

## 유전 알고리즘을 이용한 이동 에이전트 기반의 경로 탐색 기법

지홍일\*

### Mobile Agent Based Route Search Method Using Genetic Algorithm

Hong-il, Ji\*

Department of Automotive Software, Youngdong University, Yeongdong 370-701, Korea

#### 요 약

본 논문에서는 제안한 알고리즘은 이전 유전 알고리즘의 분산처리를 위해 라우터 그룹 단위인 셀을 도입하였다. 셀 단위로 유전 알고리즘을 시행하여 전체 네트워크의 탐색 지연시간을 줄이는 방법을 제시하였다. 또한, 실험을 통하여 기존 최적경로 알고리즘인 Dijkstra 알고리즘에서 네트워크가 손상되었을 경우 제안한 알고리즘에는 대체 경로 설정의 연산시간이 단축되었으며 손상된 네트워크의 셀 안에서 2순위의 경로를 가지고 있으므로 Dijkstra 알고리즘보다 신속하게 대체경로를 설정하도록 설계되었다. 이는 제안한 알고리즘이 네트워크상에서 Dijkstra 알고리즘이 손상되었을 경우 대체 경로설정을 보완할 수 있음을 확인하였다.

#### ABSTRACT

Proposed algorithm in this thesis introduced cells, units of router group, to conduct distributed processing of previous genetic algorithm. This thesis presented ways to reduce search delay time of overall network through cell-based genetic algorithm. Also, through this experiment, in case of a network was damaged in existing optimal path algorithm, Dijkstra algorithm, the proposed algorithm was designed to route an alternative path and also as it has a 2nd shortest path in cells of the damaged network so it is faster than Dijkstra algorithm, The study showed that the proposal algorithm can support routing of alternative path, if Dijkstra algorithm is damaged in a network.

**키워드** : 유전 알고리즘, 이동 에이전트, 경로 탐색, 네트워크

**Key word** : Genetic Algorithm, Mobile Agent, Route Search Method, Network

Received 15 April 2015, Revised 27 August 2015, Accepted 07 September 2015

\* Corresponding Author Hong-il, Ji(E-mail: jiji61@yd.ac.kr, Tel+82-43-740-1522)

Department of Automotive Software, Youngdong University, Yeongdong 370-701, Korea

Open Access <http://dx.doi.org/10.6109/jkiice.2015.19.9.2037>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.  
Copyright © The Korea Institute of Information and Communication Engineering.

## I. 서론

정보화 사회가 가속화 되어감에 따라 광역 네트워크를 필요로 하게 되었다. 최근 많은 통신 네트워크에서 데이터 트래픽이 급격히 증가하게 된 원인은 새로운 멀티미디어 기술 및 개인 단말시스템의 발전에서 비롯되었다고 할 수 있다. 이로 인해 네트워크의 대역폭(Bandwidth) 확장과 광대역통합망인 WAN(Wide Area Network)에서 효율적인 네트워크 프로토콜을 필요로 하게 되었고 더욱 새로운 응용 서비스의 요구가 발생하였으며 여러 통신 기술(TCP/IP, ATM, SDH/SONET, WDM)들이 발전해 왔다[1,2].

멀티미디어 콘텐츠 응용은 재전송 요청과 수신을 반복하는 일반 데이터 트래픽과 달리 전송 지연에 민감한 특성을 가지며, End-To-End 형태로 구성되어 있는 현재 인터넷 환경에서는 전송되는 전송 스트림의 QoS를 보장하기 위한 여러 연구들이 진행되고 있다.

본 논문에서는 네트워크를 셀 단위로 구분하고 유전 알고리즘을 적용하여 대체 경로를 찾는 알고리즘을 제안하였다. 이는 유전 알고리즘을 전체 네트워크에 적용하였을 때보다 연산시간을 단축할 수 있는 장점이 있을 뿐만 아니라, 네트워크 경로 상에서 특정 라우터가 손상되었을 경우 손상 라우터를 우회하는 대체 경로 설정에 유리한 특성이 있다[3,4].

논문에서 제안한 알고리즘은 목적지에서 인접한 라우터로 이동한 후 일정크기의 셀을 구성하고 일정 크기의 홉만큼 최적경로에 위치한 라우터를 탐색한다. 이 라우터를 중심으로 클러스터의 경계에 위치한 일정크기 홉 간격의 라우터를 검색한 후 역으로 일정크기 홉 간격의 라우터를 첫 번째 라우터가 위치한 클러스터를 경계로 탐색한다. 탐색된 라우터들을 기준으로 첫 번째 찾는 방법과 같이 반복 수행하여 셀을 증식시켜 나간다. 이후 생성된 셀들의 공통 영역을 바탕으로 셀 내부에서 유전 알고리즘을 적용하고 도출된 경로로부터 양 셀 간 경합으로 우성인 경로를 취합한다. 이러한 절차를 목적지가 마지막 셀에 포함될 때까지 셀을 증식시키며 수행을 반복한다. 또한, 실험을 통하여 네트워크에서 특정 라우터의 이상으로 대체경로의 설정시 본 논문에 제안한 알고리즘이 기존 Dijkstra 알고리즘이나 기존 유전자 교배연산에 기초한 라우팅유전 알고리즘보다 대체경로 설정에 우수한 지를 분석하였으며 그에 따른 지연시간

과 비용(Cost)값을 측정하였다.

## II. 제안한 경로 탐색알고리즘

### 2.1. 알고리즘의 구조

유전 알고리즘을 이용한 이동 에이전트 경로 탐색 방법은 최단거리 알고리즘의 방향성을 이용하여 라우팅 시 최적 경로를 탐색하는 알고리즘이다.

클라이언트로부터 요청된 정보를 전달하기 위해 서버쪽에서는 클라이언트로 최단 경로를 탐색하기 위해 에이전트 컨트롤러를 이용하여 경로에 위치한 라우터들을 검색하고 그들을 통해 서로 간의 최단거리를 검색하도록 하였다. 그리고 노드 장에서 클라이언트 쪽의 "Path Modification"을 통해 장애노드를 포함하고 있는 셀에서 재설정 작업을 시행하도록 하였다.

네트워크 내에서 일정 변심거리에 의해 클러스터형식의 셀을 증식시켜 그 안에서 최저비용을 지불하는 경로를 선택하는데 <그림 1>의 "Internet Route's" 안에 보이는  $C_1, C_2, C_3, C_4, \dots, C_n$ , 들이 제안한 알고리즘에 의해 생성된 셀이며  $Cs_1, Cs_2, Cs_3, Cs_4, \dots, Cs_n$ 는 각 셀 간의 중복된 라우터들의 그룹이다.

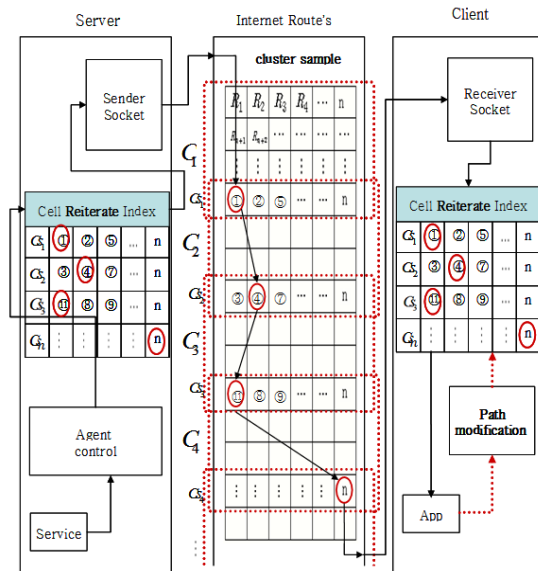


그림 1. 제안한 시스템의 구조  
Fig. 1 The structure of the proposed system

2.2. 제안한 경로 탐색 알고리즘 수행 과정

2.2.1. 동작 모델

본 논문에서 제안한 알고리즘의 기본 동작은 에이전트에 의해 초기화를 진행한 후 셀을 증식하며 각 셀 내의 에이전트들을 통해 셀 내의 유전알고리즘을 이용하여 최적경로를 구하고 인접 셀과 최적경로 비교를 통해서로 간의 우열을 가리는 과정을 거쳐 목적지까지 도달하는 과정으로 이루어지며, <그림 2>는 이러한 과정을 나타내는 모델이다[5].

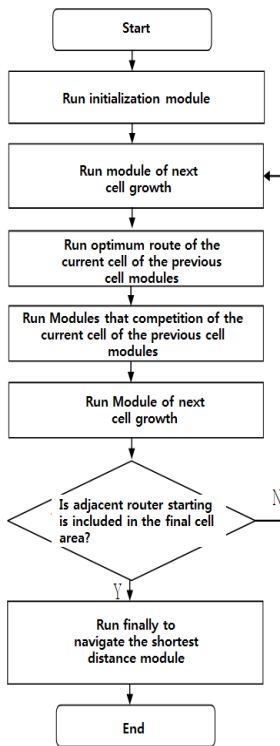


그림 2. 제안한 경로 탐색 알고리즘의 순서도  
Fig. 2 Flowchart of the proposed route search algorithm

2.2.2. 셀 생성 과정

제안한 알고리즘의 셀들은 3-블록(Block)으로 구성되어 있다. 1-블록은 시작 셀의 경계점을 생성하고 다음 라우터로 에이전트를 이주시키는 기능을 수행한다. 2-블록은 경합을 위한 셀 2개로 구성되어 있으며 2개의 셀 중 각각의 유전 알고리즘을 실행을 통한 최단경로를 찾는 기능을 수행하며 3-블록은 관문 역할을 하고 1-블록 역할과 함께 다음 2-블록 계층을 생성하는 역할을 수

행한다.

<그림 3>은 제안한 알고리즘에서의 에이전트에 의한 셀 형성에서 3-블록 단계 형태를 나타낸다.

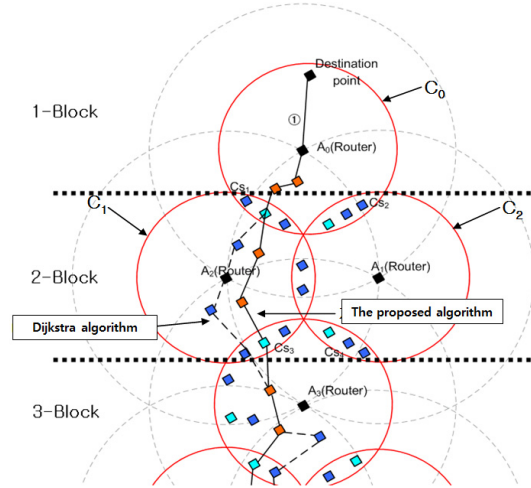


그림 3. 셀 수행 단계별 3-블록 형태  
Fig. 3 Perform step 3 blocks form the cell

(가) 1-블록

목적지(Destination point)로부터 바로 옆에 인접한 라우터의 에이전트로부터 초기화 과정을 수행한다. 초기화 수행을 시작한 라우터를 중심으로 변심거리  $\epsilon$  흡인 기존 알고리즘 경로내의 라우터  $R_n$ 를 검색한다.

$$C_0 = R_1, R_2, \dots \quad (2-1)$$

$$\text{라우터 검색 영역의 크기} = \frac{1}{2} \epsilon \pi \quad (2-2)$$

$$\text{셀 영역의 크기} = \frac{1}{2} \alpha \pi \quad (2-3)$$

$C_0$ 은 초기화 과정에 의해 생성된 첫 번째 셀의 관할 안에 있는 라우터들의 그룹이며 그에 대한 영역의 크기는 식 2-3과 같다.

(나) 2-블록

1-블록 과정에서 셀 중심 에이전트  $A_1$  로부터 검색된 라우터  $R_n$  로부터  $\epsilon$  흡이며 시작 에이전트(라우터)와 변심거리  $\epsilon$  근처에 포함되는 라우터 1개를 검색한다. 만약 그 라우터가 폭주가 있을 경우나 장애가 발생

하였을 경우 주변 근거리에서 위치한 라우터를 선별한다. <그림 4>은 셀의 영역에서 셀 에이전트 증식에 대한 그림이다.

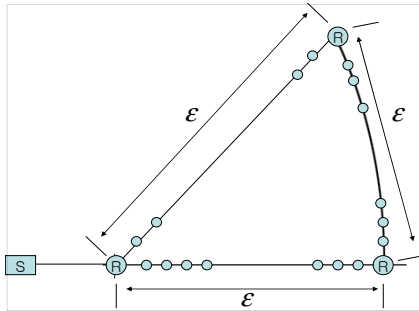


그림 4. 셀 영역에서 셀 에이전트 증식  
Fig. 4 Cell proliferative agent in the cell area

검색된 라우터로부터 거리는  $\epsilon$  홉이며 시작라우터의 클러스터 경계 근방에 위치하고 있고 이전에 검색된 라우터 R 근처의 라우터를 검색하여 에이전트를 이주한다. 이주된 에이전트에 의해 초기화 과정을 수행한 후 첫 번째 블록과 두 번째 블록간의 교집합  $Cs_1$ 과  $Cs_2$ 를 구한다.

$$\begin{aligned} Cs_1 &= C_0 \cap C_1 \\ Cs_2 &= C_0 \cap C_2 \end{aligned} \quad (2-4)$$

그리고  $C_1$  과  $C_2$ 의 라우터 그룹의 교집합( $\tau$ )을 구한다.

$$\tau = C_1 \cap C_2 \quad (2-5)$$

(다) 3-블록

2-블록의 에이전트들간 각각  $\epsilon$ 홉의 거리에 위치하며 기존 알고리즘 근처에 위치하는 라우터에 에이전트를 이식한다. 다음 절차로 이주된 에이전트에 의해 초기화 과정을 수행하여 각 셀 간의 관할 구역을 설정한 후 2-블록과 3-블록간의 교집합( $Cs_3, Cs_4$ )을 구한다.

$$Cs_3 = C_1 \cap C_3, Cs_4 = C_2 \cap C_3 \quad (2-6)$$

집합  $Cs_1, Cs_2, Cs_3, Cs_4$ 에 대하여 유전 알고리즘을 이용해 시작지점과 마지막 지점을 단위 셀로 하여 최적 경로를 구한다.

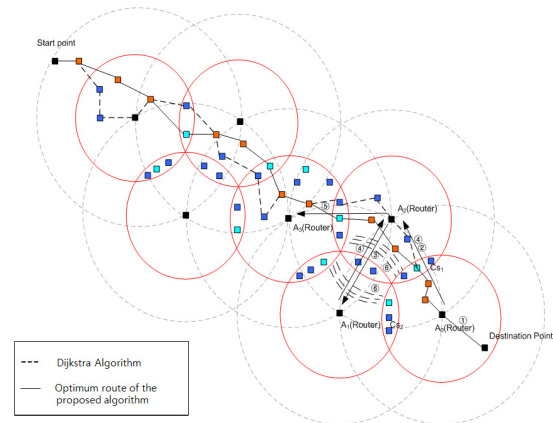


그림 5. 제안한 알고리즘의 수행 과정  
Fig. 5 Implementation process of the proposed algorithm

<그림 5>는 제안 알고리즘의 적용 예를 나타낸 것으로서, 점선 부분은 기존 라우팅 알고리즘을 적용하여 방향성을 부여한 부분이며, 실선 부분은 기존 라우팅 알고리즘의 방향성을 바탕으로 제안 알고리즘의 수행 과정을 나타낸다.

본 논문에서 제안한 알고리즘의 수행 절차를 간단히 살펴보면 다음과 같다.

- ① 단계 : 첫 번째 시작지점에서 첫 번째 라우터로 이동한 후 일정 크기의 셀을 형성한다.
- ② 단계 : 일정 크기의 홉만큼 최적경로에 위치한 라우터를 찾는다.
- ③ 단계 : 첫 번째 라우터가 위치한 클러스터의 경계에 위치한 일정크기의 홉 간격의 라우터를 검색한다.
- ④ 단계 : 그 후 반대로 일정 크기의 홉 간격의 라우터를 첫 번째 라우터가 위치한 클러스터를 경계로 찾는다.
- ⑤ 단계 : 찾은 라우터들을 기준으로 첫 번째 찾는 방법과 같이 반복해서 수행하여 셀을 증식시킨다.
- ⑥ 단계 : 그 후 생성된 셀들의 공통된 부분을 바탕으로 셀 내부에서 유전 알고리즘을 이용한 최적 경로 알고리즘을 수행한 후 도출된 경로로부터 결합하여 구성인 경로를 취합한다.

⑦ 단계 : 이러한 절차를 목적지 노드가 마지막 셀에 포함될 때까지 셀을 증식시키며 수행한다.

### III. 실험 및 성능 분석

#### 3.1. 실험 환경

본 논문에서 제안한 경로탐색 유전 알고리즘의 성능 평가를 위해 기존 유전 알고리즘에서 제시한 Munetomo 알고리즘과 최적 경로 탐색 알고리즘인 Dijkstra 알고리즘과의 전송지연 및 처리율을 비교하고 분석하였다. 실험을 위한 시뮬레이션 모델은 네트워크 모델 설정의 민감성을 반영하여 구성에 필요한 기본적인 요소를 포함시켰고 이동 에이전트의 특성을 실험할 수 있도록 검증된 토폴로지를 확장하였다[6,7].

표 1. 파라미터의 정의

Table. 1 The definition of parameters

| Setting                                       | Explanation                               | Value   |
|---|---|---------|
| Cell_range                                    | Cell range( $\mathcal{E}$ )               | 5       |
| Cell_control_range                            | Cell control range( $\alpha$ )            | 3       |
| rou_t_pop                                     | Number of objects                         | 300     |
| Generation                                    | Performance number                        | 100     |
| Deviation of the delay time                   | Deviation time delay between the router   | 0.1     |
| The initial transfer rate                     | Transmission rate when setting a route    | 60kbyte |
| Delay time rate of The target                 | Standard delay time of Dijkstra Algorithm | 0.4     |
| Scope of the route tables                     | Identifiable number of route hops         | 16      |
| The number of agents created at the same time | Agent of the largest generation           | 4       |
| Transmission control cycle                    | ICMP Cycle of dispatch                    | 20ms    |

셀 범위는 최적 셀 크기의 실험에 의하여 셀 거리를 5로 설정하였으며 그에 따른 셀의 관할 구역을 3으로 설정하였다. 그리고 실험군내의 라우터 군집은 300개로 하였으며 수행횟수는 100으로 하였다. 표 1은 경로 탐색을 위하여 셀의 영역을 구하기 위한 초기 설정을 위한 파라미터를 나타낸다.

#### 3.2. 실험 및 성능 분석

##### 3.2.1. 실험에 필요한 셀의 크기 결정

본 논문에서는 300개의 라우터를 갖는 하이브리드 토폴로지의 환경의 시뮬레이션 모델에 적합한 셀의 크기를 결정하여야 한다. 제안한 알고리즘에서는 4개 이상의 셀이 형성되어야 실험이 가능하며, 이를 본 실험에 사용된 토폴로지에 적용하면 10홉 이하의 홉 수가 형성된다. 11홉 이상의 홉 수에서는 본 실험에서 요구된 4개 이상의 셀이 형성되지 않기 때문에 본 실험환경에 적합하지 않다.

또한, 4홉 이하 경우는 Control 셀이 셀 간의 중첩부분이 생기지 않으므로 Control 셀이 형성되지 않으므로 실험 환경을 만족시킬 수 없다. 이에 본 실험을 만족하기 위해서는 홉 수가 5홉에서 10홉 사이에서 셀의 크기를 결정하여야 한다. 실험에서 연산시간이 적을수록 신뢰도가 높아지므로 적용한 셀의 크기는 연산시간이 가장 작은 5홉으로 결정하였다.

##### 3.2.2. 제안한 알고리즘의 대체 경로 설정에 따른 성능 비교 분석

###### (가) 대체 경로 설정 연산 지연 시간 비교

시뮬레이션 토폴로지 상에서 100번 라우터를 손상시켜 대체 경로를 설정할 때 Dijkstra 알고리즘, Munetomo 알고리즘과 제안한 알고리즘의 대체 경로 설정에 따른 연산 시간을 비교하였다.

<그림 6>은 대체 경로 설정에 따른 연산 시간을 비교한 그래프이다.

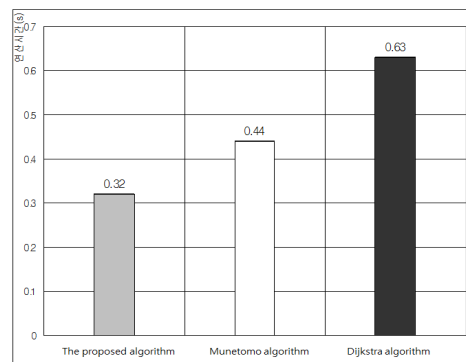


그림 6. 대체 경로 설정에 따른 연산 시간 비교

Fig. 6 Comparison operation time according to the alternative routing

이 결과에 따르면 제안한 알고리즘은 연산 시간이 약 0.32초 정도로 나타나며, Munetomo 알고리즘은 약 0.44초, Dijkstra 알고리즘은 약 0.63초로 나타났다. 이는 제안한 알고리즘이 Dijkstra 알고리즘보다 대체 경로 설정에 따른 연산 시간이 약 2배정도 빠르다는 것을 보여 주고 있으며, Munetomo 알고리즘보다는 약 37.5% 정도의 대체 경로 설정 연산 시간이 빠르다는 것을 보여 주고 있다. 이는 제안한 알고리즘은 각 셀 내에서 2 순위의 경로 정보를 가지고 있기 때문에 연산 시간이 Dijkstra 알고리즘이나 Munetomo 알고리즘보다 짧은 것이다[8,9].

(나) 대체 경로 처리 비용(cost) 비교

네트워크 경로 상에서 특정 라우터가 손상되었을 경우 손상 라우터를 우회하는 대체 경로를 설정하여야 한다. 본 실험에서는 Dijkstra 알고리즘과 Munetomo 알고리즘과의 대체 경로 설정을 비교 분석해 보았다.

시뮬레이션 토폴로지 상에서 100번 라우터를 손상시켜 Dijkstra 알고리즘, Munetomo 알고리즘과 제안한 알고리즘의 대체 경로가 어떻게 설정되는지 실험해 보았다. 먼저 Dijkstra 알고리즘은 102-168-169-177 로 대체 경로를 설정하여 패킷을 전송하였으며, Munetomo 알고리즘도 Dijkstra 알고리즘과 동일한 대체 경로를 설정하였다. 제안한 알고리즘은 67-99-98-103으로 대체 경로를 설정하였다. <그림 7>에서는 3가지 알고리즘 모두 거쳐 가는 라우터의 수는 기존 2개에서 3개로 증가하였음을 보여주고 있다. 비용(cost)면에서는 Dijkstra 알고리즘과 Munetomo 알고리즘은 기존의 경로보다 대체 경로의 비용(cost)이 146이 증가하였으며 제안한 알고리즘은 115의 비용이 증가함을 알 수 있다. 이것은 비용 측면에서 볼 때 제안한 알고리즘이 Dijkstra 알고리즘과 Munetomo 알고리즘보다 비용면에서 31이 감소함을 보여 주고 있다. 이 실험 결과로 볼 때 제안한 알고리즘이 대체 경로 설정에서는 최적 경로 설정 알고리즘인 Dijkstra 알고리즘과 기존 유전 알고리즘인 Munetomo 알고리즘보다 우수하다는 것을 알 수 있다.

대체 경로 설정 이후 전체 경로 비용에서 비교해 볼 때 Dijkstra 알고리즘의 경로 전체 경로 비용은 2,418 이고, Munetomo 알고리즘의 경우 전체 경로 비용은 2,494이며, 제안한 알고리즘은 전체 경로 비용이 2,412 로 나타났다. 이는 대체 경로가 설정되었을 때 최적성

을 비교해 보면 Dijkstra 알고리즘보다 경로 비용이 6(0.63%)이 감소하였고, 대체 경로 설정 이후에서는 기존의 Dijkstra 알고리즘보다 최적성이 우수하다는 것을 알 수 있다. 또한 기존 유전 알고리즘인 Munetomo 알고리즘과 비교해 볼 때 경로 비용이 72(3.25%)가 감소하였는데, 이는 기존 유전 알고리즘보다 최적성이 우수하다는 것을 증명한 것이다. 따라서 제안 알고리즘의 손상 경로에 대한 대체 경로 설정 시 비용측면에서의 최적성 성능에 대한 값은 Dijkstra 알고리즘이나 Munetomo 알고리즘보다 우수함을 증명하였다. <그림 7>는 대체 경로 설정에 따른 경로 설정을 나타내고 있다.

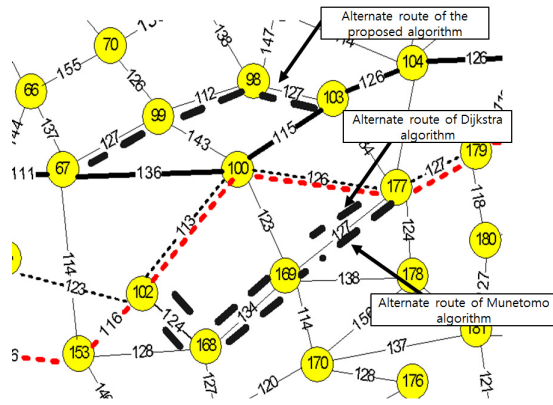


그림 7. 라우터 손상시 대체 경로 설정  
Fig. 7 Alternative routes when setting the router injury

IV. 결 론

본 논문에서는 기존 유전 알고리즘과 이동에이전트 에 대해 분석하고 기존 유전 알고리즘보다 향상된 유전 알고리즘을 이용하여 대체 경로 탐색 기능을 향상 시키는 알고리즘을 제안하였다. 이전 연구에서 최적경로 알고리즘으로 알려진 Dijkstra 알고리즘은 네트워크가 손상되었을 경우 대체 경로 설정에 지연이 발생하는 약점이 있었다. 이를 개선하기 위하여 유전 알고리즘을 이용한 대체 경로 설정기능을 추가하여 제안한 경로탐색 알고리즘의 대체 경로 설정 연산시간이 단축됨을 확인할 수 있었다. 이는 손상된 네트워크의 셀 안에서 제 2 순위의 경로를 확보하고 있으므로 Dijkstra 알고리즘보



다 신속하게 대체경로를 설정하는 기능에서 비롯되었다. 유전 알고리즘의 분산처리를 위해 도입한 라우터 그룹 단위인 셀은 경로 설정시 유전 알고리즘을 시행하는 단위로 전체 네트워크의 탐색 지연시간을 단축하는 방안으로 제시하였다.

실험을 통한 결과는 제안한 알고리즘이 네트워크 경로 상에서 특정 라우터가 손상되었을 경우 손상 라우터를 우회하는 대체 경로 설정에 따른 경로 처리 비용과 연산시간을 Dijkstra 알고리즘과 Munetomo 알고리즘에 비교 분석하였다. 실험 결과 대체 경로 처리 비용에서는 제안한 알고리즘이 Dijkstra 알고리즘과 Munetomo 알고리즘보다 대체 경로 설정에 따른 경로 처리 비용을 약 27%정도 감소시키며, 대체 경로 설정에 따른 연산시간이 Dijkstra 알고리즘보다 약 2배 정도가 빠르다는 것을 보여 주었다. 이는 라우터의 손상시 대체 경로 설정에서는 본 논문에서 제안한 알고리즘이 Dijkstra 알고리즘이나 Munetomo 알고리즘보다 우수하다고 할 수 있다.

## REFERENCES

[1] G. Cabri, L. Leonardi, F. Zambonelli, "Mobile-Agent Coordination Models for Internet Applications", *IEEE Computer*, Vol. 33, No. 2, Feb. 2000.

[2] V. A. Pham and A. Karmouch, "Mobile Software Agents: An Overview", *IEEE Communications Magazine*, Jul. 1998.

[3] Filipe Araújo, Bernardete Ribeiro, Luís Rodrigues, "A neural network for shortest path computation", *Neural Networks, IEEE Transactions on*, Volume 12, Issue 5, pp1067 - 1073, Sep. 2001.

[4] W. Stallings, "HIGH-SPEED NETWORKS TCP/IP AND ATM DESIGN PRINCIPLES", Prentice Hall, 2000.

[5] J. Baumann, F. Hohl, K. Rothermel, M. Strasser, W. Theilmann, "MOLE: A mobile agent system", *Software: Practice and Experience*, Volume 36, Issue 15, Dec. 2006.

[6] B. Liu, S. Choo, S. Lok, S. Leong, S. Lee, F. Poon, H. Tan, "Integrating case-based reasoning, knowledge-based approach and Dijkstra algorithm for route finding", *Artificial Intelligence for Applications*, 1994, Proceedings of the Tenth Conference, pp. 149 - 155, 1994.

[7] C. Xi, F. Qi, L. Wei, "A New Shortest Path Algorithm based on Heuristic Strategy", *Proceedings of the 6th World Congress on Intelligent Control and Automation*, 2006.

[8] M. Munemoto, Y. Takai, and Y. Sato, "A migration scheme for the genetic adaptive routing algorithm," in Proc. *IEEE Int. Conf. Systems, Man, and Cybernetics*, pp. 2774-2779, 1998.

[9] J. Inagaki, M. Haseyama, and H. Kitajima, "A genetic algorithm for determining multiple routes and its applications," in Proc. *IEEE Int. Symp. Circuits and Systems*, pp. 137-140, 1999.



지홍일(Hong-il, Ji)

충북대학교 컴퓨터공학과 공학박사  
 2014년 ~ 현재 영동대학교 자동차소프트웨어학과 조교수  
 ※관심분야 : 데이터통신, 임베디드시스템, 네트워크, 이터닝