

네트워크 분석을 위한 유전 알고리즘 기반 경로탐색 시스템

김 준 우*

Genetic Algorithm based Pathfinding System for Analyzing Networks

Jun-Woo Kim *

요 약

본 논문은 다양한 네트워크를 편리하게 분석할 수 있는 실용적인 유전 알고리즘 기반 경로탐색 시스템인 GAPS를 제안하고자 한다. 이러한 목적을 위해 GAPS는 네트워크 모델링을 위한 직관적인 그래픽 사용자 인터페이스와 모델링 및 탐색 과정에서 발생하는 데이터들을 관리하기 위한 데이터베이스 관리 시스템, 다양한 네트워크를 분석하기 위해 개발된 간단한 유전 알고리즘을 결합하여 개발되었다. 특히, 기존의 유전 알고리즘들이 단락이 많고 두 개 노드 간 실행가능 경로 수가 많지 않은 네트워크를 분석하는데 적합하지 않았던 반면, GAPS는 실행가능 경로와 실행불가능 경로를 모두 적절히 평가할 수 있는 적합도 함수를 사용하는 유전 알고리즘에 기반하고 있어 해 집단의 다양성을 유지하면서 다양한 네트워크들을 분석할 수 있다. 실험 결과, GAPS를 통해 단락이 많은 네트워크와 단락이 적은 네트워크를 모두 편리하게 분석할 수 있다는 점과, GAPS가 기존의 경로탐색문제를 위한 유전 알고리즘들과 대비되는 장점을 갖고 있음을 확인할 수 있었다.

▶ Keywords : 경로 계획, 경로 탐사, 미로 통과, 유전 알고리즘, 데이터베이스 응용

Abstract

This paper proposes GAPS, a practical genetic algorithm based pathfinding system for conveniently analyzing various networks. To this end, the GAPS is developed through integration of the intuitive graphic user interface for network modeling, the database management system for managing the data generated in modeling and exploring procedures, and a simple genetic algorithm for analyzing a wide range of networks. Especially, previous genetic algorithms are not appropriate for analyzing the networks with many dead-ends where there are few feasible paths between the given two nodes, however, GAPS is based on the genetic algorithm with the fitness function

•제1저자 : 김준우 •교신저자 : 김준우

•투고일 : 2013. 10. 21, 심사일 : 2013. 11. 12, 게재확정일 : 2013. 12. 06.

* 동아대학교 산업경영공학과(Dept. of Industrial and Management Systems Engineering, Dong-A University)

appropriate for evaluating both feasible and infeasible paths, which enables GAPS to analyze a wide range of networks while maintaining the diversity of the population. The experiment results reveal that GAPS can be used to analyze both networks with many dead-ends and networks with few dead-ends conveniently, and GAPS has several advantages over the previous genetic algorithms for pathfinding problems.

▶ Keywords : Path planning, Pathfinding, Maze passing, Genetic algorithm, Database application

I. 서 론

평면 상에서 주어진 두 개 지점을 연결하는 최적의 경로를 탐색하는 것을 흔히 경로계획(path planning) 또는 경로탐색(path finding) 문제라고 하며, 최적의 경로는 일반적으로 최단경로나 최소비용경로를 의미한다[1]. 아울러, 이러한 경로탐색 문제는 버스나 항공기와 같은 대중 교통수단들의 순회 경로 관리[2-4], 제조 및 생산 시스템에서의 작업물 흐름 관리[5], 이동통신 시스템에서의 트래픽 라우팅 관리[6-8], 모바일 로봇 또는 무인 이송 장치의 관리[9-12] 및 기타 여러 가지 경로탐색 서비스[13] 등 다양한 분야에 응용할 수 있어 활발히 연구되어져 왔다.

전통적으로 경로탐색 문제의 분석 대상이 되는 평면 상의 공간은 노드(node)와 아크(arc)로 구성된 네트워크의 형태로 모델링되는 경우가 많았으며, 이 경우 경로탐색 문제의 최적해는 주어진 시작 노드에서 출발하여 종료 노드에 도달할 수 있는 연속된 아크들로 이루어진다[14]. 이러한 접근은 문제를 단순화시켜 풀이를 용이하게 만드는 장점이 있지만, 모델링된 네트워크가 실제 공간의 모습과 다소 다르고, 적절한 모델링 절차가 필요하다. 반면, 실제 공간을 격자[15]로 나타내거나, 실제 공간의 이미지 자체를 분석하여 최적의 경로를 찾는 방법[16]들도 연구되고 있으나, 이러한 방법들의 경우 문제 풀이 절차가 복잡해질 수 있어, 아직까지는 네트워크 형태의 접근이 많이 사용되는 것으로 생각된다.

한편, 네트워크의 두 개 노드 간 최적의 경로를 탐색하는 것은 전형적인 조합최적화(combinatorial optimization) 문제 중의 하나로 최적해를 다항시간(polynomial time) 안에 구하는 것이 곤란함이 잘 알려져 있고, 이를 해결하기 위

한 방법으로는 A-star, Dijkstra, Bellman-form 알고리즘이나 tabu search, simulated annealing 등과 같은 메타 휴리스틱 기법들이 연구되어져 왔다[14][16-17]. 유전 알고리즘(genetic algorithm)은 대표적인 확률적 탐색 기법(stochastic search method)의 하나로, 탐색의 효율성과 다양한 대체경로 탐색의 용이성으로 인해 경로탐색문제의 해법으로 널리 적용되어 왔다[4]. 하지만 유전 알고리즘을 네트워크의 두 개 노드 간 경로탐색문제에 적용하는 데에는 다음과 같은 문제점들이 여전히 존재한다. 첫째, 기존의 유전 알고리즘의 경우 특정 형태의 네트워크에만 적합한 경우가 많아 다양한 문제 상황을 다루는데 한계가 존재한다. 둘째로는 네트워크를 구성하는 노드와 아크 등을 모델링하는 절차가 불편할 수 있다는 점을 들 수 있다. 끝으로, 기존의 연구들은 보통 문제 풀이를 위한 유전 연산자(genetic operator)들의 설계에 집중하고, 상대적으로 네트워크 모델링부터 경로탐색 과정과 결과 조회 등을 통합적으로 지원하는 편리한 분석 시스템의 설계 및 구현은 다루어지지 못하였다.

이러한 점들을 보완하기 위하여 본 논문은 네트워크의 특정 두 개 노드 간 최적의 경로탐색을 편리하게 실시할 수 있는 유전 알고리즘 기반 경로탐색 시스템(GAPS, GA based Pathfinding System)을 제안하고자 한다. 제안하는 시스템은 적절한 GUI를 제공하여 사용자가 분석 대상 네트워크를 편리하게 관찰 및 설계할 수 있으며, 입력된 네트워크 및 경로탐색 과정에서 발생하는 데이터들을 데이터베이스에 저장하여 전체 분석 과정을 효과적으로 관리할 수 있다. 아울러, 본 논문에서는 다양한 네트워크를 분석할 수 있는 간단한 유전 알고리즘을 제안하여, GAPS의 활용성을 높이고자 하였다.

본 논문의 2장에서는 경로탐색 문제에 적용되어온 유전 알고리즘들에 대한 기존 문헌에 대해 소개하고, 3장에서는 제안하는 유전 알고리즘의 세부 요소 및 GAPS의 구조와 기능에

대해 설명한다. 이어 4장에서는 GAPS를 경로탐색 문제에 적용한 실험 결과를 소개하며, 끝으로 5장에서 결론 및 후후 과제를 제시하고자 한다.

II. 관련 연구

유전 알고리즘은 생태계의 진화 과정을 모방하여 개발된 메타 휴리스틱(meta heuristic) 탐색 기법 중의 하나로, 탐색 과정 중 일정한 크기의 해 집단(population)을 유지하면서 선택(selection), 교차(crossover), 돌연변이(mutation) 등의 유전 연산자를 통해 반복적으로 개선된 해를 탐색해나간다[18]. 특정 문제에 유전 알고리즘을 적용하기 위해서는 해의 표현(representation) 및 다양한 유전 연산자들이 적절히 설계되어야 하며, 적절히 설계된 유전 알고리즘은 다양한 조합최적화 문제에 적용되어 좋은 성과를 거두어왔음이 잘 알려져 있다[19].

네트워크의 두 개 노드 간 경로탐색 문제의 해는 본질적으로 시작 노드를 출발하여 도중에 다른 적당한 노드들을 경유한 후 정해진 종료 노드에 도착하는 하나의 경로를 형성하게 된다. 이러한 점을 반영하여 기존 연구들에서는 가변길이(variable length) 표현을 통해 경로탐색 문제의 해를 나타내는 방법이 사용되기도 하였다[2][20-21]. 이러한 표현 방법은 이산적인 항목들의 나열 순서를 결정하는 시퀀싱(sequencing) 문제를 위한 유전 알고리즘들에서 흔히 사용되는 순열 기반 표현(permutation based representation)[22]을 약간 변형한 것으로, 모든 항목들을 나열해야 하기 때문에 항목의 총 개수와 같은 길이를 갖는 순열 기반 표현과 달리, 경로탐색 문제에서는 존재하는 노드 중 일부만을 경유하여 경로가 만들어지기 때문에 각 해의 표현 길이에 차이가 생긴다. 나아가, 가변길이 표현을 사용하는 유전 알고리즘의 경우 시작 노드에서 종료 노드에 도달하는 실행가능한(feasible) 경로들을 개체로 하는 해 집단을 유지하면서 최적의 경로를 탐색해나간다. 하지만 이러한 유전 알고리즘들의 경우, 탐색 과정에서 실행불가능한 경로가 나오는 것을 방지하기 위해 복잡한 유전 연산자가 필요하다[4]. 또한, 이로 인해 공통의 중간 경유 노드를 갖는 부모 해 한 쌍에 대해서만 교차가 적용 가능하다거나, 일점 교차 방식만 가능한 점과 같은 제약이 존재한다[8]. 나아가 이렇게 실행가능 경로들의 해를 유지하는 방식은 경로탐색 문제 중, 막힌 곳, 즉, 단락(dead-end)이 많은 네트워크를 다루는 미로 통과(maze passing) 문제[23]와 같은 경우를 잘 다루지 못한다는 한계가 있다.

경로탐색 문제에서 자주 사용되는 또 다른 해 표현 방법은 노드 개수와 동일한 숫자의 유전자로 해 한 개를 표현하는 노드 설정(node configuration) 방식이다. 노드 설정 방법을 사용할 경우, 각 유전자는 노드 1개에 대응되며, 해당 노드 도착 후 다음으로 방문할 인접 노드를 표시하게 된다. 이러한 표현을 사용하는 유전 알고리즘들은 실행불가능한 경로가 해 집단에 포함되는 것을 허용하는 경우가 많으며, 탐색 과정에서 실행가능한 경로 중 최적의 경로를 찾아나간다[1][4]. 이러한 방법은 유전 연산자가 비교적 간단하고, 단락이 많은 네트워크나 네트워크가 아닌 격자형으로 주어지는 공간 상에서의 경로탐색 문제에도 손쉽게 적용할 수 있다는 장점이 있다[1][15][24-25]. 반면, 이러한 경우에는 실행불가능한 경로들이 해 집단에 포함되기 때문에 적합도(fitness) 함수가 주의깊게 설계되어야 한다. 그 이유는 일반적으로 실행가능한 경로들의 경우 가급적 총 이동거리나 총 비용이 작은 경우 높은 적합도를 부여하는 반면, 실행불가능한 경로들의 경우 많은 탐색을 유도하기 위해 순환이나 단락을 만나 탐색이 중단되는 시점까지 발생한 총 이동거리나 총 비용이 큰 것에 높은 적합도를 부여할 필요가 있기 때문이다. 이를 위해 기본적으로 이동거리가 짧은 개체에 높은 적합도를 부여하되, 도착 노드까지 완주하지 못할 경우 적절한 벌점(penalty)을 부과하는 방법이 흔히 사용된다. 이 경우 이동거리가 긴 실행불가능 경로보다 이동거리가 짧은 실행가능 경로의 적합도가 높도록 적절히 파라미터를 설정해야 하는 경우가 많다. 이 외에도 각 노드의 우선 순위를 연산자로 사용하는 방법[26-27] 등이 있으나, 이는 복잡한 해의 해독(decoding) 절차가 필요하다는 단점이 존재한다. 본 논문에서 제안하는 GAPS의 경우, 다양한 네트워크를 편리하게 분석하기 위하여, 노드 설정 방식에 기반한 유전 알고리즘을 사용하였으며, 사용자가 별도의 파라미터 설정을 할 필요가 없는 간단한 적합도 함수를 새로이 적용하였다.

한편, 노드, 노드 간 인접 여부 및 아크들의 길이 또는 비용 등 다양한 구성 요소를 갖는 경로탐색 문제와 같은 경우, 풀이 과정에서 이러한 정보들이 효과적으로 이용되어야 하며, 이를 위해 데이터베이스와 같은 수단을 이용하는 것이 바람직할 수 있다[4]. 아울러, 여러 가지 문제를 편리하게 분석하기 위해서는 사용하기 편리한 분석 시스템의 개발이 필요하다[28]. 이러한 측면들은 기존의 연구들에서 충분히 고려되지 않는 경우가 많았으나, 본 논문에서는 GAPS를 설계함에 있어 편리한 사용자 인터페이스와 탐색 과정을 효과적으로 관리할 수 있는 데이터베이스를 포함시켜, 실용적인 시스템을 개발하고자 하였다.

III. GAPS 구성 요소

1. 네트워크 설계를 위한 GUI

네트워크에 포함된 두 개 노드 간 경로탐색을 실시하기 위해서는 먼저, 주어진 네트워크를 시스템에 입력하는 것이 필요하다. 본 논문에서 제안하는 GAPS는 이러한 목적을 위해 그림 1과 같은 네트워크 모델러를 제공한다.

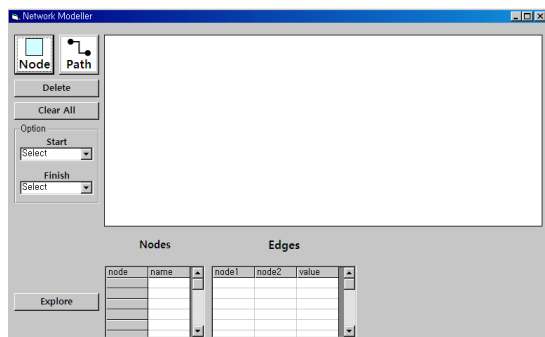


그림 1. 네트워크 모델러
Fig. 1. Network Modeller

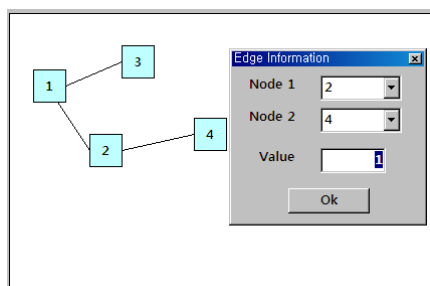


그림 2. 두 개 노드 간 아크 생성
Fig. 2. Creating an Arc between Two Nodes

네트워크 모델러의 우측에는 넓은 작업 영역이 있어, 분석 대상 네트워크를 시각적으로 묘사할 수 있으며, 좌측 상단의 Node 버튼과 Path 버튼은 각각 네트워크를 구성하는 노드와 아크를 작업 영역에 추가하는데 사용된다. 예를 들어, Node 버튼을 클릭한 상태에서 작업 영역의 한 곳을 클릭할 경우, 해당 위치에 새로운 노드 1개가 추가되며, Path 버튼을 클릭한 상태에서 작업 영역의 두 개 노드를 차례로 클릭하면 해당 노드들을 연결하는 아크 1개가 생성된다.

아크를 추가할 경우에는 그림 2에서와 같이 팝업창이 하나 생성되면서 해당 아크의 길이 또는 비용 등을 의미하는 값을 입력하게 된다. 나아가, 추가된 노드 및 아크들에 대한 정보는 네트워크 모델러의 하단에 스프레드시트 형태의 목록으로 표시되며, 필요한 경우 목록에서 노드의 이름이나 아크의 값과 같은 세부 정보를 수정하는 것도 가능하다.

분석 대상 네트워크를 완성한 경우에는 좌측의 Option 패널에서 시작 노드 및 종료 노드를 설정할 수 있으며, 이상의 작업들이 완료된 경우에는 좌측 하단의 Explore 버튼을 클릭하여 명시된 노드들을 연결하는 최적의 경로 탐색을 시작한다.

2. GAPS 데이터베이스 구조

GAPS는 사용자가 입력한 내용 및 경로탐색 과정에서 발생하는 다양한 데이터를 효과적으로 관리하기 위하여 적절히 설계된 데이터베이스를 이용한다. 그림 3은 이러한 데이터의 전체 구조를 보여준다.

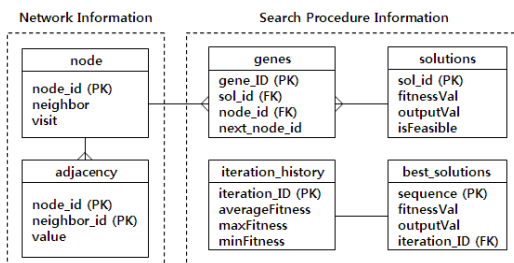


그림 3. GAPS 데이터 모델
Fig. 3. The Data Model for GAPS

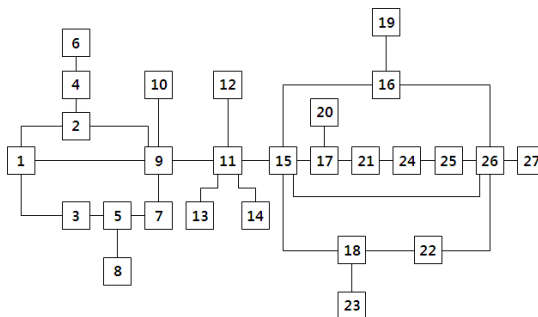


그림 4. 분석할 네트워크 예시
Fig. 4. An Example of a Network to be Analyzed

GAPS의 데이터 모델은 크게 사용자가 생성한 네트워크 구조를 관리하는 부분과 유전 알고리즘에 의한 최적의 경로 탐색 과정을 관리하는 부분으로 나누어진다. 먼저, node 데이

블과 adjacency 테이블은 네트워크 구조를 관리하는 부분으로 사용자가 네트워크 모델러에서 explore 버튼을 클릭한 직후 레코드가 삽입이다. 이 때, node 테이블에는 각 노드의 인접 노드 개수가 함께 저장되며, adjacency 테이블에는 한 노드의 인접 노드 각각에 대하여 레코드를 생성하면서 해당 아크의 거리나 비용 등과 같은 값을 함께 저장한다. 예를 들어, 그림 4의 네트워크를 등록할 경우, node 테이블과 adjacency 테이블에는 각각 표 1, 표 2와 같은 레코드가 삽입된다. 단, 표 1에서는 모든 아크의 값을 1로 하였고, 이는 시작 노드를 출발하여 가장 적은 개수의 노드를 경유하면서 종료 노드에 도달하는 경로를 찾는다는 것을 의미한다. 또한, 표 2의 N.id, Ne.id는 각각 node_id와 neighbor_id를 나타낸다.

이러한 정보들은 모두 네트워크 모델러에서 쉽게 구할 수 있으며, 네트워크 모델러에서 등록된 노드 한 개는 표 1에서 볼 수 있듯이 한 개 레코드로 저장되고, 아크 한 개는 표 2에서와 같이 두 개의 레코드로 저장된다. 예를 들어, 그림 4에서 노드1과 노드2를 연결하는 아크의 경우, 표 2의 첫 번째, 네 번째 레코드에 해당한다.

표 1. Node 테이블에 삽입된 레코드 예시
Table 1. An Example of the Records inserted into Table 'Node'

node_id	neighbor	visit	node_id	neighbor	visit
1	3	0	15	5	0
2	3	0	16	3	0
3	2	0	17	3	0
4	2	0	18	3	0
5	3	0	19	1	0
6	1	0	20	1	0
7	2	0	21	2	0
8	1	0	22	2	0
9	5	0	23	1	0
10	1	0	24	2	0
11	5	0	25	2	0
12	1	0	26	5	0
13	1	0	27	1	0
14	1	0			

Node 테이블이 가진 필드 중, visit는 나머지 테이블들과 함께 유전 알고리즘에 의한 경로탐색 과정에서 사용된다. 이들에 대해서는 다음 절에서 설명하기로 한다.

3. 경로탐색을 위한 유전 알고리즘

3.1 해의 표현

본 논문에서 제안하는 GAPS는 주어진 네트워크를 분석하

여 최적의 경로를 탐색하기 위하여 유전 알고리즘을 사용하며, 이러한 문제 풀이에 자주 활용되는 표현 방법인 노드 설정 표현을 채택하였다. 총 n 개의 노드가 존재할 경우, 이 표현 방법은 n 개의 유전자 G_1, G_2, \dots, G_n 을 이용하여 해 한 개를 나타내며, 유전자 G_i 는 노드 $i(1 \leq i \leq n)$ 에 대응된다. G_i 의 값은 노드 i 의 인접 노드들의 집합인 NE_i 의 원소들 중 하나로 제한되며, 특정 노드의 인접 노드 목록은 표 2와 같이 adjacency 테이블에서 얻을 수 있다.

노드 설정 방식으로 표현된 해는 시작 노드에 해당하는 유전자에서 출발하여 매번 현재 유전자 값이 의미하는 다음 노드로 이동하는 경로를 생성함으로써 해독될 수 있다. 아울러, 해독 시에는 순차적으로 경로를 순회하다 기존에 방문했던 노드를 재방문하는 순환이 발생하는 해의 경우, 순환으로 인해 도착 노드에 도달할 수 없는 실행가능 경로가 되고, 순환이 없이 도착 노드에 도달하는 것은 실행가능 경로가 된다. 예를 들어, 그림 5는 그림 4의 네트워크에 대한 해 하나를 노드 설정 방식으로 표현한 것으로, $1 \rightarrow 2 \rightarrow 9 \rightarrow 7 \rightarrow 5 \rightarrow 8 \rightarrow 5$ 순으로 노드들을 순회하다가 마지막에 기존에 이미 한 번 방문했던 노드 5를 재방문하면서 종료되는 실행가능 경로를 의미한다.

표 2. Adjacency 테이블에 삽입된 레코드 예시
Table 2. An Example of the Records inserted into Table 'Adjacency'

N.id	Ne.id	value	N.id	Ne.id	value	N.id	Ne.id	value
1	2	1	9	11	1	18	15	1
1	3	1	10	9	1	18	22	1
1	9	1	11	9	1	18	23	1
2	1	1	11	12	1	19	16	1
2	4	1	11	13	1	20	17	1
2	9	1	11	14	1	21	17	1
3	1	1	11	15	1	21	24	1
3	5	1	12	11	1	22	18	1
4	2	1	13	11	1	22	26	1
4	6	1	14	11	1	23	18	1
5	3	1	15	11	1	24	21	1
5	7	1	15	16	1	24	25	1
5	8	1	15	17	1	25	24	1
6	4	1	15	18	1	25	26	1
7	5	1	15	26	1	26	15	1
7	9	1	16	15	1	26	16	1
8	5	1	16	19	1	26	22	1
9	1	1	16	26	1	26	25	1
9	2	1	17	15	1	26	27	1
9	7	1	17	20	1	27	26	1
9	10	1	17	21	1			

2	9	1	6	8	4	5	7	9	1	2	1	1	1	1	6	1	9	2	0	2	3	1	6	1	7	2	4	2	6	1	8	2	5	2	4	2	7	2
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

그림 5. 노드 설정 표현 방법
Fig. 5. Node Configuration Representation

3.2 적합도 함수

적합도 함수는 유전 알고리즘의 탐색 과정을 강건하게 만들기 위하여 주의 깊게 설계되어야 한다. 특히, 노드 설정 방식의 표현을 사용할 경우에는 실행불가능한 경로와 실행가능 경로를 적절히 구분하여 적합도를 평가하는 것이 중요하다. 일반적으로 경로탐색 문제에서는 총 이동거리(총 이동비용)를 최소화하는 것이 목표이므로, 기존의 유전 알고리즘들의 적합도 함수는 일반적으로 (1)과 같이 총 이동거리의 역수 형태를 기본으로 하였다. 단, V_p 는 해 p 가 시작 노드를 출발한 후, 순환 또는 종료 노드에 도달하여 순회를 마치기까지 발생한 총 이동거리를 의미하며, A 는 분석자에 의해 부여되는 상수이다.

$$fitness(p) = \frac{A}{V_p} \tag{1}$$

이러한 형태의 적합도 함수는 실행가능 경로 중 짧은 총 이동거리를 갖는 것이 높은 적합도를 갖는다는데 의미가 있으며, 가변길이 표현을 이용하여 실행가능 경로들의 해 집단을 유지하는 유전 알고리즘의 경우에는 좋은 결과를 얻을 수 있다. 반면, 그림 4와 같이 단락이 많은 미로 형태의 네트워크에서는 실행가능 경로의 개수가 많지 않아, 이들의 해 집단을 유지한다는 것이 극히 비효율적이 된다.

따라서, 단락의 개수에 무관하게 다양한 네트워크를 다루기 위해서는 실행가능 경로와 실행불가능 경로를 모두 포함하는 해 집단을 유지하는 것이 바람직하나, 이 경우에는 (1)의 적합도는 실행불가능 경로의 적합도가 실행가능 경로보다 더 높게 나올 수 있다는 문제가 발생한다. 예를 들어, 그림 4에서 노드 1을 출발하여 노드 27에 도달하고자 하는 경우, 실행가능 경로인 1→9→11→15→18→22→26→27은 총 이동거리가 7인 반면, 단락을 만나는 실행불가능 경로 1→2→4→6의 총 이동거리는 3으로 짧기 때문에 더 높은 적합도를 갖게 된다. 이러한 점을 방지하기 위하여 기존 연구들에서는 (2)와 같은 형태의 적합도 함수가 사용되기도 하였다.

$$fitness(p) = \frac{A}{V_p} - B \tag{2}$$

(2)는 기존의 적합도 (1)에서 벌점 항인 B 를 추가한 형태로, 실행가능 경로의 경우 $B=0$ 이지만, 실행불가능 경로에 대해서는 $B>0$ 이 되어, 실행불가능 경로의 적합도는 종전보다 감소하게 된다. 그러나 이 경우에는 항상 0이상의 적합도가 산출됨과 동시에, 실행가능 경로가 실행불가능 경로보다 높은 적합도를 갖도록 상수 A 와 B 의 값을 분석자가 적절히 조정해야 하고, 일반 사용자들이 이들을 결정하기 어려울 수 있다.

한편, GAPS에는 이러한 점을 보완할 수 있는 새로운 적합도 함수를 적용하였다. 제한하는 적합도 함수는 실행가능 경로의 경우, 기존의 연구들에서와 같이 총 이동거리가 짧은 것에 높은 적합도가 부여되어야 하는 반면, 실행불가능한 경로들은 총 이동거리가 긴 것에 높은 적합도를 부여하도록 설계하였다. 이는 실행불가능 경로들의 경우 초기에 순환을 만나 탐색이 종료되기보다 가급적이면 시작 노드에서 출발하여 다양한 노드들을 많이 거칠수록 종료 노드를 찾을 가능성이 커질 것이기 때문이다. 이러한 점을 반영하여, 본 논문에서는 크기가 N_p 인 모집단에 포함된 해 p 의 적합도 $fitness(p)$ 를 (3)과 같이 계산한다. 단, V_p 는 등록된 모든 아크들의 거리 총합을 의미한다.

(3)에서 실행가능한 경로와 실행불가능한 경로의 적합도는 서로 역수의 관계를 갖고 있음을 볼 수 있는데, 임의의 네트워크의 경로 p 에 대하여 항상 $V_p \leq V_{total}$ 의 관계가 성립한다. 따라서, 경로 p 가 실행가능한 경우, 항상 $fitness(p) \geq 1$ 이 되고, 실행불가능한 경우에는 $fitness(p) \leq 1$ 이 되어 별도의 파라미터 설정 없이도 실행가능한 경로들의 적합도가 실행불가능한 경로들보다 크다는 점이 보장된다. 아울러, 실행가능한 경로의 적합도는 V_p 가 작을수록 높고, 그렇지 않은 경우 V_p 가 낮을수록 높게 설계하여, 세대교체를 통해 점차 실행가능한 경로 중 총 이동거리가 작은 것을 탐색할 수 있도록 하였다.

$$fitness(p) = \begin{cases} \frac{V_{total}}{V_p}, & \text{if solution } p \text{ is feasible} \\ \frac{V_p}{V_{total}}, & \text{otherwise} \end{cases} \tag{3}$$

3.3 유전 연산자

GAPS는 탐색 과정에서 필요한 정보들을 모두 데이터베이스를 통해 관리한다. 먼저, 그림 3의 genes 테이블은 특정 시

점에서 해 집단을 구성하는 해들의 각 유전자 정보를 레코드로 저장하며, sol_id 필드는 각 해의 번호 p 를, node_id 필드는 각 유전자에 대응되는 노드의 번호 i 를 나타낸다. 또한, 각 노드에서 다음에 방문할 인접 노드의 번호가 next_node_id 필드에 저장된다. 이에 따라, 한 세대의 해를 저장하기 위해 총 $n \times N_p$ 개의 레코드가 필요하며, 해 집단 초기화 단계에서는 유전자 NE_i 의 원소 중 임의의 한 개를 골라 G_i 의 값으로 설정하게 된다. 이러한 해 집단 초기화 절차는 표 3에 요약되어 있다.

초기 해 집단 생성 이후에는 선택, 교차 및 돌연변이와 같은 유전 연산자들을 반복적으로 수행하여 최적의 경로를 탐색하게 된다. GAPS의 유전 알고리즘에서는 일반적으로 사용되는 룰렛바퀴(roulette wheel) 방식의 선택 연산을 채택하되, 이전 세대의 해 중 가장 우수한 소수의 개체들을 다음 세대에 반드시 생존시키는 엘리트 우대(elitism)를 함께 적용하였다.

선택 연산을 통해 생존한 해들 중 일부는 교차 연산에 참여하게 되고, 이 때 참여자들의 비율은 미리 정의된 교차율 r_c ($0 \leq r_c \leq 1$)과 같다. 교차 연산은 이들 참여자 중 임의로 선정된 두 개의 부모 해 p_1, p_2 의 유전자를 조합하여 새로운 후손 해 2개를 생성한 후, 이들 후손 해들로 기존의 부모 해를 대체한다. 표 4는 일양 교차(uniform crossover) 방식으로 구현된 GAPS의 교차 연산 절차를 보여주며, 여기서 for 루프를 통해 후손 해를 genes 테이블에 추가한 후, 마지막에 기존 부모 해를 삭제하고 후손 해들이 그 자리를 대신하는 것을 볼 수 있다.

표 3. 해 집단 초기화 절차
Table 3. Population Initialization Procedure

```

For p = 1 To  $N_p$ 
  For i = 1 To  $n$ 
    set theNext = random element of  $NE_i$ 
    set gene_id = (p-1)×n + i
    insert into genes values(gene_id, p, i, theNext)
  Next i
Next p
    
```

부모 세대와 비슷한 형질을 후손 해들이 물려받도록 하는 교차 연산과 달리, 돌연변이 연산은 해의 다양성을 위해 부모 세대 형질과는 별개의 새로운 유전자를 해 집단에 발생시키는 데 사용된다. 표 5는 돌연변이율 r_m 이 주어졌을 때, GAPS의 돌연변이 연산 절차를 보여준다.

표 4. 교차 연산 절차
Table 4. Crossover Operation Procedure

```

set  $p_1$  = sol_id of parent1
set  $p_2$  = sol_id of parent2
set  $T = n \times N_p$ 

For i = 1 To  $n$ 
  set mark = random number between 0 and 1

  If mark <= 0.5 Then
    insert into genes values( $T+i, N_p+1, i, G_i$  of  $p_1$ )
    insert into genes values( $T+n+i, N_p+2, i, G_i$  of  $p_2$ )
  Else
    insert into genes values( $T+i, N_p+1, i, G_i$  of  $p_2$ )
    insert into genes values( $T+n+i, N_p+2, i, G_i$  of  $p_1$ )
  End If
Next i

delete from genes where sol_id= $p_1$  or sol_id= $p_2$ 
update genes set gene_id=gene_id-( $T-(p_1-1)n$ ), _
sol_id= $p_1$  where sol_id= $N_p+1$ 
update genes set gene_id=gene_id-( $T-(p_2-1)n$ ), _
sol_id= $p_2$  where sol_id= $N_p+2$ 
    
```

표 5. 돌연변이 연산 절차
Table 5. Mutation Operation Procedure

```

For p = 1 To  $N_p$ 
  For i = 1 To  $n$ 
    set mark = random number between 0 and 1

    If mark <=  $r_m$  Then
      set  $G_{new}$  = random element of  $NE_i - \{G_i$  of  $p\}$ 
      update genes set next_node_id= $G_{new}$  where _
      sol_id=p and node_id=i
    End If
  Next i
Next p
    
```

3.4 기타

기본적인 유전 알고리즘의 탐색 과정 외에도 GAPS는 분석자의 사용을 돕기 위한 부가적인 기능을 가지고 있다. 먼저, 그림 3의 solutions 테이블에는 한 세대를 거칠 때마다 해 집단에 포함된 각 해들을 (3)의 적합도에 의해 평가한 결과와 함께 각 해가 실행가능경로인지를 저장하며, iteration_history 테이블에는 진화 과정에서 생성되는 각 세대의 최소 적합도, 최대 적합도 및 평균 적합도 등을 기록하여, 분

석자가 이들의 변화 추이를 관찰할 수 있게 하였다. 끝으로 best_solutions 테이블에는 진화 과정에서 지금까지 탐색된 개체들보다 더 좋은 해가 등장할 때마다 각각이 몇 번째 세대에서 등장했는지를 기록하여, 최종적으로 찾은 해 뿐 아니라 탐색 과정에서 얻어진 다양한 해들을 관찰할 수 있게 하였다.

IV. 실험 결과

본 장에서는 GAPS를 몇 가지 경로탐색 문제에 적용한 결과를 소개한다. 먼저, 그림 4의 네트워크에서 모든 아크의 길이를 1로 가정하고, 노드 1을 출발하여 노드 27에 도달하는 최적의 경로, 즉, 최소의 경유 노드를 거치는 경로를 찾고자 하였다. 그림 4에서 볼 수 있듯이, 이 네트워크에는 노드 6, 8, 10, 12, 13, 14와 같은 여러 개의 단락이 존재하며, 노드 집합 {1, 2, 3, 5, 7, 9} 및 {15, 16, 17, 18, 20, 21, 22, 24, 25, 26} 들의 경우 순환 경로를 형성하고 있어 미로 통과 문제에 가까운 특성을 보인다. 특히, 실행가능한 경로의 개수가 적은 미로 통과 문제의 경우, 가변길이 표현 방법에 기반하여 실행가능 경로들의 해 집단을 유지하는 방식의 유전 알고리즘이 비효율적이며, 본 논문에서와 같이 노드 설정 방식이 적합하다. 아울러, 적합도 함수는 본 논문에서 제안하는 (3)을 사용하였다.

그림 4에서 노드 1에서 노드 27까지를 순회하기 위해서는 반드시 중간에 노드 15를 경유해야 하는데, 노드 1에서 15까지의 경로가 $1 \rightarrow 2 \rightarrow 9 \rightarrow 11 \rightarrow 15$, $1 \rightarrow 9 \rightarrow 11 \rightarrow 15$, $1 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 9 \rightarrow 11 \rightarrow 15$ 의 세 가지, 노드 15에서 노드 27까지의 경로가 $15 \rightarrow 16 \rightarrow 26 \rightarrow 27$, $15 \rightarrow 17 \rightarrow 21 \rightarrow 24 \rightarrow 25 \rightarrow 26 \rightarrow 27$, $15 \rightarrow 26 \rightarrow 27$, $15 \rightarrow 18 \rightarrow 22 \rightarrow 26 \rightarrow 27$ 의 네 가지가 있기 때문에 총 $3 \times 4 = 12$ 가지의 실행가능한 경로가 있음을 알 수 있다. 이 중 가장 적은 개수의 경유 노드를 거치는 최적의 경로는 $1 \rightarrow 9 \rightarrow 11 \rightarrow 15 \rightarrow 26 \rightarrow 27$ 이다. 나아가, 이 경로의 총 이동거리는 5이고, 현재 네트워크에서 $V_{total} = 31$ 이므로, 최적의 경로의 적합도 값은 $31/5 = 6.2$ 임을 알 수 있다.

그림 6과 7은 해 집단 크기 $N_p = 50$, 교차율 $r_c = 0.8$, 돌연변이율 $r_m = 0.01$, 반복 횟수 = 100을 사용하여 그림 4의 네트워크를 분석한 결과를 보여준다. 단, 그림 6의 경우 선택 연산 시 살아남는 이전 세대의 좋은 해 개수 $n_{elite} = 5$ 를 사용하여 얻은 결과이고 그림 7은 $n_{elite} = 0$ 으로 엘리트 우대를 적용하지 않고 얻은 결과이다. 그림 6의 경우, 세대 내 최대 적

합도가 비교적 초반(7세대)에 1을 초과하여, 실행가능 경로를 매우 효과적으로 탐색한다는 점을 볼 수 있다. 나아가, 초반에 얻어진 실행가능 경로보다 더 높은 적합도를 갖는 것들을 지속적으로 탐색하여 27세대에 최적의 경로(적합도 6.2)를 찾아내었다. 또한, 해 집단의 평균 적합도 역시 진화를 거듭하면서 꾸준히 상승하여 GAPS의 유전 알고리즘이 미로 형태의 네트워크에서 최적의 경로를 매우 잘 찾아낸다는 것을 볼 수 있다.

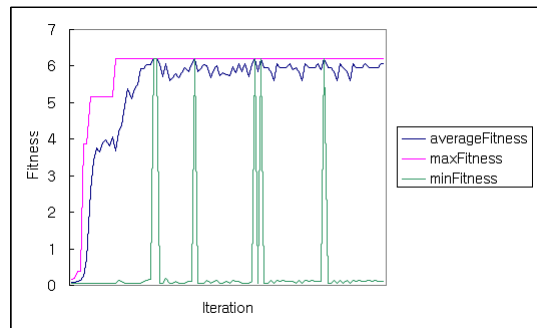


그림 6. $n_{elite} = 5$ 일 때의 실험 결과
Fig. 6. The Experiment Result under $n_{elite} = 5$

반면, 그림 6에서 세대 내 최소 적합도는 중간에 몇 번 큰 폭으로 상승했다 다시 떨어지는 모습을 보여주는데, 급격한 상승은 해 집단 내 모든 경로가 실행가능 경로가 되어 1을 초과하는 높은 적합도를 갖는 경우이다. 급격한 하강의 경우, 실행가능 경로 중 일부가 교차나 돌연변이를 통해 실행불가능한 경로를 후손으로 남게 되는 경우를 의미한다. 이는 노드 설정 방식으로 표현된 해가 유전 연산을 거치면서 갑자기 안 좋은 해로 변할 가능성이 높다는 점을 보여준다. 예를 들어, 그림 4의 네트워크에 대한 최적의 경로 $1 \rightarrow 9 \rightarrow 11 \rightarrow 15 \rightarrow 26 \rightarrow 27$ 에 해당하는 해의 경우, 교차나 돌연변이에 의해 G_9 의 값이 기존 11에서 10으로 바뀌기만 해도 실행불가능한 경로가 되면서 총 이동거리는 2가 되어 적합도가 기존의 6.2에서 $2/31 = 0.0645$ 로 급격히 감소하게 된다. 따라서, 이러한 점을 감안하여 노드 설정 표현을 사용할 경우에는 지금까지 탐색된 좋은 해들의 성질을 유지하기 위해 적절한 엘리트 우대를 사용하는 것이 중요한 것으로 보인다.

이러한 점은 엘리트 우대를 사용하지 않고 얻어진 실험 결과인 그림 7에서도 확인가능하다. 그림 7에서는 100번의 반복을 거치는 동안, 세대 내 최대 적합도, 최소 적합도 및 평균 적합도가 비교적 완만하게 상승하나, 탐색을 종료할 때까지 적합도 1을 초과하는 해, 즉, 실행가능한 경로를 전혀 찾지

못함을 보여준다. 이러한 점들을 볼 때, GAPS에서는 적절한 개수의 엘리트 우대를 적용하는 것이 바람직할 것이다.

다음으로는 비교 분석을 위하여 기존의 적합도 함수들을 앞과 같은 조건에서 그림 4의 네트워크에 적용시켜보고, 얻어진 결과들을 표 6에 요약하였다. 각 적합도 함수들은 (1)~(2)에서 볼 수 있는 것과 같은 형태로 작성되었으며, A , B 는 각각 총 이동거리 역수의 분자항과 실행불가능 해에 대한 벌점항을 의미한다. 따라서 표 6의 기존 적합도 1은 (1)과 같은 총 이동거리 역수 형태에 해당하며, 실행불가능 경로더라도 이동거리가 짧으면 적합도가 높게 산출되는 단점으로 인하여, 탐색 과정 동안 최적해는 물론이고 실행가능해 자체를 탐색하기 어려움을 볼 수 있다. 기존 적합도 2, 3의 경우에는 실행불가능 경로들에 벌점을 부과하는데, 기존 적합도 2는 모든 실행불가능 경로에 일괄적인 벌점을 주는 반면, 기존 적합도 3은 각 실행불가능 경로의 종료 시점에서 방문되지 않은 노드 개수를 벌점으로 주어 조기 종료된 경우에 높은 벌점이 부과되도록 하였다. 단, 음수의 적합도가 산출될 경우에는 무조건 적합도를 1로 수정하도록 하였으며, 이에 따라 조기종료되는 실행불가능 경로들이 같은 적합도를 갖는다는 문제가 있다. 이를 피하기 위하여 A 의 값을 늘릴 경우에는 반대로 조기종료되는 실행불가능 경로의 적합도가 이동거리가 긴 실행가능 경로보다 높게 산출될 수 있어, 기존 적합도에서 A , B 의 값을 주의깊게 설정되어야 한다.

표 6에서는 기존 적합도 2, 3을 사용하는 경우, 최적해를 찾는 데는 성공하나, 본 논문에서 제안하는 (3)에 비해 탐색 속도가 느린 점을 볼 수 있으며, A , B 의 값에 따라 탐색 성능은 더 나빠질 가능성도 존재한다. 이러한 점에서 본 논문에서 제안하는 유전 알고리즘의 장점을 확인할 수 있다.

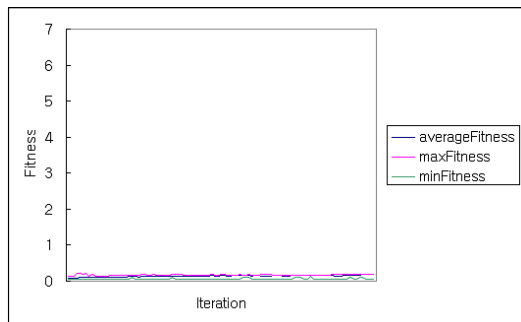


그림 7. $n_{elite} = 0$ 일 때의 실험 결과
Fig. 7. The Experiment Result under $n_{elite} = 0$

표 6. 기존 적합도 함수를 사용하여 얻은 실험 결과
Table 6. Experiment Results Obtained by Using Previous Fitness Functions

기존 적합도	A	B	실행가능해 (세대)
1	V_{total}	0	X
2	V_{total}	$V_{total} - 1$	5(62)
3	V_{total}	$V_{total} - V_p$	8(57)
			7(64) 5(72)

그림 8은 그림 4의 네트워크에 적절히 아크를 추가하여 단락이 없도록 수정한 것이다. 이 경우, 종전에 비해 많은 실행가능 경로가 존재하고, 가변길이 표현 방법의 적용도 용이해진다. 그림 9는 첫 번째 시험에서와 같이 엘리트 우대를 적용한 상태에서 GAPS를 이용하여 이 네트워크를 분석한 결과를 보여주며, 세대 내 최대 적합도와 평균 적합도가 지속적으로 상승하면서 최적해를 잘 찾아내는 모습을 보여준다. 아울러, 노드 1을 출발하여 노드 27에 도달하기 위해 필요한 최소의 총 이동거리는 5이며, 표 7과 같은 복수의 최적해들이 탐색되었다.

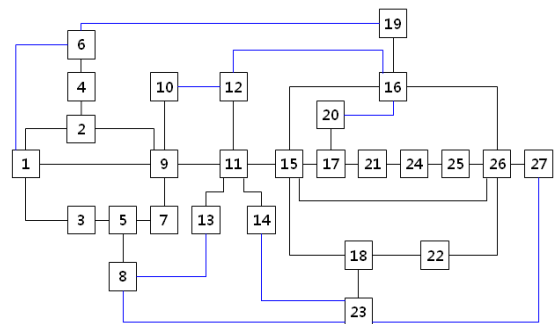


그림 8. 단락이 없는 네트워크
Fig. 8. A Network with No Dead-ends

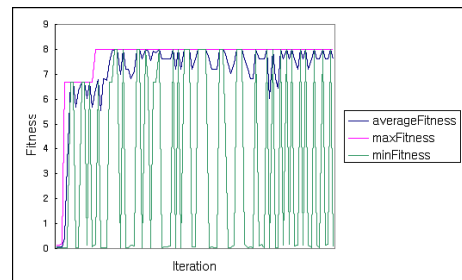


그림 9. 단락이 없는 네트워크 실험 결과
Fig. 9. The Experiment Result for the Network with No Dead-ends

표 7. GAPS에 의해 얻은 최적해들
Table 7. The optimal Solutions Obtained by GAPS

순번	경로
1	1→6→19→16→26→27
2	1→9→11→15→26→27
3	1→9→11→14→23→27
4	1→3→5→8→23→27

한편, 그림 9에서는 세대 내 최소 적합도의 편차가 그림 6에서보다 더 심하게 나타나는 것을 볼 수 있다. 이는 그림 8과 같은 네트워크는 실행가능경로를 만드는 것이 비교적 용이하여 탐색 과정에서 다양한 실행가능경로들이 나타났다가 유전 연산을 거치면서 파괴되는 일이 좀 더 빈번함을 의미한다. 나아가, 이러한 현상은 탐색의 효율성을 저하시킬 수 있고, 첫 번째 실험에서처럼 적절한 엘리트 우대를 통해 이 점을 어느 정도는 보완할 수 있겠으나, 경로들의 실행가능성을 유지하기 위한 부차적인 방법이 필요할 수도 있을 것으로 생각된다.

V. 결론 및 추후 연구 과제

경로탐색 문제는 다양한 분야에 응용할 수 있어, 다양한 해법이 연구되어왔다. 그러나 기존의 해법들을 종종 특정 형태의 네트워크에 특화되어 있거나 복잡한 작업 절차 또는 파라미터 설정이 필요하였다. 또한, 일반 사용자가 편리하게 쓸 수 있는 분석 시스템의 구현이 고려되지 않아 실제 현장에서 사용하기 곤란한 경우들도 존재하였다. 이러한 문제점들을 해결하기 위하여 본 논문에서는 네트워크 설계를 위한 GUI와 분석 과정을 통합적으로 관리하기 위한 데이터베이스를 포함하는 유전 알고리즘 기반 경로탐색 시스템, GAPS를 제안하였다.

GAPS는 네트워크 모델링을 위한 직관적인 GUI를 제공하며, 생성된 네트워크 정보와 탐색 절차는 모두 데이터베이스를 통해 관리된다. 최적의 경로탐색에는 유전 알고리즘이 사용하며, 일반적인 노드 설정 표현 방법을 이용하되, 실행 불가능 경로는 총 이동거리가 길수록 높은 적합도를 부여하여, 초기에는 실행가능 경로를 형성하는데 집중하고, 실행가능 경로가 찾아진 이후에는 총 이동거리가 짧은 것들에 높은 적합도를 부여하여 최적의 경로를 찾도록 하였다. 실험 결과, GAPS는 일반적인 네트워크 뿐 아니라 가변길이 표현 방법이 적용되기 어려운 미로 형태의 네트워크에도 잘 적용되며, 기존의 적합도보다 우수한 성능을 확인할 수 있었다.

반면, GAPS의 탐색 과정에서는 유전 연산을 거치면서 총

은 경로들의 실행가능성이 파괴될 가능성이 있기 때문에 이러한 점을 보완하는 방법이 필요할 것으로 보이며, 세대교체를 통해 얻어진 바람직한 경로에 대한 정보들을 활용하는 하이브리드 형태의 유전 알고리즘 개발 등이 대안이 될 수 있을 것으로 보인다. 아울러, 본 논문에서는 기존의 유전 알고리즘들과의 비교 분석만을 수행하였으나, 향후에는 다른 탐색 기법 및 메타 휴리스틱 방법들과의 비교 분석을 통해 GAPS의 성능 평가 및 개선도 병행할 것이다.

참고문헌

- [1] J.W. Kim, and M.J. Lee, "Applying Genetic Algorithm for Exploring the Effective Path in Network with Dead Ends," Journal of the Korea Knowledge Information Technology Society, Vol.8, No.1, pp.55-62, Feb. 2013.
- [2] S.B. Pattnaik, S. Mohan, and V.M. Tom, "Urban Bus Transit Route Network Design using Genetic Algorithm," Journal of Transportation Engineering, Vol.124, No.4, pp.368-375, Jul. 1998.
- [3] I-S. Chang, "A Genetic Algorithm for Route Guidance System in Intermodal Transportation Networks with Time-Schedule Constraints," Journal of the Korean Institute of Industrial Engineers, Vol.27, No.2, pp.140-149, Jun. 2001.
- [4] S. Li, M. Ding, C. Cai, and L. Jiang, "Efficient Path Planning Method based on Genetic Algorithm Combining Path Network," Proceedings of the 4th International Conference on Genetic and Evolutionary Computing, pp.194-197, Dec. 2010.
- [5] L.S. White, "Shortest Route Models for the Allocation of Inspection Effort on a Production Line," Management Science, Vol.15, No.5, pp.249-259, Jan. 1969.
- [6] L.S. Buriol, M.G.C. Resende, C.C. Ribeiro, and M. Thorup, "A Hybrid Genetic Algorithm for the Weight Setting Problem in OSPF/IS-IS Routing," Networks, Vol.46, No.1, pp.36-56, Jan. 2005
- [7] H. Chen, and B. Sun, "Multicast Routing Optimization Algorithm with Bandwidth and

- Delay Constraints based on GA," *Journal of Communication and Computer*, Vol.2, No.5, pp.63-67, May. 2005.
- [8] C. Chitra, and Subbaraj, P., "A Nondominated Sorting Genetic Algorithm Solution for Shortest Path Routing Problem in Computer Networks," *Expert System with Applications*, Vol.39, No.1, pp.1518-1525, Jan. 2012.
- [9] A.T. Ismail, A. Sheta, and M. Al-Weshah, "A Mobile Robot Path Planning using Genetic Algorithm in Static Environment," *Journal of Computer Science*, Vol.4, No.4, pp.341-344, Apr. 2008.
- [10] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path Planning for Autonomous Vehicles in Unknown Semi-Structured Environments," *The International Journal of Robotics Research*, Vol.29, No.5, pp.485-501, Apr. 2010.
- [11] M. Katsev, J. Yu, and S.M. LaValle, "Efficient Formation Path Planning on Large Graphs," *Proceedings of the IEEE International Conference on Robotics and Automation*, May. 2013.
- [12] D. Kim, H. Kim, and K. Yi, "Design of Near-Minimum Time Path Planning Algorithm for Autonomous Driving," *Transactions of the Korean Society of Mechanical Engineering A*, Vol.37, No.5, pp.609-617, May. 2013.
- [13] J.-Y. Kim, and S.-G. Kim, "Design and Implementation of Optimal Path Search Service using GPS Information in Photo File," *Journal of the Korea Society of Computer and Information*, Vol.17, No.12, pp.199-207, Dec. 2012.
- [14] L. Fu, D. Sun, and L.R. Rilett, "Heuristic Shortest Path Algorithms for Transportation Applications: State of the Art," *Computers and Operations Research*, Vol.33, No.11, pp.3324-3343, Nov. 2006.
- [15] V.S. Gordon, and Z. Matley, "Evolving Sparse Direction Maps for Maze Pathfinding," *Proceedings of Congress on Evolutionary Computation*, pp.835-838, Jun. 2004.
- [16] Y.V. Pehlivanoglu, "A New Vibrational Genetic Algorithm Enhanced with a Voronoi Diagram for Path Planning of Autonomous UAV," *Aerospace Science and Technology*, Vol.16, No.1, pp.47-55, Jan. 2012.
- [17] B. Coppin, "*Artificial Intelligence Illuminated*," Jones & Barlett Learning, 2004.
- [18] J.H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.
- [19] F. Altiparmak, M. Gen, L. Lin, and I. Karaoglan, "A Steady-state Genetic Algorithm for Multi-Product Supply Chain Network Design," *Computers and Industrial Engineering*, Vol.56, No.2, pp.521-537, Mar. 2009.
- [20] C.W. Ahn, and R.S. Ramakrishna, "A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations," *IEEE Transactions on Evolutionary Computation*, Vol.6, No.6, pp.566-579, Dec. 2002.
- [21] W. Wu, and Q. Ruan, "A Gene-Constrained Genetic Algorithm for Solving Shortest Path Problem," *Proceedings of the 7th International Conference on Signal Processing*, Vol.3, pp.2510-2513, Sep. 2004.
- [22] C. Bierwirth, D.C. Mattfeld, and H. Kopfer, "On Permutation Representation for Sequencing Problems," *Parallel Problem Solving from Nature IV*, Springer Berlin Heidelberg, pp.310-318, 1996.
- [23] S. Su and K. Tsuchiya, "Learning of a Maze using a Genetic Algorithm," *Proceedings of the International Conference on Industrial Electronics, Control, and Instrumentation*, pp.376-379, Nov. 1993.
- [24] C. Carrick, and K. MacLeod, "An Evaluation of Genetic Algorithm Solution in Optimization and Machine Learning," *Proceedings of the 1st Annual Conference Canadian Association for Information Sciences*, pp.224-231, Jul. 1993.

- [25] N. Baba, and H. Handa, "Genetic Algorithm Applied to Maze Passing Problem of Mobile Robots - A Comparison with the Learning Performance of the Hierarchical Structure Stochastic Automata," Proceedings of the 1994 IEEE International Conference on Neural Networks, pp.2690-2695, Jun. 1994.
- [26] M. Gen, R. Cheng, and D. Wang, "Genetic Algorithms for Solving Shortest Path Problems," Proceedings of the IEEE International Conference on Evolutionary Computation, pp.401-406, Apr. 1997.
- [27] M. Ericsson, M.G.C. Resende, and P.M. Pardalos, "A Genetic Algorithm for the Weight Setting Problem in OSPF Routing," Journal of Combinatorial Optimization, Vol.6, pp.299-333, Sep. 2002.
- [28] A.L. Corcoran, and R.L. Wainwright, "LibGA: A User-friendly Workbench for Order-based Genetic Algorithm Research," Proceedings of the 1993 ACM/SIGAPP Symposium on Applied Computing: States of the Art and Practice, pp.111-117, Mar. 1993.

감사의 글

이 논문은 동아대학교 교내연구비 지원에 의하여 연구되었음.

저 자 소 개



김 준 우

2001: 한국과학기술원
산업공학과 공학사.

2003: 한국과학기술원
산업공학과 공학석사.

2009: 한국과학기술원
산업 및 시스템공학과 공학박사

현 재: 동아대학교
산업경영공학과 조교수

관심분야: 데이터마이닝, 지능형시스템,
조합최적화, 데이터 시각화,
피지컬 컴퓨팅

Email : kjunwoo@dau.ac.kr