이용하는 Climaco and Martins의 방법을 k=10, 100, 200으로 설정해서 적용한 결과를 보여준다. k 값이 커질수록 파레토 최적해의 수는 증가하지만, 매우 작고 극점(z1, z2)에 치우친 파레토 최적해들이 얻어지므로 정보로서의 다양성이 상실되었다. <그림 9>는 문제 2에 대하여 k=200으로 설정한 경우에 얻어진 파레토 최적해들을 나타내고 있다. 비슷한 계산시간동안 제안된 알고리즘을 수행시킨 결과인 <그림 7(c)>와의 비교로부터 본 알고리즘이 우수하다는 것을 알 수 있다. 문제 2에서 기점과 종점을 잇는 가능한 경로들의 수가 3432개가 존재한다. 따라서 양 극점에 넓게 분포하는 다양한 파레토 최적해들

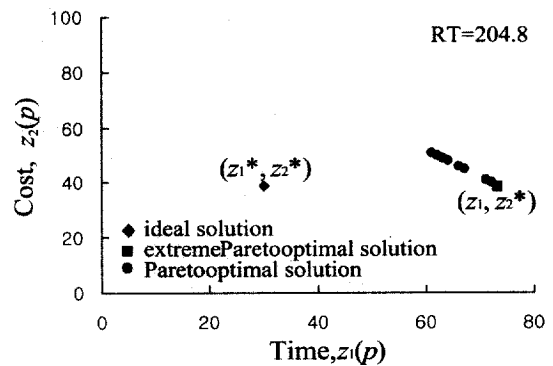


그림 9. 문제 2의 K 최단경로 알고리즘에 의한 파레토 최적해(k=200).

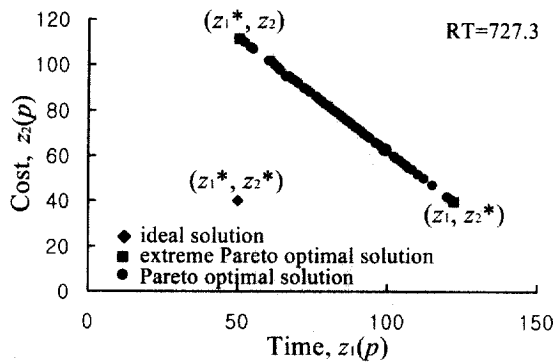


그림 10. 문제 3의 제안된 알고리즘에 의한 결과.
(population_size=400)

얻기 위해서는 이론적으로 k 의 값을 3432에 가깝게 설정하면 되지만 이것은 실제로 계산이 불가능하다. <그림 10>은 20×20 의 대규모 네트워크에 대하여 제안된 알고리즘을 적용한 결과이다. 양 극점의 파레토 최적해에 넓게 분포하는 다양한 파레토 최적해가 비교적 빠른 시간 안에 얻어지는 것을 알 수 있다. 따라서 본 알고리즘은 현실성이 반영된 대규모의 네트워크에서도 다양한 경로정보를 효율적으로 여행자에게 제공하기에 유용하다고 할 수 있다.

7. 결론

본 연구에서는 A^* 알고리즘과 유전자 알고리즘을 이용하여 다목적 최단경로문제의 다양한 파레토 최적해들을 탐색하기 위한 기법을 개발하였다. 기존의 방법들은 목적들의 어느 한쪽에 치우친 하나 또는 일부의 파레토 최적해만을 탐색하기 때문에 경로에 대한 다양한 정보를 제공하지 못한다. 본 연구에서 개발된 알고리즘의 성능을 소규모 네트워크에 대하여 평가한 결과 존재하는 파레토 최적해 전부를 탐색하는 것으로 나타났다. 또한 중규모 네트워크와 대규모 네트워크에 대하여 본 알고리즘과 기존의 알고리즘을 비교분석한 결과 본 알고리즘의 효율성 및 우수성이 입증되었다. 제안된 알고리즘에 의해 얻어진 다양한 파레토 최적경로를 이용하면 도로망을 효율적으로 이용하고 여행자의 편리성을 향상시킬 수 있다. 따라서 본 알고리즘은 현실성이 반영된 다목적 환경 하에서 다양한 경로정보를 요구하는 장래의 ATIS분석에 유용하게 활용될 수 있다.

참고문헌

- 김익기 (1998), ATIS를 위한 수정형 덩굴망 최단경로 탐색알고리즘의 개발, *대한교통학회지*, 16(2), 157-167.
- 이승환, 최기주, 김원길 (1996), 도시부 ATIS 효율적 적용을 위한 탐색 영역기법 및 양방향 링크탐색 알고리즘의 구현, *대한교통학회지*, 14(3), 45-59.
- 장인성, 김현철 (1999), 대중교통망 최단경로 탐색을 위한 유전 알고리즘, *춘계 한국보존공학회 학술논문집*.
- 장인성, 이승재 (1999), 다목적 최단경로 선정을 위한 유전알고리즘의 개발, *추계 산업공학회 학술논문집*.
- Climaco, J. C. N. and Martins, E. Q. V. (1982), A bicriterion shortest path algorithm, *European Journal of Operational Research*, 11, 399-404.
- Dijkstra, E. W. (1959), A Note on Two Problems in Connection with Graphs, *Numerische Mathematik*, 1, 269-271.
- Dossey, J., Otto, A., Spence, L. and Eynden, C. (1993), *Discrete Mathematics*, Harper Collins.
- Hansen, P. (1980), Bicriterion path problems, in: G. Fandel and T. Gal (eds.), *Multiple Criterion Decision Making: Theory and Applications*, Lecture Notes in Economics and Mathematical Systems 177, Springer, 109-127.
- Horn, J., Nafpliotis, N. and Goldberg, D. E. (1994), A niched Pareto genetic algorithm for multiobjective optimization, *Proceedings of the First IEEE Conference on Evolutionary Computation*, 82-87.
- Kim, Y. K., Kim, Y. J. and Kim, Y. H. (1996), Genetic algorithms for assembly line balancing with various objectives, *Computers & Industrial Engineering* 30, 397-409.
- Lawler, E. L. (1972), A procedure for computing the K best solutions to discrete optimization problems and its application to the shortest path problem, *Management Science* 18, 401-405.
- Michalewicz, Z. (1994), *Genetic Algorithm + Data Structure = Evolution Programs*, Springer-Verlag.
- Vignaux, G. A. and Michalewicz, Z. (1991), A genetic algorithm for the linear transportation problem, *IEEE Transaction on Systems, Man, and Cybernetics* 21, 445-452.
- Murata, T., Ishibuchi, H. and Tanaka, H. (1996), Multi-objective genetic algorithm and its applications to flowshop scheduling, *Computers & Industrial Engineering* 30, 957-968.
- Osyczka, A. and Kundu, S. (1996), A modified distance method for multi-criteria optimization using genetic algorithms, *Computers & Industrial Engineering* 30, 871-882.
- Shaffer, J. D. (1985), Multiple objective optimization with vector evaluated genetic algorithms, *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, 93-100.
- Steuer, R. E. (1986), *Multiple Criteria Optimization: Theory, Computation and Application*, John Wiley & Sons.
- Winston, P. H. (1992), *Artificial Intelligence*, Addison-Wesley.
- Zhou, G. and Gen, M. (1999), Genetic algorithm approach on multi-criteria minimum spanning tree problem, *European Journal of Operation Research*, 114, 141-152.