

Q-Learning 기반 재난환경에서 사용자의 최적 탈출 경로에 대한 연구

최 승 희*, 유 상 조^o

Q-Learning Based Optimal Escape Route Decision in a Disaster Environment

Seung-Hee Choi*, Sang-Jo Yoo^o

요 약

재난 상황 중 대형 화재의 경우 화재 초기에 사망자 대부분이 발생하기 때문에 신속히 안전한 경로로 대피하는 것이 중요하다. 하지만 건물 내부에 머물던 사람의 경우 자신이 위치한 층 외의 다른 층까지 고려한 상황 판단을 할 수 없다는 문제가 있다. 따라서 다른 층의 상황을 고려한 최적의 탈출 경로를 제시하여 화재에 의한 피해를 줄일 수 있는 시스템이 필요하다. 이에 본 논문은 건물에 화재가 발생한 상황을 고려하여 사용자의 현재 위치에서 최적의 탈출 경로를 찾기 위한 Q-Learning 기반 시스템 모델을 제시한다. 하나의 층에서의 최단 경로 예측이 아닌 다층 건물 구조에서 층마다 다른 출구의 위치와 화재의 위치를 고려한 최단 경로를 예측한다. 또한, 사용자의 현재 위치 주변에 사람이 많은 경우 모두 최단 경로만을 선택하면 병목현상이 일어나 대피가 원활하지 않을 수 있다는 점을 고려한다. 사용자 주변에 사람의 밀집 정도에 따라 최적의 대피경로와 차선의 대피경로로 적절히 분산시키는 시스템 모델을 제안하고 모의실험을 통해 결과를 확인한다.

Key Words : Q-Learning , Reinforcement learning, Optimal escape route, Disaster, Fire, IoT based Fire Detection, IoT

ABSTRACT

It is important to evacuate quickly to a safe route since most of the deaths occur in the initial fire. However, there is a problem that a person in a building cannot decide considering other floors situation. Therefore, a system is needed to reduce the damage caused by fire by presenting an optimal escape route considering the situations of other floors. Thus, we propose a Q-Learning based system model to find the optimal escape route from the current location of the user, considering the fire in a building. The shortest path is predicted according to multi-story building structure and considers the different location of exits and fire each floor. Also, consider that if there are many people around the current location of the user, selecting only the shortest path may cause bottlenecks and poor evacuation. We propose a system model that appropriately distributes the optimal escape route and the next optimal escape route according to the number of people around the user and confirm the results through simulation.

* 이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (2020R1F1A1053006)

• First Author : Inha University Department of Electrical and Computer Engineering, cshhhh16@inha.edu, 학생회원

o Corresponding Author : Inha University Department of Information and Communication Engineering, sjyoo@inha.ac.kr, 종신회원
논문번호 : 202010-266-B-RN, Received October 27, 2020; Revised January 13, 2021; Accepted January 14, 2021

I. 서론

재난·재해에 의한 인명 피해에 따라, 국내외에서는 재난 발생 시 적합한 대응책과 대피 요령의 필요성에 대한 인식이 증가하고 있다. 재난 상황 중 대형 화재의 경우 사망자의 대부분이 화재 초기에 발생한다. 따라서 화재에 의한 인명 피해를 줄이기 위해서는 정확한 상황 판단에 따라 신속하고 효율적인 대피 경로를 파악하는 것이 중요하다. 하지만 현재로써 재난 발생 시 대응책으로는 위험 상황을 알리는 긴급 재난 문자 시스템만 존재할 뿐, 개개인의 특수한 상황을 고려한 시스템은 제대로 구축되지 않았다는 문제가 있다. 따라서 본 논문에서는 건물 내에 화재가 발생하였을 때, 사용자의 현재 위치에서 최적의 탈출 경로를 찾을 수 있는 Q-Learning 기반의 시스템 모델을 제시한다.

IoT (Internet of Things) 시스템을 이용하여 화재의 위치와 사용자의 위치를 파악한다.^[1,2] 건물 내에 화재 감지기를 설치하여 화재의 위치를 감지한다. 일반적으로 사용자의 위치를 파악하기 위해 GPS를 이용하는데 GPS의 경우 기본적으로 10m 이상의 오차가 발생하고 실내의 경우에는 그 오차가 더욱 커지기 때문에 실내에서 사용하기에는 어려움이 있다. 따라서 건물 내에 BLE(Bluetooth Low Energy) 비콘을 설치하고 비콘에서 받은 데이터를 이용하여 사용자의 위치를 파악한다.

경로를 안내하기 위한 시스템 모델은 Fool Proof와 Fail Safe의 두 가지 원칙을 고려한다. Fool Proof와 Fail Safe는 피난 대책이 따르는 두 가지 원칙으로, Fool Proof는 비상 상태의 혼란스러운 상황에 놓인 사람은 제대로 된 판단을 하지 못 하는 것을 고려한 원칙을 말한다. Fail Safe는 하나의 수단이 실패하여도 다음 수단에 의하여 성공할 수 있어야 함을 의미한다. 두 원칙에 따라 피난 시 경로는 간단 명료해야하고 양방향의 피난로를 상시 확보해야한다는 점을 고려하여 시스템 모델을 구성한다.

긴급 상황에서 건물 내부에 머물던 사람의 경우, 건물의 구조를 제대로 인식하지 못해 출구를 찾지 못하거나 비효율적인 경로를 선택하여 탈출에 걸리는 시간이 지연되는 문제들이 발생한다.^[3] 기존 Q-Learning을 이용하여 경로를 찾는 방법은 점의 대칭 이동을 이용하여 최단 거리를 구하는 방식과 같다. 이 방식으로 다층 건물의 최단 경로를 찾는 경우, 건물의 층이 증가함에 따라 학습을 위한 환경의 크기가 기하급수적으로 증가하는 문제가 발생한다. 이에 본 논문은 기존 방식에 비해 학습에 사용되는 메모리와 걸리는 시간을 절약하여 최단 경로를 찾을 수 있는 Q-Learning 모델을

제안한다.

또한 사용자의 현재 위치 주변에 사람이 많은 경우 모두에게 최단 경로만을 제시하면 병목 현상이 일어나 탈출이 원활하지 않을 수 있다는 점을 고려한다. 따라서 양방향 피난로를 이용하여 최적의 탈출 경로를 안내할 수 있는 시스템 모델을 제안한다. 이때 경로 파악에는 화재에 의한 고립과 같이 화재로 인해 발생할 수 있는 특수한 상황 역시 고려한다.

본 논문의 구성은 다음과 같다. 2장에서는 Q-Learning에 필요한 입력 값을 구하는 방법에 관련된 연구와 최적의 경로를 찾는 방법에 관련된 연구들을 살펴본다. 3장에서는 본 논문이 제안하는 최적 경로를 찾기 위한 Q-Learning 기반 시스템 모델에 대해서 설명한다. 4장에서는 시뮬레이션을 위한 설정 및 시뮬레이션 결과에 대해서 설명하고 이를 이용하여 본 논문이 제안하는 시스템 모델의 성능을 확인한다. 마지막 5장에서는 본 논문의 결론에 대해서 설명한다.

II. 관련 연구

최적 경로를 얻기 위한 학습을 위해서는 사용자의 현재 위치 및 사용자 주변에 사람의 수, 건물의 구조에 대한 정보, 화재의 위치에 대한 정보가 입력으로 필요하다. 따라서 본 장에서는 학습에 필요한 정보들을 얻는 방법에 관련된 연구와 최적 경로를 얻기 위한 관련 연구에 대해서 살펴본다.

화재의 위치를 파악하기 위해 H.J. Oh는 사물인터넷 기반 화재감지 및 실내 대피 내비게이션 시스템을 제안하였다.^[4] 화재 감지를 위해 화재 감지 센서를 사용하였다. 단, 단일 센서를 이용하여 화재를 감지하는 경우 오작동의 문제가 발생할 수 있다는 점을 고려하여 온도 센서, 불꽃 감지 센서를 융합하여 화재 감지를 구현하였다. 화재 감지기가 주변에서 발생한 화재를 감지하면 화재 발생 정보를 서버에 전송한다. 서버는 화재 발생에 대한 사실과 발생 위치에 대한 정보를 받게 된다. T.Y. Kim은 실시간 화재 감지를 위한 상황인지 기반 IoT 시스템을 제안하였다.^[5] 이 시스템은 화재 감지 센서 데이터와 영상 인식 결과를 모두 사용한다. 온습도 센서, 유해 가스 감지 센서, 일산화탄소 감지 센서, 이산화탄소 감지 센서를 융합하여 화재 감지기를 구현하였고 MQTT(Message Queuing Telemetry Transport)를 이용하여 정보를 전송하였다. 또한 웹 카메라로부터 얻게 되는 실시간 영상 데이터를 미리 학습된 Deep-CNN 모델에 입력으로 사용하여 화재 발생 확률을 계산하였다. 센서의 측정 범위를 벗어나는 경우

에는 영상 인식 결과를 사용하였고, 카메라의 사각지대의 경우에는 센서 데이터를 이용하여 화재 감지의 정확도와 속도를 향상 시켰다.

사용자의 위치를 파악하기 위해 Y.J. Koh는 실내 위치 추적과 경로 최적화를 통한 실시간 화재 대피 시스템을 제안했다.^[6] Y.J. Koh는 단순히 비콘의 값이 큰 곳을 사용자의 위치로 결정하는 경우, 비콘의 반경 및 실내 장애물로 인한 위치 측정 오차율이 커지는 것을 지적하며 비콘을 이용한 핑거프린팅 기법을 선택하였다. 핑거프린팅 기술은 건물 내부 구조를 일정한 크기의 셀로 나누고 각각의 셀의 위치에서 얻게 되는 비콘의 신호 세기를 데이터베이스화하여 위치를 추적할 때 사용한다. 핑거프린팅을 이용한 위치 파악은 정확도가 높지만 데이터베이스를 수집하는 과정이 복잡하고 비콘의 위치가 바뀌는 경우 데이터베이스를 새로 수집해야 한다는 단점을 가지고 있다. H.J. Oh는 BLE(Battery Low Energy) 비콘의 무선 신호 세기를 이용한 삼각측량법을 이용하여 사용자의 실내 위치를 측정하였다.^[4] J.S. Kim은 비콘을 이용한 실내 내비게이션 시스템을 제안하였는데 사용자의 위치를 측정하기 위해 마찬가지로 BLE 비콘과 삼각측량법을 이용하였다.^[7] 삼각측량법만으로 사용자의 위치를 추정하는 경우 벽면과 주변 물체로부터 반사된 신호로 인한 오차가 거리가 늘어남에 따라 크게 증가한다는 것을 지적하였다. 이에 따라 칼만 필터와 파티클 필터를 이용하여 측정값 중 잡음을 걸러내었다

S.P. Jung은 대표적인 길 찾기 알고리즘인 A* 알고리즘과 Dijkstra 알고리즘, 그리고 NavMesh 알고리즘에 대해 분석하였다.^[8] A* 알고리즘은 우선순위를 부여하는 휴리스틱을 사용하여 추정 방향으로의 경로탐색으로서 목표지점까지의 일직선(Link) 거리 값을 찾는 방식이다. Dijkstra 알고리즘은 최단거리 탐색을 보장하는 알고리즘으로써, 매 반복마다 거리의 누적 값이 가장 작은 노드의 경로를 표시하는 Label-setting 기법이다. NavMesh는 3D의 입체공간을 2D로 바꾸어 사용한다. 경로탐색은 앞서 설명한 A*의 탐색방법과 같으나 A*는 이동하는 공간을 격자(Cell)로 나누어 각 격자가 절점이 되어 목표지점까지를 계산하게 되고, 내비게이션의 경우 절점을 활용하기보다 공간에서 길(표면)을 계산한다. 이러한 길 찾기 알고리즘들은 만약 중간 노드의 상태가 동적으로 변화하는 경우, 전체 토폴로지를 다시 구성하고 각 위치에서의 최적경로를 찾기 위한 비용을 다시 계산해야 한다. 또한 재난 경로 설정에서 각각의 노드의 연결 상황과 비용이 미리 알려져 있지 않은 경우에는 이 방식들을 사용할 수 없다는 단점

이 있다. 네트워크의 트래픽을 고려한 최단 경로를 찾기 위해 Q-Learning 기반의 라우팅 방식인 Q-routing을 이용한 연구도 진행되어왔다. Q-routing은 실시간으로 변화하는 트래픽에서 최단 경로를 찾기 위해 현재 위치에서 목적지까지 걸리는 시간을 최소화하도록 Q-table을 업데이트하게 된다.^[9] 일반적인 Q-routing에서 Q-table을 업데이트하는 수식은 Q값의 수렴이 보장되지 않기 때문에 최적 경로를 찾지 못할 수도 있다는 단점이 있다.

이를 고려하여 본 논문은 Q-Learning을 이용한 최적 경로를 탐색하는 알고리즘을 제안한다. Q-learning에서는 동적 변화에 대응할 수 있고 노드간의 연결과 비용이 미리 알려지지 않은 경우에도 경로를 파악할 수 있으며 사람들의 이동에 대한 보상을 실시간으로 업데이트 할 수 있다는 장점이 있다. 또한 기존의 길 찾기 알고리즘의 경우에는 node의 수에 따라 계산량이 최대 $O(n^2)$ 으로 증가하고 우선순위를 사용했을 때에도 $O(n \log n)$ 의 비용이 들기 때문에 학습 환경이 커지는 경우 메모리가 많이 소비된다는 단점이 있지만 Q-Learning의 경우에는 반복(iteration)을 사용하기 때문에 보다 효율적이다.

Ad-hoc 네트워크에서 효율적인 라우팅을 위해 강화 학습이 많이 적용되어왔다. Q. Yang은 데이터 패킷 전달에 최적 경로를 산출하기 위해 Q-Learning을 이용하였다.^[10] 이는 데이터 통신을 위해 주변 노드의 상황을 고려한 최적 경로 설정으로 본 논문에서는 데이터 통신 상황이 아닌 재난 상황에서 최적 경로를 산출하기 위한 시스템 모델을 제시한다. 기존 알고리즘 기반 대부분 피난 분석 프로그램은 사고 위치를 고려하지 않고 단순히 현재 위치로부터 최단 경로를 탐색하는 기법을 사용해왔다. 이에 W.O. Kim은 Q-Learning을 이용하여 해상 사고 발생 시 최적의 퇴선 경로 산출 방법에 대해 연구하였다.^[11] 이는 선박의 구조에 가중치(reward)를 부여하여 사용자(agent)가 사고 위치를 피해 경로를 탐색하도록 진행된다. 이는 단일층을 고려한 최적의 퇴선 경로 산출 방법이며 입체적인 구조를 고려한 최적 경로에는 부적합하다는 단점이 있다. Z.P. Su는 Q-Learning 기반의 경로를 선택하는 알고리즘을 제시했다.^[12] 이는 Markov Decision Process (MDP)를 이용하여 최적 경로를 추론한다. 이들이 제시하는 알고리즘은 환경을 안전한 장소, 위험한 장소, 접근이 불가능한 장소로 구분하고 장소에 따라 받는 보상(reward)을 다르게 지정한다. 대부분의 연구에서 경로를 산출할 때 신속한 탈출 시간만을 고려하여 경로를 생성한다.

J.G. Kim은 강화학습을 이용하여 탈출 시간뿐만 아니라 혼잡도까지 동시에 고려한 대피 경로를 생성하는 알고리즘을 제안하였다.^[13] 이 알고리즘에서는 게임 이론에 기반을 두어 개별 에이전트를 위한 경로 산출이 아닌 다수 에이전트를 고려하는 대피 경로를 산출한다. 이 알고리즘을 사용하여 탈출 경로를 파악하는 경우, 각각의 에이전트를 위한 Q-table을 생성하기 때문에 메모리가 많이 사용된다는 문제가 있다. 또한 학습 시 경로를 찾기 위한 환경은 에이전트마다 동일하기 때문에 경로에 약간씩 차이가 생기지만 결국 모든 에이전트가 하나의 출구로 이동하는 결과가 나타난다. 이는 양방향 피난 원칙을 고려하지 않은 경로이기 때문에 에이전트가 많은 경우, 출구 근처에서 혼잡이 발생할 가능성이 존재한다. 이러한 기존의 Q-Learning을 이용한 경로 찾기 알고리즘들은 입체적인 건물 구조를 고려하지 않고 하나의 층에서 하나의 출구로 가는 최단 경로를 찾는 것에 집중해왔다. 하지만 각각의 층에서 최단으로 인식했던 경로가 전체적인 경로를 고려했을 때에는 최단이 아닐 수도 있다. 따라서 본 논문에서는 이러한 건물의 입체적인 구조를 고려한다. 또한 양방향 피난 원칙을 고려하기 위해 출구에 따라 받는 보상을 다르게 지정하여 우선순위를 부여하고 사용자의 수에 따라 경로를 다르게 선택하여 출구 근처의 혼잡을 방지한다.

III. Q-Learning 기반 최적 탈출 경로 결정 모델

사용자의 현재 위치에서 최적의 탈출 경로를 찾기 위한 작업 수행 과정은 그림 1과 같다. 화재의 위치와 건물 구조에 대한 정보를 이용하여 학습을 위한 환경

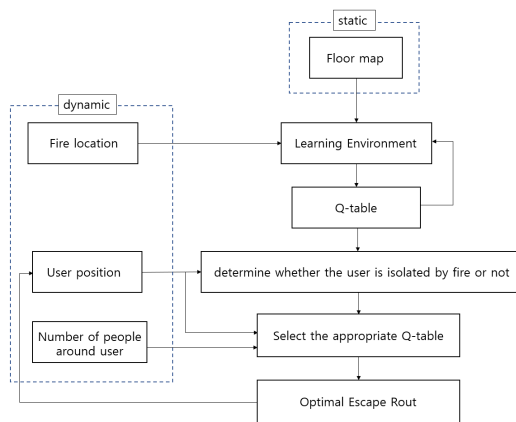


그림 1. 작업 수행 과정
Fig. 1. The process of performing

을 만들고 Q-Learning을 통해 경로 안내를 위한 Q-table을 구한다. 그리고 Q-table을 이용하여 사용자가 현재 위치에서 화재에 의해 고립되었는지 여부를 판단한다. 사용자가 화재에 의해 고립되었다고 판단된 경우에는 고립에 대한 경고를 띄우고 최단 경로를 안내하는 Q-table을 이용하여 사용자의 현재 위치에서 가장 가까운 출구까지의 경로를 표시한다.

사용자가 화재에 의해 고립되지 않았다고 판단된 경우에는 사용자의 주변 사람 수에 따라 적절한 Q-table을 선택하여 탈출 경로를 분산시킨다. 출구까지 경로는 단순히 하나의 층이 아닌 건물 전체에서의 경로를 고려한다. 이렇게 구한 경로를 다시 사용자에게 전달하고 최종적으로 사용자가 건물에서 탈출할 때까지 이 과정을 반복한다.

3.1 Q-Learning

화재가 발생한 건물에서 사용자로부터 가장 가까운 출구까지의 경로를 찾기 위해 강화학습의 한 종류인 Q-Learning을 사용한다.^[14]

Q-Learning은 Q 함수를 이용하여 현재 상태(State)에서 어떤 행동(Action)을 취하는 것이 최적인지 학습을 통해 판단하는 방식이다. 이때 학습은 보상(Reward)을 최대화하는 방향으로 진행된다. 본 논문에서 사용한 Q 함수는 수식(1)과 같다.

$$Q(s_t, a_t) = (1 - \alpha)Q(s_t, a_t) + \alpha[r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})] \quad (1)$$

수식에서 s_t 와 a_t 는 각각 시간 t 에서의 상태와 행동을 의미하고, r_t 는 행동 a_t 에 의해 미래에 얻어지는 보상을 의미한다.^[15] 또한 α 는 학습률(Learning rate)이고 γ 는 보상 할인율(Discounted factor)이다.^[16] α 는 학습의 강도를 조절한다.^[17] α 를 1로 설정하면 새로운 정보에 대한 학습 강도가 커져 미래에 대한 보상인 r_{t+1} 에 가중치를 두게 된다. 반면에 α 를 0으로 설정하면 초기 추정값에 가중치를 두기 때문에 새로운 행동

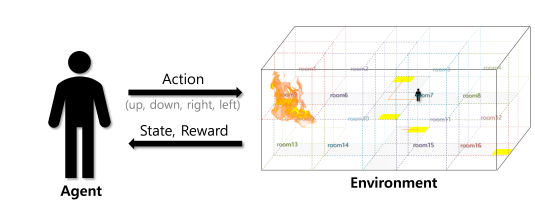


그림 2. Q-learning Agent & Environment
Fig. 2. Q-learning Agent & Environment

으로부터 학습이 이루어지지 않는다. γ 는 수식의 합이 발산하지 않도록 보장하는 상수로 0에서 1까지의 범위를 갖는다.^[18] 만약 값이 0에 가까워지면 현재의 보상에 가중치를 두고 1에 가까우면 미래의 보상에 가중치를 둔다.

3.2 역전파를 이용한 최단 경로 산출 Q-Learning 모델

본 논문은 다층 건물 구조에서 역전파를 이용하여 최단 경로를 산출할 수 있는 Q-Learning 모델을 제안한다. 사용자가 현재 상태에서 취할 수 있는 행동은 위, 아래, 오른쪽, 왼쪽 4가지의 이동이며, 선택한 행동에 따라 다음의 상태가 정해진다.

$$r = \begin{cases} r_{ua}, & \text{if the area type is unaccessible area} \\ r_r, & \text{if the area type is road} \\ r_i, & \text{if the area type is exit} \end{cases} \quad (2)$$

Environment를 구성하는 것은 사용자의 상태에 따라 받게 되는 보상 r 을 지정하는 것과 같다. 수식(2)는 사용자가 현재 상태에서 어떤 행동을 취했을 때 받게 되는 보상에 대해 수식으로 표현한 것이다. 만약 사용자의 다음 상태가 진입할 수 없는 공간일 경우 보상 r_{ua} 에 큰 불이익을 주어 사용자가 그 공간에 들어가지 않도록 한다. 다음 상태가 길이 되는 경우에도 보상 r_r 에 작은 불이익을 주어 불필요한 경로를 선택하는 것을 막고 최단 경로를 찾는 Q-table을 만들 수 있도록 한다. 다음 상태가 출구일 경우 받는 보상 r_i 은 층과 출구의 위치에 따라 다르게 지정한다. 그리고 사용자의 위치를 아래층으로 이동시킨다.

그림 3은 r_i 을 결정하는 방법에 대한 흐름도이다. 1층에서 사용자의 다음 상태가 출구일 때에는 R_e 의 보상을 얻게 된다. 2층 및 그 이상의 출구에 대한 보상을 정할 때는 아래층에서 해당 출구 위치의 최대 Q값을 사용한다. 아래층에서 각각의 출구 위치가 갖는 최대 Q값들 중 가장 작은 값을 m 으로 설정한다. m 이 음수인 경우에는 p 를 $-m + R_e$ 로 지정하고 양수인 경우에는 p 를 0으로 지정한다. 이렇게 정한 p 를 최대 Q값에 더해 r_i 로 지정하면 m 이 음수인 경우엔 r_i 의 최솟값이 3이 되고 양수인 경우에는 최대 Q값이 그대로 보상이 된다.

이렇게 구성된 환경에서 학습을 진행하면 그림 4와 같이 아래층의 출구와 가까운 출구에 어드벤처지가 생

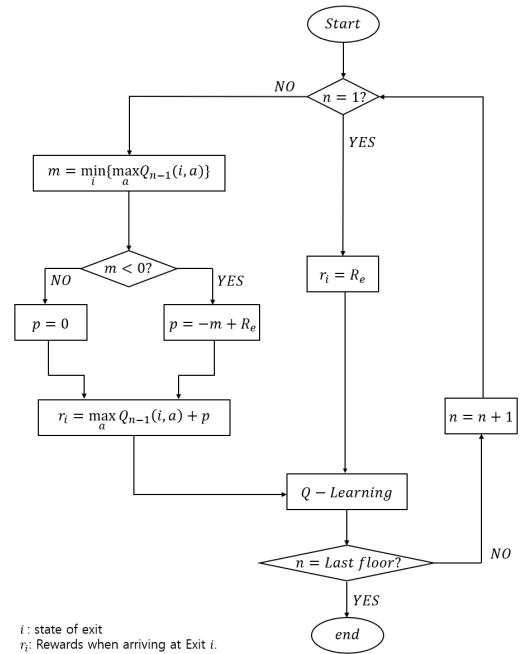


그림 3. Q-learning 흐름도
Fig. 3. Q-learning flow chart

겨 건물 전체 구조를 고려한 최단 경로를 예측할 수 있다. 단, 현재 위치한 층에서 출구로 내려갔을 때 아래층의 화재에 의해 고립되는 경우에는 해당 출구의 보상을 정할 때 최대 Q값을 그대로 사용한다. 또한 출구에 화재가 발생한 경우에는 r_{ua} 의 보상을 사용한다. 그림 4에서 파란색으로 표시된 경로는 각각의 층에서 최단 경로로 이동했을 때의 경로이고 주황색은 건물 전체 구조를 고려하여 이동했을 때의 경로이다.

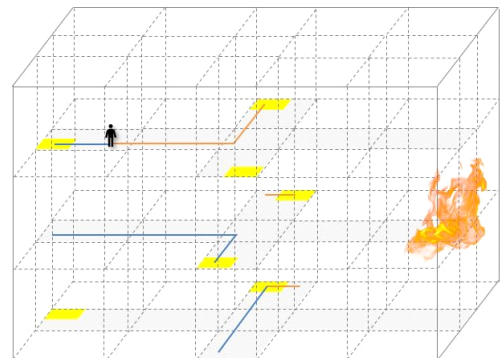


그림 4. 최단 경로 이미지
Fig. 4. The shortest path image

3.3 사용자 주변 사람의 밀집 정도 고려

피난 대책의 일반적인 원칙에는 건물에 양방향의 피난로를 확보 한다는 원칙과 피난 유도 시에는 병목현상이 발생하지 않도록 분산 유도를 한다는 원칙이 존재한다.

그림 5의 (a)와 같이 사용자의 현재 위치 주변에 사람이 많은 경우, 모든 사람에게 최단 경로를 안내하면 병목 현상으로 인해 출구 근처에서 혼잡이 발생해 탈출이 원활하지 않을 수 있다는 점을 고려한다.

사용자의 주변에 사람이 많은 경우에는 그림 5의 (b)와 같이 사용자의 현재 위치 주변 사람의 수에 따라 사용할 수 있는 출구들 중 가장 가까운 출구와 그 다음으로 가까운 출구로 분산시켜서 경로를 안내한다.

현재 위치에서 두 번째로 가까운 출구로 가는 경로 찾는 Q-table을 얻기 위해 학습을 위한 환경을 수정한다. 현재 위치에서 가장 가까운 출구에 도착했을 때 큰 불이익을 주도록 보상을 수정한 환경에서 학습을 진행한다. 사용자의 현재 위치 주변에 사람이 많은 경우 분산시켜 경로를 안내하기 위해 가장 가까운 출구로 가는 Q-table과 그 다음으로 가까운 출구로 가는 Q-table을 사용한다. 그리고 수식 (3)을 통해 경로를 안내할 최종 Q-table을 선택한다.

$$c = \begin{cases} -0.5\log_{10}N+1 & (N < 10) \\ 0.5 & (N \geq 10) \end{cases} \quad (3)$$

c 는 현재 위치 주변에 있는 사람 수 N 에 따라 가장 가까운 출구로 가는 Q-table을 선택할 확률이다. 따라서 (1-c)는 두 번째로 가까운 출구로 가는 Q-table을 선택하는 확률이 된다. 수식에 따라 현재 위치 주변에

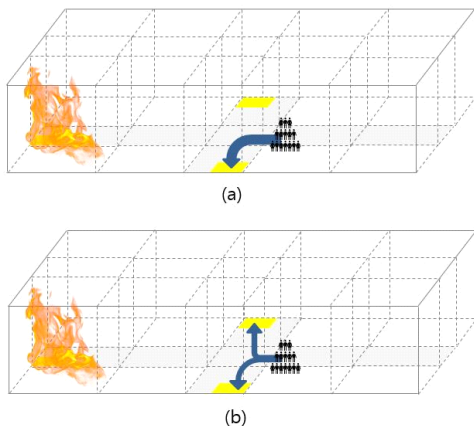


그림 5. 분산된 경로 이미지
Fig. 5. the distributed path image

사람의 수가 적은 경우에는 높은 확률로 가장 가까운 출구로 가는 Q-table을 선택한다. 사람의 수가 증가함에 따라 두 번째로 가까운 Q-table을 선택할 확률이 점점 높아지다가 N 이 10 이상이 되면 두 Q-table을 선택할 확률이 각각 50%가 되어 효율적으로 사람들을 분산시킬 수 있다.

3.4 화재에 의한 특수 상황 고려

화재가 발생한 위치에 따라 사용자의 현재 위치에서 출구로의 이동이 불가능할 수 있다는 점을 고려한다. 사용자가 현재 위치에서 화재로부터 고립되었는지 여부에 대해 시스템 내에서 판단할 수 있도록 한다. 사용자가 화재에 의해 고립되지 않았다고 판단된 경우에는 현재 위치에서 최적의 경로를 표시한다. 화재에 의해 고립되었다고 판단된 경우에는 화재에 의해 고립되었다는 경고 알림을 띄우고 현재 위치에서 최단 경로를 표시한다. 화재에 의해 고립되었는지 여부를 판단하기 위해 두 가지의 경우를 고려한다.

첫 번째로 화재에 의해 고립되어 사용자의 이동이 불가능한 경우에 대해서 고려한다. 그림 6의 (a)는 사용자가 화재에 의해 주변이 완전히 둘러싸여 이동이 불가능한 경우와 그때 Q-table을 나타낸 그림이다. 그림에 나타난 Q-table은 사용자의 현재 위치 ps 에서 취할 수 있는 행동을 했을 때 얻게 되는 보상을 나타낸다. 사용자가 취할 수 있는 행동 a_1, a_2, a_3, a_4 는 순서대

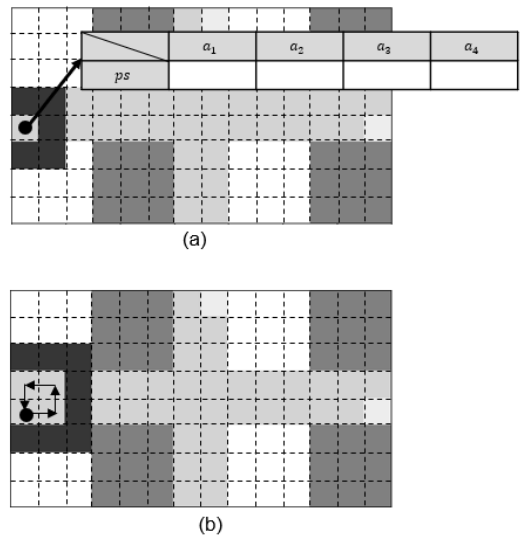


그림 6. 화재에 의한 고립 (a) 사용자가 움직일 수 없는 경우 (b) 사용자가 움직일 수는 있지만 화재로부터 탈출할 수 없는 경우
Fig. 6. isolated by fire (a) agent cannot move (b) agent can move but cannot escape from a fire

로 위, 아래, 오른쪽, 왼쪽 이동을 의미한다. 그림 6의 (a)와 같은 상황에서 Q-table을 이용하여 경로를 찾게 되면 Q-table의 최댓값에 따른 다음 상태가 화재 혹은 사용자가 진입할 수 없는 공간이 된다. 이를 막기 위해 사용자의 현재 위치가 ps 일 때 Q-table에서 가장 큰 값이 Q_s 보다 작은 경우 화재에 의해 고립되었다고 판단한다.

두 번째로 사용자의 이동이 가능하지만 화재에 의해 고립되어 사용자의 이동이 불가능한 경우에 대해서 고려한다. 그림 6의 (b)는 현재 위치에서 사용자의 이동은 가능하지만 화재에 가로막혀 출구로는 갈 수 없는 경우를 나타낸 것이다. 이러한 상황에서 Q-table의 최댓값을 이용하여 경로를 나타낼 경우 그림과 같이 같은 자리만 맴돌고 출구에는 도착하지 못 하는 문제가 발생한다. 따라서 count factor를 추가하여 고립에 대해 판단한다. count factor의 초기 값은 0으로 설정하고 사용자의 현재 위치에서 다음 위치로 이동할 때마다 1씩 증가시킨다. count factor가 환경에서 길에 해당되는 상태의 수보다 커지면 현재 위치에서 화재에 의해 고립되었다고 판단한다.

IV. 모의 실험

입의의 환경에서 최적의 탈출 경로 안내를 위한 Q-Learning 기반 시스템 모델의 성능을 확인하기 위한 시뮬레이션을 진행하였다. 성능에 대해서 확인하기 위해 두 가지 실험을 진행하였다. 첫 번째로 최단 경로 안내가 제대로 이루어지는지 확인하기 위해 다른 두 비교 모델과 함께 시뮬레이션 결과를 확인하였다. 두 번째로 출구에서 혼잡을 방지하기 위해 분산을 사용했을 때의 성능을 확인하기 위한 시뮬레이션을 진행하였다. 시뮬레이션은 파이썬(python)을 이용하여 구현하였으며 파이썬 내에 내장된 라이브러리인 matplotlib을 이용하여 그래픽 유저 인터페이스(GUI)를 구현하였다.

시뮬레이션을 위한 환경은 표 1과 같다. 시뮬레이션

표 1. 환경 파라미터
Table 1. environment parameter

parameter	value
column	14
row	8
number of cell (per floor)	112
number of floor	2
number of BLE beacon	18

환경은 가로로 14개, 세로로 8개의 셀로 각각 112개의 셀을 갖는 층 3개로 구성된다. 그림 7과 같이 1층의 왼쪽 맨 위에서부터 셀마다 0에서 111까지의 번호를 붙였고 2층의 왼쪽 맨 위에서부터 112에서 223, 3층의 왼쪽 맨 위에서부터 224에서 335까지의 번호를 붙였다.

각각의 셀은 가로로 3m, 세로로 3m의 크기를 갖는다고 가정한다. 각각의 셀은 방, 길, 출구, 화재로 구분한다. 하늘색과 분홍색으로 각각의 방을 구분하였고 회색으로 길을 나타내었다. 노란색은 출구이며 붉은색은 화재의 위치를 의미한다.

사용자의 현재 위치에서 최적의 탈출 경로를 찾기 위한 입력값으로 화재의 위치, 사용자의 위치에 대한 정보가 필요하다. 따라서 본 논문은 화재 위치를 임의

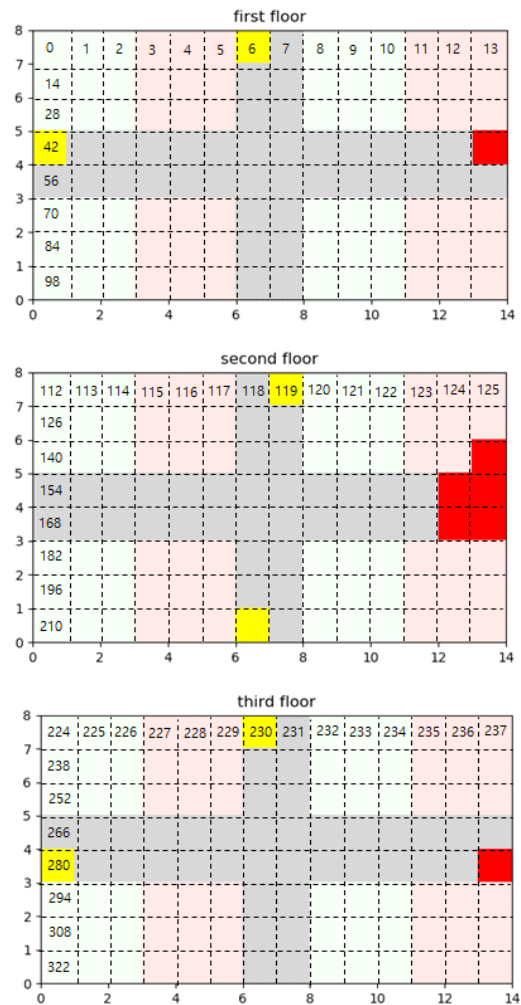


그림 7. 시뮬레이션 환경
Fig. 7. Simulation environment

의 고정된 장소라고 가정한다. 그리고 BLE 비콘과 사용자의 스마트폰까지 거리에 대한 RSSI(Received Signal Strength Intensity)를 이용하여 사용자의 위치를 추정한다. 실내의 환경에 따라 발생하는 RSSI 오차를 고려하여 사용자의 위치를 정확하게 추정하기 위해 심층 신경망(DNN : Deep Neural Network) 모델을 이용한다. 심층 신경망은 입력층과 출력층 사이에 여러 개의 은닉층으로 이루어진 인공신경망을 의미한다. 심층 신경망은 비콘으로부터 받은 RSSI값을 입력으로 했을 때 사용자의 위치를 나타내기 위한 좌표 (x, y, z) 값을 출력한다. 심층신경망에 사용된 파라미터는 표 2와 같다.

학습 시에 은닉층으로 3개를 사용하였고 각각 은닉층의 노드 수는 800, 600, 400을 사용하였다. 학습에 사용한 데이터 중 80%는 학습 데이터셋으로, 20%는 테스트 데이터셋으로 사용하였으며 오버피팅을 방지하기 위해 드롭아웃(drop out)을 사용하였다. 활성화 함수(Activation function)으로는 ReLU를 사용하였고 최적화를 위하여 Adam optimizer를 사용하였다. 심층신경망에 사용된 손실함수는 수식 (4)와 같다.

$$\text{Loss} = \sqrt{(x_t - x_p)^2 + (y_t - y_p)^2 + (z_t - z_p)^2} \quad (4)$$

실제 사용자의 위치 (x_t, y_t, z_t) 와 예측된 사용자의 위치 (x_p, y_p, z_p) 를 이용하여 두 지점 사이의 거리를 손실 함수로 사용하였다. 심층 신경망을 이용하여 사용자의 위치를 추적하면 오차 1.5m 이내의 결과를 찾을 수 있다. 이렇게 구한 사용자의 위치를 셀 단위로 변환시켜 Q-Learning에 사용한다. Q-Learning을 위한 파라미터는 표 3의 값을 사용하여 구현하였다.

사용자가 출구에 도착하거나 사용자가 접근할 수 없

표 2. 심층신경망 파라미터
Table 2. DNN parameter

parameter		value
Node	Hidden layer 1	800
	Hidden layer 2	600
	Hidden layer 3	400
Learning rate		0.1
Training dataset rate		0.8
Test dataset rate		0.2
Drop out rate		0.3
Activation function		ReLU
Optimizer		Adam

표 3. Q-Learning 파라미터
Table 3. Q-Learning parameter

parameter		value
α		0.7
γ		0.85
Iteration number		3000
Epsilon(ϵ)		$\frac{1}{(I/10+1)}$
Q_{is}		-70
r	r_{ua}	-100
	r_r	-2
	R_e	3

는 곳에 도착했을 경우 에피소드를 종료한다. Q-table을 얻기 위해 3000번의 에피소드를 반복한다. 사용자는 현재 위치에서 위, 아래, 오른쪽, 왼쪽 네 가지의 행동을 취할 수 있다. 사용자가 행동을 취해서 다음 상태가 되었을 때 방에 해당되는 곳인 경우, 시뮬레이션 환경에서 벗어나는 곳인 경우, 화재일 경우에는 r_{ua} 로 -100의 보상을 주었다. 또한 사용자의 다음 상태가 길일 경우에는 r_r 로 -2의 보상을 주었다. 출구에 대한 보상은 층과 출구의 위치에 따라 다르게 지정하였다. R_e 는 3으로 설정하였다. 학습 시에 사용자의 현재 위치가 방인 경우에는 사용자가 방의 입구까지는 무사히 이동할 수 있다고 가정하여 방의 입구로 사용자의 위치를 옮겨서 학습을 진행하였다.

학습을 통하여 Q-table을 만들 때에는 최적의 경로를 찾기 위해서 충분히 다양한 경로를 탐험(Exploration)하는 과정이 필요하다. 따라서 학습을 진행할 때 에이전트가 다음 행동을 선택함에 있어 무작위성을 추가한 방식인 decaying epsilon-greedy를 적용한다.^[19] decaying epsilon-greedy를 수식으로 표현하면 다음과 같다.

$$a = \begin{cases} \text{argmax } Q(s, a_{t+1}), & \text{with probability } (1 - \epsilon) \\ \text{random } a \in A, & \text{with probability } \epsilon \end{cases} \quad (5)$$

수식 (5)에서 ϵ 은 행동을 무작위로 선택할 확률로 0에서 1 사이의 값을 갖는다. ϵ 이 높으면 행동을 선택할 때 축적된 정보에 새로운 정보가 추가될 가능성이 높아진다. 낮은 경우에는 이전에 축적된 정보만을 이용하여 다음 행동을 선택하게 된다. 이때 다양한 경로에 대한 학습이 충분히 이루어지면 탐험을 하는 것이 의미

가 없기 때문에 몇 번째 iteration인지에 따라 epsilon의 값을 작게 만들어 무작위한 행동을 선택할 확률이 낮아지게 한다. 이를 decaying epsilon-greedy 방식이라고 하고 시뮬레이터는 epsilon의 값으로 $\frac{1}{(I/10+1)}$ 을 사용하여 iteration I에 따라 값이 결정되도록 하였다.

사용자의 위치에 따라 적절한 Q-table을 선택해 경로를 안내할 수 있도록 한다. 만약 사용자의 현재 위치가 2층이라면 해당 층에서 최적 경로를 안내하는 Q-table을 이용하여 해당 층의 출구까지 경로를 표시한다. 해당 층의 출구에 도착하는 경우 사용자의 위치를 아래층으로 이동시켜 다시 해당 층에서 최적 경로를 안내하는 Q-table을 이용하여 출구까지 경로를 표시한다.

Q-Learning을 위한 시뮬레이션 환경을 구성한 뒤 첫 번째로 3층 건물 구조에서 최단 경로 안내가 제대로 이루어지는지 확인하였다. 그림 8은 2층에서부터 최종 출구까지 두 모델의 최단 경로를 비교한 것이다. 주황색으로 표시된 경로는 본 논문이 제안하는 Q-Learning을 사용한 결과이고 파란색으로 표시된 경로는 각각의 층에서 최단 경로로 이동하는 Q-Learning 모델을 사용한 결과이다.

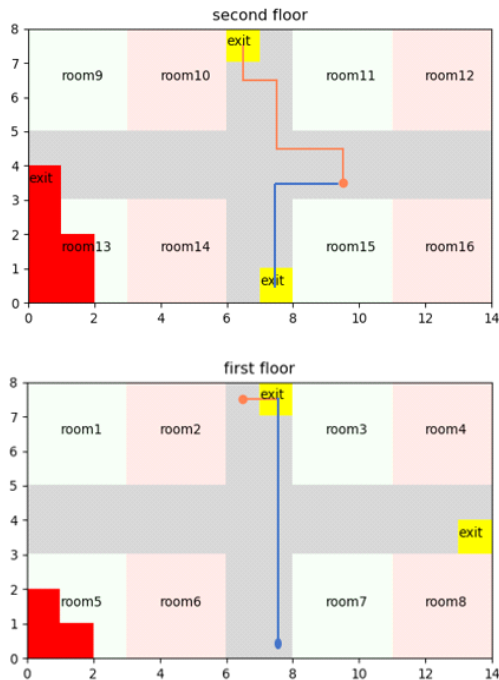
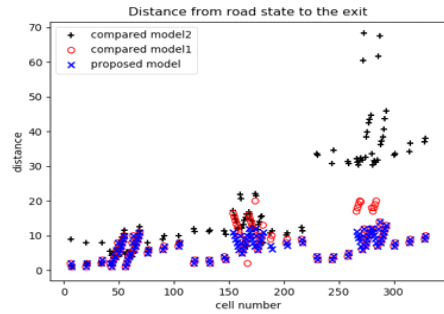
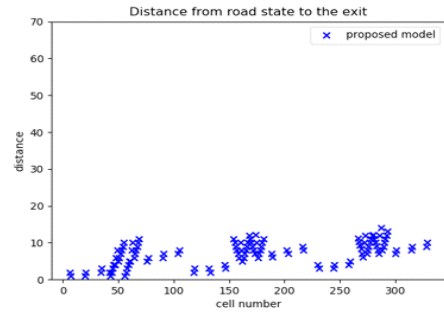


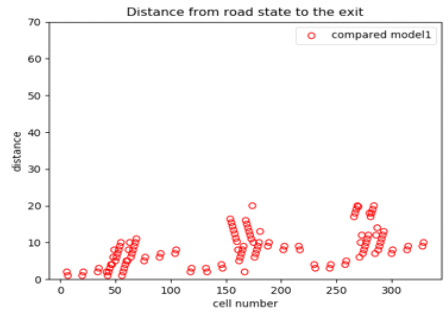
그림 8. 최단 경로 비교
Fig. 8. compare the shortest path



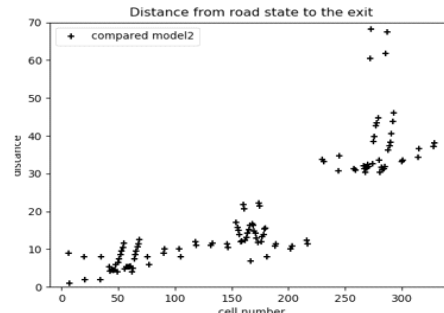
(a)



(b)



(c)



(d)

그림 9. Q-Laerning 모델 성능 비교 (a) 모든 결과를 합친 그래프, (b) 제안 모델의 결과, (c) 비교 모델1의 결과, (d) 비교 모델 2의 결과

Fig. 9. compare Q-Learning model performance (a) the result of all models together, (b) the result of proposed model, (c) the result of compared model1, (d) the result of compared model2

각각의 층만을 고려한 모델의 경우 건물의 전체 구조를 고려하지 않기 때문에 비효율적인 경로를 출력하는 것을 확인할 수 있다. 이에 비해 본 논문이 제안하는 Q-Learning 모델은 건물의 전체 구조를 고려하여 효율적인 최단 경로를 제대로 출력하는 것을 확인할 수 있다.

그림 9는 그림 7과 같은 3층 구조의 시뮬레이션 환경에서 길에 해당하는 모든 state에 대해 최종 출구에 도착할 때까지 거쳐 간 state의 수를 비교한 그래프이다. 그래프의 가로축은 셀의 번호를 의미하고 세로축은 각각의 셀에서 최종 출구까지 거쳐 간 state 수를 의미한다. 가로축에서 0부터 111까지는 1층에서의 결과이고 112부터 223까지는 2층, 224부터 335까지는 3층에서의 결과이다. (b)는 proposed model에 대한 결과로, 파란색 'x'로 표시된 그래프는 본 논문이 제안하는 Q-Learning 모델을 사용했을 때의 결과이다. (c)는 compared model1에 대한 결과로, 빨간색 'o'로 표시된 그래프는 각각의 층만을 고려하여 최단 경로를 출력하는 Q-Learning 모델을 사용했을 때의 결과이다. (d)는 compared model2에 대한 결과로, 검정색 '+'로 표시된 그래프는 각각의 층에서 출구를 무작위로 선택했을 때의 결과이다. 무작위로 경로를 선택했을 경우, 우연히 성능이 안 좋게 나올 수 있으므로 100회 진행하고 이를 평균을 내어 결과를 확인하였다. (a)는 이를 모두 합쳐서 나타냈을 때의 결과이다.

표 4는 세 모델에 대해서 각각의 state에서 최종 출구에 도착할 때까지 거쳐 간 state의 수를 이용하여 평균과 분산을 계산한 결과이다. 평균과 분산은 1층에서 탈출하는 경우(cell number : 0~111), 2층에서 탈출하는 경우(cell number : 112~223), 3층에서 탈출하는 경우(cell number : 224~335), 전체 state (cell number : 0 ~335)에 대해 평균과 분산을 계산하였다. 본 논문이 제안하는 모델인 proposed model과 각각의 층만을 고려하여 최단 경로를 출력하는 모델인 compared model1을 비교하면 1층에서는 같은 결과를 내는 것을 확인할 수 있다. 하지만 2층의 경우에는 평균적으로 proposed model이 2.31개의 state를 단축시키고 3층의

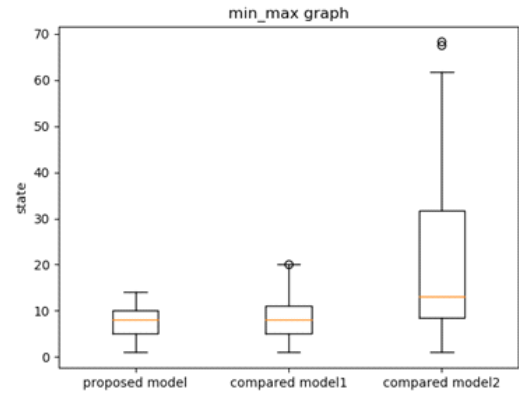


그림 10. Q-Learning model들의 min max 그래프
Fig. 10. min max graph of Q-Learning models

경우에는 2.39개의 state를 단축시키는 것을 확인할 수 있다. 각각의 층에서 무작위로 출구를 선택하여 경로를 안내하는 compared model2와 proposed model을 비교했을 때 proposed model이 전체 평균 12.32개의 state를 단축시키고 2층의 경우에는 6.07개, 3층의 경우에는 29.19개까지 단축시키는 것을 확인할 수 있다. 모든 경우에 대해서 proposed model의 분산은 9.71, compared model1의 분산은 23.49, compared model2의 분산은 219.78으로 compared model들의 분산이 상대적으로 더 크다는 것을 확인할 수 있다.

또한 그림 10의 그래프를 통해 일부 state에서 compare model이 출구까지 더 먼 경로를 선택하는 것을 확인할 수 있다. 층의 구조가 복잡해질수록 성능의 격차는 커져 본 논문이 제시하는 모델의 성능이 두드러질 것으로 예상된다.

두 번째로 출구에서 혼잡을 방지하기 위해 분산을 사용했을 때의 성능을 확인하기 위한 시뮬레이션을 진행하였다. 이를 위해 사용자가 최단 경로만 선택하는 경우와 분산된 경로를 선택하는 경우에 대해서 시뮬레이션을 진행한다.

그림 11의 (a)는 21명의 사람이 그림과 같은 분포에서 사용자의 현재 위치에서 최단 경로로만 이동했을 때의 경로에 대해 나타낸 것이다. 분산된 경로를 선택

표 4. 평균과 분산
Table 4. mean and variance

	1 st floor		2 nd floor		3 rd floor		All state	
	mean	var	mean	var	mean	var	mean	var
proposed	5.35	8.43	7.66	6.92	8.81	7.55	7.27	9.71
compared1	5.35	8.43	9.31	19.08	11.02	26.06	8.56	23.49
compared2	6.83	7.69	13.73	11.78	38.20	96.24	19.59	219.78

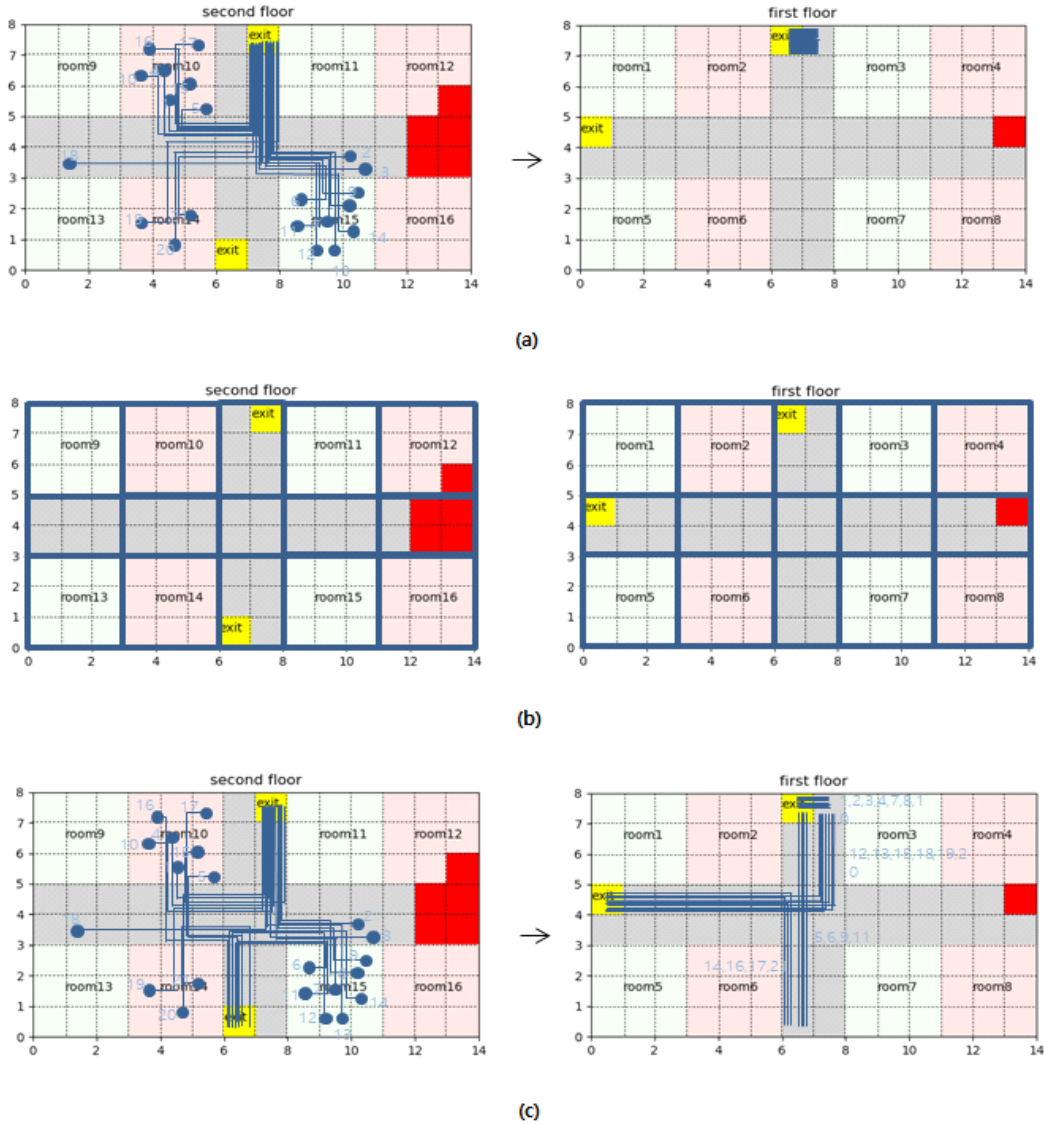


그림 11. (a) 최단 경로만 선택하는 경우, (b) 시뮬레이션 환경 나누는 격자 모양, (c) 분산된 경로를 선택하는 경우
 Fig. 11. (a) only shortest path select, (b) divide simulation environment into lattice, (c) distributed path select

하는 기준을 나누기 위해서 시뮬레이션 환경을 그림 11의 (b)와 같이 구분한다. 같은 격자 안에 있는 사람의 수에 따라 수식(3)을 이용하여 경로를 선택한다. 그림 (c)는 (a)와 같은 분포에서 사람의 수에 따라 분산된 경로를 선택했을 때의 경로에 대해 나타낸 것이다. 사용자의 위치에서 최단 경로로만 이동한 경우와 분산된 경로를 선택한 경우에 대해서 성능을 비교한다.

그림 12의 가로축은 21명의 각각의 사용자를 의미하고 탈출까지 걸린 비용을 의미한다. 사용자의 위치에서 출구까지 state를 이동할 때마다 0.1의 비용을 부여

하고 사용자가 출구에 도착한 순서대로 0에서부터 1씩 증가하는 비용을 부여받는다. 그림 12의 파란색 'x'는 각각의 사용자가 최단 경로로만 이동한 경우이고 빨간색 '+'는 같은 위치에서 분산된 경로를 선택했을 때의 경우이다. 최단 경로로만 이동했을 때의 비용은 평균적으로 20.87, 분산된 경로를 선택했을 때의 비용은 평균적으로 8.83으로 분산된 경로를 선택했을 때 12.04의 적은 비용이 드는 것을 확인할 수 있다.

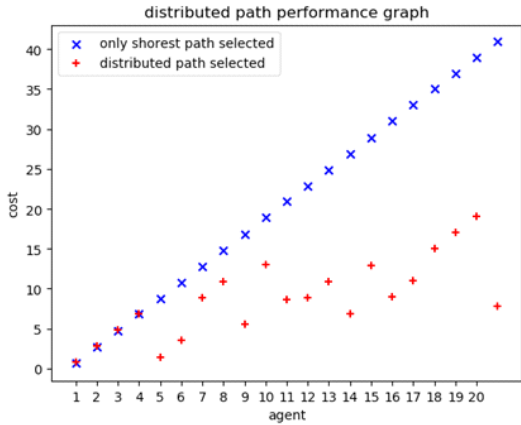


그림 12. 분산된 경로의 성능
Fig. 12. distributed path performance

V. 결론

건물에 화재가 발생했을 때 건물의 구조가 익숙하지 않은 사람이 실내에 있는 경우 가장 짧은 경로를 찾는 데 걸리는 시간을 줄이는 것이 중요하다. 따라서 본 논문은 건물에 화재가 발생했을 때 사용자의 현재 위치에서 최적의 탈출 경로를 찾을 수 있는 Q-Learning 기반 시스템 모델을 제안했다. 기존의 Q-learning 방식에서 다층 건물의 최단 경로를 찾는 방법은 점의 대칭 이동을 이용한 최단 거리 구하는 것과 같다. 이 방식으로 다층 건물의 최단 경로를 찾는 경우 건물의 층이 증가함에 따라 학습을 위한 환경의 크기가 기하급수적으로 증가하는 문제가 발생한다. 따라서 본 논문에서는 층에 따라 출구의 보상을 다르게 지정하는 방법을 사용하여 학습에 사용되는 메모리와 걸리는 시간을 절약할 수 있도록 하였다. 또한 한 공간에 사람이 많은 경우 모두가 최단 경로만 선택했을 때 병목 현상에 의해 원활한 탈출이 불가능하다는 점을 고려하였다. 문제를 해결하기 위해 사용자 주변의 사람 수에 따라 최적 경로와 차선 경로로 사용자들을 분산시켜 원활하게 경로를 안내할 수 있도록 하였다. 마지막으로 화재에 의해 고립된 특수한 경우에 대해서도 고려하였다. 사용자가 고립되었다고 판단된 경우에는 고립되었다는 알림을 띄우고 최단 경로를 표시하도록 하였다. 제안된 모델이 실질적인 최단 경로를 선택한다는 것을 확인하기 위해 각각의 셀에서 최단 경로까지 걸리는 state 수에 대해 다른 모델들과의 성능 비교를 진행했을 때 보다 향상된 결과를 내는 것을 확인하였다. 또한 분산된 경로를 선택했을 경우와 최단 경로만 선택하는 경우에 대해서 성능을 비교하여 더 나은 성능을 보이는 것을 확인하였다.

References

- [1] A. Khan, A. A. Aesha, J. S. Aka, S. M. F. Rahman, and M. J. Rahman, "An IoT based intelligent fire evacuation system," *2018 21st IEEE ICCIT*, pp. 1-6, 2018.
- [2] J. Jeong, S. Yeon, T. Kim, H. Lee, S. Kim, and S. Kim, "SALA: Smartphone-assisted localization algorithm for positioning indoor iot devices," *Wireless Netw.*, vol. 24, pp. 27-47, 2018.
- [3] R. Vancetti and S. Angelini, "Fire safety engineering: Evacuation management in emergency conditions," *TEMA: Technol., Eng., Mater. and Architecture*, vol. 4, no. 1, pp. 127-136, 2018.
- [4] H. J. Oh, Y. J. Kim, B. N. Kim, D. G. Choi, and J. H. Jeong, "Internet-of-things based fire detection and indoor evacuation navigation system," in *Proc. Symp. KICS*, pp. 280-281, 2018.
- [5] T. Y. Kim, M. S. Kim, and K. S. Chung, "Context-aware IoT system for real-time fire detection," in *Proc. Symp. KICS*, pp. 423-424, 2019.
- [6] Y. J. Koh, N. E. Go, C. Y. Kim, S. H. Lee, and S. S. Park, "Real-time fire evacuation system based on indoor location tracking and route optimization," in *Proc. Symp. Korea Inf. Sci. Soc.*, pp. 1659-1661, 2018.
- [7] J. S. Kim and K. W. Choi, "Indoor navigation using beacons," in *Proc. Symp. KICS*, pp. 1150-1151, 2019.
- [8] S. P. Jung, H. Y. Lee, J. Y. Song, and J. W. Jin, "Analysis of a path search algorithm for development of a realtime evacuation guidance system," in *Proc. Symp. Architectural Inst. of Korea*, pp. 140-141, 2019.
- [9] R. A. Haraty and B. Traoulsi, "MANET with the Q-Routing protocol," *ICN The 11th Int. Conf. Netw.*, 2012.
- [10] Q. Yang, S. J. Jang, and S. J. Yoo, "Q-learning-based fuzzy logic for multi-objective routing algorithm in flying ad hoc networks," *Wireless Pers. Commun.*, pp. 1-24, 2018.

2020.

- [11] W. O. Kim, D. H. Kim, and D. G. Youn, "Optimum evacuation route calculation using AI Q-Learning," *J. Korean Soc. Marine Environ. & Safety*, vol. 24, no. 7, pp. 870-874, 2018.
- [12] Z. P. Su, J. G. Jiang, C. Y. Liang, and G. F. Zhang, "Path selection in disaster response management based on q-learning," *Int. J. Automat. and Comput.*, vol. 8, no. 1, pp. 100-106, 2011.
- [13] J. G. Kim and H. S. Lee, "Multi-agent reinforcement learning based evacuation framework considering both evacuation time and crowdedness," *J. Korean Inst. of Intell. Syst.*, vol. 27, no. 4, pp. 334-341, 2017.
- [14] Q. Yang and S. J. Yoo, "Dynamic routing path selection algorithm using reinforcement learning in wireless adhoc networks," *J. KICS*, vol. 43, no. 7, pp. 1227-1235, 2018.
- [15] Y. J. Choi, J. H. Kwon, H. J. Na, and S. J. Yoo, "Q-learning based dynamic routing protocol with low latency and high reliability for medical data collection system using body area networks," *J. KICS*, vol. 44, no. 11, pp. 2090-2101, 2019.
- [16] M. A. Hossen and S. J. Yoo, "Q-learning based multi-objective clustering algorithm for cognitive radio ad hoc networks," *IEEE Access*, vol. 7, pp. 181959-181971, 2019.
- [17] J. Yan, H. He, X. Zhong, and Y. Tang, "Q-learning-based vulnerability analysis of smart grid against sequential topology attacks," *IEEE Trans. Inf. Forensics and Secur.*, vol. 12, no. 1, pp. 200-210, 2016.
- [18] J. Anand, R. P. Perinbam, and D. Meganathan, "Q-learning-based optimized routing in biomedical wireless sensor networks," *IETE J. Res.*, vol. 63, no. 1, pp. 89-97, 2017.
- [19] S. J. Jang, C. H. Han, K. E. Lee, and S. J. Yoo, "Reinforcement learning-based dynamic band and channel selection in cognitive radio ad-hoc networks," *EURASIP J. Wireless Commun. and Netw.*, vol. 2019, no. 1, pp. 1-25, 2019.

최 승 희 (Seung-Hee Choi)



2020년 2월 : 인하대학교 정보통신공학과 학사 졸업
 2020년 2월~현재 : 인하대학교 전기컴퓨터공학과 석사
 <관심분야> 무선 센서 네트워크, 머신러닝, IoT, 이동통신

유 상 조 (Sang-Jo Yoo)



1988년 2월 : 한양대학교 전자통신학과(공학사)
 1990년 2월 : 한국과학기술원 전기및전자공학과(공학석사)
 2000년 8월 : 한국과학기술원 전자전산학과(공학박사)
 1990년 3월~2001년 2월 : KT 연구 개발 본부

1990년 3월~2000년 11월 : NIST(미국 표준기술연구원) 초빙연구원
 2001년 3월~현재 : 인하대학교 정보통신공학과 교수
 <관심분야> 무선 네트워크 프로토콜, Cross-layer 프로토콜 설계, Cognitive Radio Network, 무선센서 네트워크, 미래인터넷
 [ORCID:0000-0003-1533-0814]