

탐욕 알고리즘 기반 다중 출구 대피경로 할당

An Evacuation Route Assignment for Multiple Exits based on Greedy Algorithm

이민혁* · 남현우** · 전철민***

Lee, Min Hyuck · Nam, Hyun Woo · Jun, Chul Min

要 旨

재난 발생 시, 실내 공간에서의 신속한 대피를 위해 총 대피시간 최소화를 목적으로 일부 연구들이 진행되었다. 하지만 대부분의 연구가 총 대피시간이 최소화되는 최적 대피경로를 산출하는데 오랜 연산시간이 소요되어 실제 재난 상황에 적용하기 어렵다는 한계를 가지고 있다. 이에 본 연구는 짧은 연산시간으로 총 대피시간을 단축시킬 수 있는 대피경로 할당 알고리즘을 제안하고자 한다. 대피경로 할당 알고리즘은 다수의 출구가 존재하는 건물에서 대규모 인원 대피 시, 각 출구에 적절하게 대피인원을 할당하여 출구들의 교통 정체 상황을 균형적으로 유지함으로써 총 대피시간을 단축시키는 알고리즘이다. 각 출구에 대피인원을 할당하는 방법은 그래프 이론을 기반으로 탐욕 알고리즘의 접근방식을 활용하였다. 본 연구에서는 알고리즘의 검증에 cellular automata 기반 대피 시뮬레이터를 이용하였으며 실제 건물과 유사한 구조에서 다양한 인원분포를 적용한 뒤 실험을 수행하였다. 결과적으로 최단거리 출구 대피보다 알고리즘을 적용하였을 때 총 대피시간이 감소되었고 대형 건물 구조에서도 짧은 연산시간이 소요되는 것을 확인하였다.

핵심용어 : 실내 대피, 총 대피시간, 최적 대피경로, 대피인원 할당, 탐욕 알고리즘

Abstract

Some studies were conducted for the purpose of minimizing total clearance time for rapid evacuation from the indoor spaces when disaster occurs. Most studies took a long time to calculate the optimal evacuation route that derived minimum evacuation time. For this reason, this study proposes an evacuation route assignment algorithm that can shorten the total clearance time in a short operational time. When lots of exits are in the building, this algorithm can shorten the total clearance time by assigning the appropriate pedestrian traffic volume to each exit and balances each exit-load. The graph theory and greedy algorithm were utilized to assign pedestrian traffic volume to each exit in this study. To verify this algorithm, study used a cellular automata-based evacuation simulator and experimented various occupants distribution in a building structure. As a result, the total clearance time is reduced by using this algorithm, compared to the case of evacuating occupants to the exit within shortest distance. And it was confirmed that the operation takes a short time in a large building structure.

Keywords : Indoor Evacuation, Total Clearance Time, Optimal Evacuation Route, Traffic Assignment, Greedy Algorithm

1. 서 론

안전에 대한 사회적 인식이 고조되면서 실내공간에서의 신속한 대피는 중요한 문제가 되었다. 대형 쇼핑몰 및 영화관 등에서는 대피시간의 단축을 위해 영상물

이나 게시판 등을 이용하여 최단거리에 위치한 출구를 안내하고 있다. 그러나 건물 내 다수의 출구가 존재하고 인원분포가 불균형적인 상황에서 단순히 최단거리에 있는 출구로 재실자들을 유도하게 되면, 특정 출구에 많은 인원이 집중되어 오히려 총 대피시간을 지연시

Received: 2016.02.23, revised: 2016.03.17, accepted: 2016.03.22

* 정희원 · 서울시립대학교 공간정보공학과 석사과정(Member, Master candidate, Dept. of Geoinformatics, University of Seoul, lmhll123@uos.ac.kr)

** 서울시립대학교 공간정보공학과 박사과정(Ph. D. candidate, Dept. of Geoinformatics, University of Seoul, hwnam@uos.ac.kr)

*** 교신저자 · 정희원 · 서울시립대학교 공간정보공학과 교수(Corresponding Author, Member, Professor, Dept. of Geoinformatics, University of Seoul, cmjun@uos.ac.kr)

킬 수 있다. 따라서 총 대피시간을 최소화하려면, 재난 현장의 인원분포를 고려하여 대피 안내가 이루어져야 한다.

대상 공간의 인원분포를 기반으로 총 대피시간을 최소화하는 연구는 일부 진행된 바 있다. 일반적으로 그래프 이론을 이용하여 실내공간을 모델링하고(Chalmet et al., 1982) 수학적 기법(Hamacher and Tjandra, 2001) 및 network flow(Pursals and Garzón, 2013), 휴리스틱 기반의 최적화 알고리즘(Yusoff et al., 2008) 등을 활용해 최소 대피시간을 탐색하고 있다. 최근에는 모델의 검증을 위해 cellular automata(CA)와 같은 보행모델을 적용한 미시적 시뮬레이션도 활용되었다(Kwak et al., 2012). 하지만 이러한 연구들은 최적화 및 시뮬레이션 등 다소 긴 연산시간이 요구되는 방법들을 이용해 총 대피시간 최소화에 접근하기 때문에 재난 상황에 실시간으로 적용하기에는 한계가 있다.

이에 본 연구는 짧은 연산시간으로 총 대피시간을 단축시킬 수 있는 대피경로 할당 알고리즘을 제안한다. 대피경로 할당 알고리즘은 다수의 출구가 존재하는 건물에서 대규모 인원 대피 시, 각 출구에 적절하게 대피 인원을 할당하여 출구들의 교통 정체 상황을 균형적으로 유지함으로써 총 대피시간을 단축시킨다. 이와 같은 방법은 비록 최적화된 결과는 아닐지라도 그와 유사한 결과를 빠르게 산출할 수 있다는 장점이 있다(Xie et al., 2014). 각 출구에 할당할 대피인원을 결정하는 방법은 그래프 이론을 기반으로 탐욕 알고리즘의 접근 방식을 활용하였다. 본 연구에서는 제한한 알고리즘의 검증을 위해 CA 기반 대피 시뮬레이터를 이용하였고 실제 건물과 유사한 구조에서 다양한 인원분포를 적용하여 실험 결과를 도출하였다.

2. 관련 연구 분석

실내공간에서 대피 문제를 해결하기 위해 개발된 모델들은 Lovas(1995), Gwynne et al.(1999), Kuligowsky et al.(2005) 등의 연구에 의해 다뤄진 바 있다. 본 연구는 총 대피시간의 단축이 목적이기 때문에 이번 장에서는 총 대피시간 최소화와 관련되어 수행된 연구들 위주로 살펴보고자 한다. 앞서 설명한 바와 같이 대부분의 관련 연구가 그래프 기반의 데이터 구조를 이용하기 때문에 본 장에서 언급하는 노드는 보행자들이 머무는 방과 같은 공간을 의미하고 에지는 복도와 같은 이동통로를 의미한다.

총 대피시간 최소화를 목적으로 개발된 대표적인 대피 모델로는 network flow 방식을 활용한 EVACNET+

가 있다(Kisko and Francis, 1985). 이 모델은 총 대피시간의 최대 범위를 예측한 뒤, 전체 노드(source)와 출구(sink) 사이의 가능한 모든 경로에 대하여 시간 전개형 네트워크를 생성한다. 이 후, 시간 전개형 네트워크 상에서 각 노드별 최소 비용 경로를 탐색하여 대피경로를 산출하고 산출된 대피경로의 총 대피시간이 최대 범위보다 작을 경우 최적해로 판단한다. 이와 같은 방법은 대상 공간 전체의 인원분포를 고려한 최적해 산출이 가능하다는 장점이 있다. 하지만 네트워크 사이즈가 커질수록 총 대피시간의 최대 범위를 예측하는 일이 매우 어려워지고 시간 전개형 네트워크 생성에 오랜 연산시간이 소요되기 때문에 복잡한 건물에서 발생한 재난 상황에 실시간으로 적용하기는 어렵다.

Lu et al.(2005)는 시간 전개형 네트워크를 이용한 최적 대피경로 탐색 방법의 높은 연산복잡도 문제를 지적하며 휴리스틱 기반의 Capacity Constrained Route Planner를 제안하였다. 이 알고리즘은 네트워크 상의 노드와 에지의 한계 용량, 에지에서의 이동시간, 전체 보행자 수 및 보행자들의 초기 위치, 대피 목적지 등이 주어지면 최적에 근사한 대피경로를 산출한다. 네트워크 사이즈가 큰 상황에서도 비교적 짧은 연산시간이 소요된다. 하지만 에지의 한계용량이 보행 흐름에 기반하여 설정된 것이 아니고 에지에서의 이동이 반드시 First In First Out 규칙에 따라야 한다는 제한사항이 존재한다.

이 외에 simulated annealing(Kim et al., 2008) 및 유전자 알고리즘(Kwak et al., 2012; Li et al., 2010)과 같은 휴리스틱 기법들도 대피 문제에 적용되었다. Kwak et al.(2012)는 임의의 인원분포 상황에서 최적의 대피결과를 도출하기 위해 CA 기반 보행모델에 유전자 알고리즘을 적용한 연구를 수행하였다. 총 대피시간 최소화 뿐만 아니라 미시적 관점에서 대피의 전개상황을 파악할 수 있었다. 그러나 시뮬레이션으로 인해 다소 긴 연산시간이 소요된다는 한계가 여전히 존재하였다.

Wang and Liu(2012)는 재난 상황에서 실시간 대피 안내 뿐만 아니라 총 대피시간 최소화를 도출하는 방법으로 개선된 Fast Flow Control Algorithm(Chen and Feng, 2009)을 제안하였다. 이 알고리즘은 개별 보행자 단위로 산술적으로 최소 대피시간이 소요되는 출구를 탐색하여 총 대피시간을 최소화하는 방식이다. 하지만 개별 보행자 단위로 대피경로를 산출하게 되면 보행자들 간의 마찰로 인해 오히려 혼란이 가중될 확률이 높다. 또한 현실적으로 동일 공간에 있는 보행자들에게 서로 다른 대피경로를 안내한다는 것은 매우 어려운 일이다.

Xie et al.(2014)는 다수의 출구가 존재할 때, 각 출구의 교통 정체 상황을 균형적으로 유지함으로써 총 대피시간을 최소화하는 전략을 제안하였다. 이 연구에서는 각 노드에 최단거리에 위치한 출구를 할당한 뒤, 대피종료시간이 가장 오래 걸리는 출구를 할당받은 노드 중 해당 출구로부터 멀리 떨어진 노드부터 대피종료시간이 가장 적게 걸린 출구로 재할당하는 과정을 반복하면서 총 대피시간의 최소값을 도출하였다. 또한 노드별로 대피 시작 시간에 차등을 주어 출구에서의 교통 혼잡을 최소화하였다. 그러나 대피 시작 시간을 노드별로 다르게 설정한다는 전제는 도시를 대상으로 한 대규모 대피계획에는 적용될 수 있으나 건물 내 화재나 붕괴와 같은 긴급한 상황에 적용하기에는 부적합하다. 그리고 특정 노드에 대하여 할당받은 출구를 변경하면서 산술적으로 소요되는 총 대피시간을 줄이는 방식은 인접 노드가 할당받은 출구를 고려하지 않기 때문에 대피경로의 엇갈림으로 인한 혼란을 야기하기 쉽다.

본 연구는 Xie et al.(2014)의 접근방식과 유사하다. 서론에서 언급한 바와 같이 다수의 출구가 존재하는 상황에서 각 출구에 적절하게 대피인원을 할당하는 방식으로 총 대피시간을 단축시키기 때문이다. 알고리즘에 대한 구체적인 설명은 3장에서 진행한다.

3. 대피경로 할당 알고리즘

3.1 대피경로 할당 알고리즘

Fig. 1은 다수의 출구가 존재하는 건물에서 대규모 인원이 불균형적으로 분포된 상황을 나타낸 것이다. 만약 이러한 상황에서 최단거리에 있는 출구로 대피 안내가 이루어진다면 Exit C에 너무 많은 인원이 집중되게 된다. 따라서 비교적 인원이 적은 R3과 R9를 Exit A로 유도하고 R5는 Exit B, R11은 Exit D로 유도하여 Exit C에는 R6와 R12의 재실자들만 대피할 수 있도록 하는 대피 전략이 필요하다

대피경로 할당 알고리즘은 이와 같은 전략을 그래프 이론을 통해 구현한다. 즉, Fig. 2와 같이 각 출구에 할

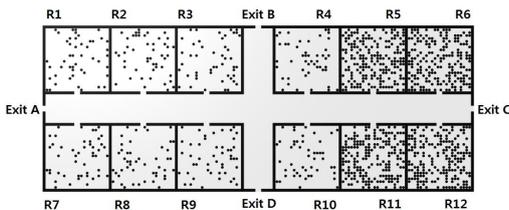


Figure 1. Building structure and distribution of occupants

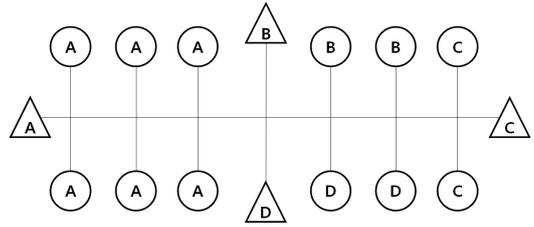


Figure 2. Evacuation route assignment using a graph network

당할 노드(방)를 결정하는 방법을 제시하는 것이다. 특정 출구에 할당할 노드를 결정하는 방식은 탐욕 알고리즘을 기반으로 이루어진다. 탐욕 알고리즘은 Fig. 3과 같이 우선 후보군들 중 최적이라고 판단되는 부분 해를 선택하고 선택한 부분 해를 적용할 수 있는지 검사한 뒤, 적용 가능하다면 해 집합에 추가한다. 그리고 이 과정을 수렴조건에 도달할 때까지 반복하여 최종 해 집합을 산출한다(Cormen et al., 1990). 본 연구에서는 Fig. 3에 표시한 활용 부분, 즉, 후보군 중 부분 해 선택, 적용 가능성 검사, 해 집합에 추가 과정을 각 출구에 적용하여 대피경로 할당 알고리즘을 완성하였다.

Fig. 4는 대피경로 할당 알고리즘의 수도 코드이다. 입력 데이터로는 건물 구조를 나타내는 그래프와 각 방과 출구들의 위상학적 인접성 네트워크가 있다. 건물 구조 그래프는 대피경로를 산출하는데 사용된다. 본 연구에서는 다익스트라 알고리즘을 이용하여 서로 다른 노드 간의 최단거리를 계산한다. 각 방과 출구는 벽을 공유하거나 근거리에서 서로 마주보는 혹은 대각에 위

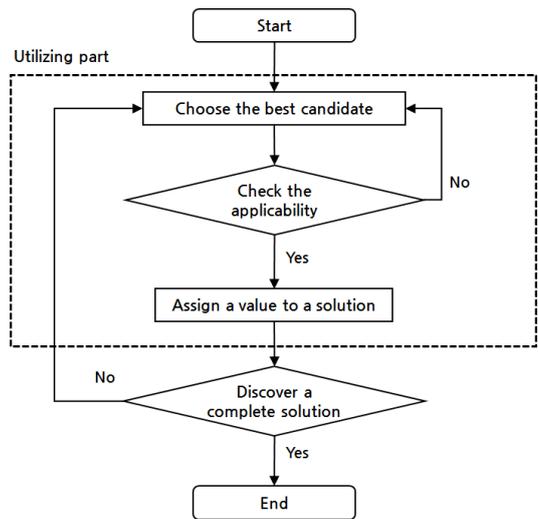


Figure 3. Greedy algorithm process

```

Graph of Building structure, Topological(mesh) Network
O=N    (n: # of rooms)           // Open node group is composed of all room node
E=M    (m: # of exits)          // Exit node group is composed of all exit node
C(e)={e}   for  $\forall e: e \in E$       // Assign node group per exits. Initial value is allocated by own
w(e)=0    for  $\forall e: e \in E$       // A wait persons per exits. Initial value is 0

while(O  $\neq \emptyset$ )              // Repeat while open node group is to be empty set
{
  Let e  $\in E$  be an exit for which  $w(e) = \min\{w(e): e \in E\}$  // Select exit have minimum waiting people
  if(n(e)>1)                      // Select random exit when minimum wait people exit is multiple
    Let e  $\in E$  be an exit randomly selected for which  $w(e) = \min\{w(e): e \in E\}$ 

  // Node group is adjacent to current exit
  Let A(i)  $\subset O - C(e)$  be a set of nodes for which  $i \in C(e)$ 

  if(A(i)  $\neq \emptyset$ )
  {
    Let j be a node for which  $d(j) = \min\{d(j): j \in A(i)\}$  // Select nearest node from current exit
    C(e) = C(e)  $\cup \{j\}$  // Add assign node group
    O = O - {j} // Except from open node group
  }
  else
  {
    // If all adjacent nodes are assigned to other exits, Ignore assign status make adjacent node
    Let A(i)  $\subset N - C(e)$  be a set of nodes for which  $i \in C(e)$ 
    AssignCondition = false

    while(A(i)  $\neq \emptyset \wedge AssignCondition \equiv false$ ) // A(i) can be reduced to by following rule
    {
      Let j be a node for which  $d(j) = \min\{d(j): j \in A(i)\}$ 
      AssignCondition = AssignRule(j, e') for  $\exists e': e' \in E - \{e\}$  //Check assign rule
      if(AssignCondition  $\equiv true$ )
      {
        C(e) = C(e)  $\cup \{j\}$ 
        C(e') = C(e') - {j} // Delete from assigned to exit group was before
      }
      else { A(i) = A(i) - {j} } // If un follow the rule, except from adjacent node group
    }
    if(A(i)  $\equiv \emptyset$ )
      w(e) =  $\infty$  // If node to assign is none, Current exit's wait people set as  $\infty$ 
  }
}

```

Figure 4. Pseudo code of evacuation route assignment algorithm

치하는 관계에 따라 인접성을 가지게 된다. Fig. 5는 Fig. 1의 구조에서 생성한 위상학적 인접성 네트워크에 해당한다.

초기화하는 변수로는 오픈노드 집합, 출구 집합, 출구별 할당노드 집합, 출구별 대기인원이 있다. 오픈노드 집합은 어떠한 출구로도 할당되지 않은 노드들의 집합을 의미하며 초깃값은 전체 방 노드들로 이루어진다. 출구 집합은 실외로 대피가 가능한 건물 내 전체 출구들을 의미하며 각 출구는 할당노드 집합과 대기인원을

속성으로 가지게 된다. 할당노드 집합은 출구에 할당된 노드들, 즉 해 집합을 의미하며 모든 출구는 할당노드 집합의 초깃값으로 자기 자신을 가진다. 대기인원은 출구가 할당받은 대피인원들 중 아직 대피를 완료하지 못한 인원들을 의미한다. 초기에는 어떠한 방 노드도 출구가 결정되지 않았으므로 모든 출구의 대기인원은 0이다.

대피경로 할당 알고리즘은 모든 방 노드의 출구 할당이 완료될 때까지 지속된다. 즉, 수렴조건은 오픈노드

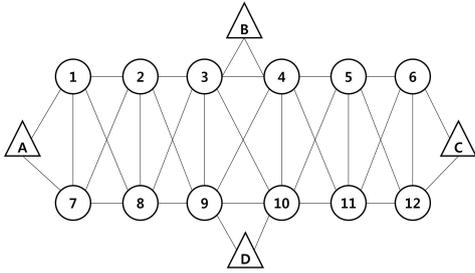


Figure 5. Example of topological network

집합이 공집합이 되는 것이다. 오픈노드 집합이 공집합이 아니라면 대기인원이 최소인 출구가 선택된다. 만약 최소 대기인원을 가진 출구가 복수라면 그 중 임의의 출구가 선택되게 된다.

3.2 노드 할당

출구가 정해지면 부분 해 선택, 즉, 후보군 중 할당노드 집합에 포함시킬 노드를 선택하게 된다. 후보군은 할당 가능한 노드들의 집합으로 오픈노드들 중 현 출구의 할당노드들과 인접한 노드들로 구성된다.

예를 들어 위상학적 인접성 네트워크가 Fig. 5와 같고, 어떠한 방 노드도 출구가 결정되지 상태에서 Exit A가 선택되었다면($O=N$, $e=Exit A$) Exit A의 할당노드 집합에는 자기 자신밖에 없다($C(e)=\{A\}$). 따라서 후보군은 A와 인접한 오픈노드들로 구성되어 $\{N1, N7\}$ 이 된다($A(i)=\{N1, N7\}$). 만약 Exit A의 할당노드 집합이 $\{A, N1\}$ 이라면 후보군은 $\{N7, N2, N8\}$ 이 되게 된다. 오픈노드들로 후보군이 구성될 경우, 현 출구와 가장 근접한 노드를 부분 해로 선택하게 되고 최근접 노드가 복수라면 재실인원이 더 많은 노드를 부분 해로 선택한다. 그리고 선택한 노드를 적용가능성 검사 없이 할당노드 집합에 추가하게 된다.

그러나 노드 할당 과정을 여러 출구가 반복하다 보면 특정 출구는 할당노드 집합의 인접 노드들이 모두 다른 출구로 할당이 완료된 상황을 직면하게 된다. 이와 같은 상황이 발생한다면, 이미 다른 출구로 할당이 완료

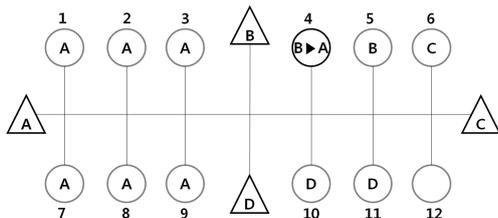


Figure 6. Example of unreasonable evacuation route assignment I

된 노드라 할지라도 인접 조건을 만족할 시, 후보군에 포함시킨다. 할당노드 집합에 포함시킬 노드를 선택하는 과정에는 할당규칙의 준수 여부를 검토하게 된다. 각 출구가 단순히 인접해있는 노드들을 흡수하는 방식으로만 진행된다보면 비합리적인 결과가 나올 수 있기 때문이다. 할당규칙에는 최대 할당 영역, 할당 연결 규칙, 교착상태 판단 등이 있다. 현 출구와 근접한 노드 순으로 할당규칙을 검토하고 모든 할당규칙을 만족하는 노드가 있다면 할당노드 집합에 추가한다.

3.2.1 최대 할당 영역

Fig. 6는 Exit A에 Exit B에 할당되어 있던 N4를 재할당한 상황이다. 이 상태에서 각 노드가 할당받은 출구로 대피를 진행한다면, N4의 재실자들은 최단거리에서 Exit B가 있음에도 불구하고 Exit A로 대피해야 한다. 이처럼 특정 출구에 과도하게 노드들을 할당하여 발생하는 비합리적인 결과를 방지하기 위해 최대 할당 영역 규칙을 고안하였다.

최대 할당 영역은 각 출구가 가질 수 있는 할당노드 집합의 최대 범위를 의미한다. 각 출구는 노드 할당 과정을 반복하다가 타 출구의 인접 노드를 만날 수 있다. Fig. 6에서 N4가 할당되기 전 Exit A의 할당노드 집합은 $\{N1, N7, N2, N8, N3, N9\}$ 이다. 따라서 후보군은 $\{N4, N10\}$ 이며 각각 Exit B와 Exit D의 인접 노드이다. 여기서 Exit A에 어떠한 노드를 할당하더라도 B와 D 중 하나는 자신의 인접 노드를 모두 잃게 된다.

이처럼 어떠한 노드를 할당하더라도 타 출구가 자신의 인접 노드를 모두 잃어버리는 상황이 발생한다면, 이 상태를 최대 할당 영역을 만족한 것으로 판단한다. 그리고 최대 할당 영역을 만족한다면 해당 출구는 더 이상의 노드 할당 과정을 수행하지 않도록 대기인원 값을 ∞ 으로 갱신한다. 따라서 Exit A의 최대 할당 영역은 Fig. 7과 같다.

3.2.2 할당 연결 규칙

Fig. 8은 Exit A에 Exit B에 할당되어 있던 N3을 재

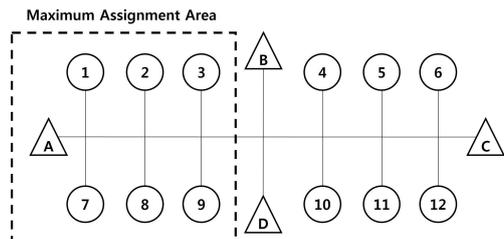


Figure 7. Maximum assignment area

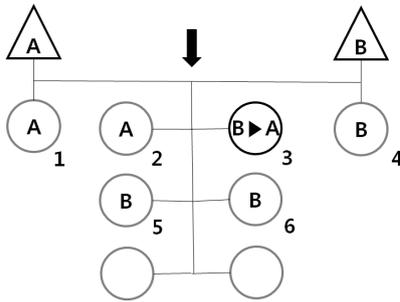


Figure 8. Example of unreasonable evacuation route assignment II

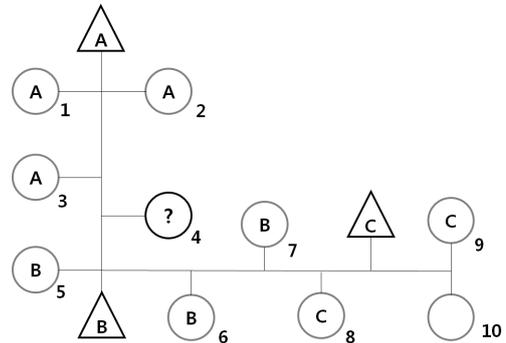


Figure 10. Example of deadlock

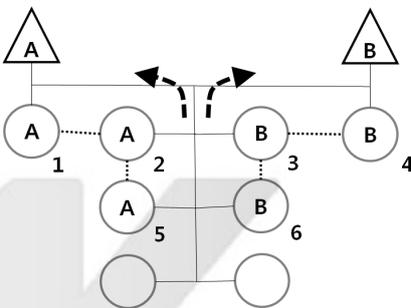


Figure 9. Assign connection rule

할당된 상황이다. 이 상태에서 각 노드가 할당받은 출구로 대피를 진행한다면, 화살표로 표시한 지역에서 N2, N3의 재실자들은 Exit A로 향하려 하고 N5, N6의 재실자들은 Exit B로 향하려 하기 때문에 대피인원들 간의 마찰로 인해 대피의 혼란이 발생할 수 있다.

할당 연결 규칙은 동일한 출구를 할당받은 노드끼리는 인접관계를 통해 모두 연결되어야 함을 의미한다. 현 출구는 인접노드들로 이루어진 후보군 중에서 해를 선택하기 때문에 연결성이 유지된다. 하지만 타 출구에 할당된 노드를 재할당하는 과정에서 타 출구는 연결 관계가 무너질 수 있다. Fig. 8에서 Exit A에 N3을 할당하면 Exit B의 할당노드 집합은 {N4, N6, N5}가 된다. N6과 N5는 인접관계를 가지지만 N4와 나머지 노드들은 인접관계가 아니다. 즉, Exit A에 N3을 할당하게 되면 B의 할당 연결 규칙이 무너지는 것이다. 따라서 Fig. 9과 같이 Exit A에는 N5를 할당하여 A와 B의 연결 규칙을 유지하고 대피인원들 간의 마찰을 최소화해야 한다.

3.2.3 교착상태 발생 및 대립해결기법

Fig. 10에서 N4가 Exit B에 할당되어 있고 Exit A의 대기인원이 최소라면 Exit A는 N4를 할당노드 집합에

추가하게 된다. N4가 Exit A에 재할당되고 Exit B의 대기인원이 최소가 되면 Exit B는 다시 N4를 재할당 받는다. 즉, N4가 어디에 할당되느냐에 따라 A와 B중 대기인원이 최소인 출구가 결정나게 되고 대기인원이 최소인 출구는 다시 N4를 할당받는 상황이다. 따라서 복수의 출구가 동일한 노드에 대하여 할당과 재할당을 반복하면서 연산시간이 무한으로 길어지게 되는데 이를 교착상태라고 부른다. 본 연구에서는 교착상태를 해결하기 위해 다음과 같은 대립해결기법을 적용하였다.

N4가 Exit B에 할당되어 있다고 가정하면 Exit A가 가질 수 있는 후보군은 {N4, N5}이며 후보 노드들이 할당받은 출구는 모두 B이다. N4가 Exit A에 할당되어 있다면 Exit B가 가질 수 있는 후보군은 {N4, N3, N8}이며 후보 노드들이 할당받은 출구는 각각 A, A, C이다. Exit A의 경우, 후보 노드들이 모두 B에 할당되어 있기 때문에 어떠한 노드를 흡수하더라도 B와 대립이 발생한다. 하지만 Exit B의 경우는 C의 노드를 흡수할 수도 있다. 따라서 Exit B가 A에 할당된 N4와 N3 대신 N8을 흡수한다면 Exit A와 Exit B의 대립은 발생하지 않는다. 즉, 대립하고 있는 두 출구의 후보 노드들이 할당된 출구를 조사하여, 후보 노드들이 할당된 출구가

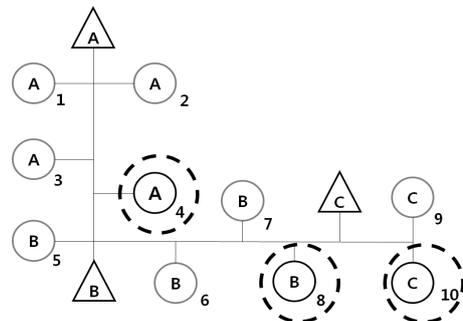


Figure 11. Example of conflict resolution

복수개로 나타나는 출구가 후보 노드들이 할당된 출구가 하나뿐인 출구에 해당 노드를 양보함으로써 대립을 해결할 수 있는 것이다. Fig. 11은 동일한 출구에 할당된 노드만으로 후보군이 구성된 Exit A에 N4를 할당하고 Exit B가 Exit C의 노드를 흡수한 뒤, Exit C에 추가적인 노드할당 과정을 진행하여 대립을 해결한 예시이다.

3.3 대기인원 계산

현 출구에 노드 할당 과정을 마무리하면 수렴조건을 검토한다. 만약 오픈노드 집합이 공집합이 아니라면 모든 출구의 대기인원을 계산하여 대기인원이 최소인 출구에 추가적인 노드 할당 과정을 실시한다.

앞서 언급하였듯이 대기인원은 현재 계산시점을 기준으로 전체 대피인원 중 아직 대피를 완료하지 못한 인원들을 의미한다. 따라서 현재 계산시점을 t_c 라 한다면, 대기인원(E_i)은 출구에 할당된 전체 대피인원(x_{ij})에서 대피가 일어난 시간동안 출구에서의 보행흐름(f_i)을 제외하여 산출할 수 있다. 수식으로 나타내면 Eq. (1)과 같다.(Table 1 참조)

$$E_i = \sum_{j=1}^{m_i} x_{ij} - \int_{t_{i1}}^{t_c} f_i dt \quad (1)$$

f_i 는 단위시간당 대피할 수 있는 인원수를 의미한다 (Fruin, 1992; Pursals and Garzón, 2009). 본 연구에서는 f_i 를 출구의 너비에 비례한 상수로 설정하였다 (Chen and Feng, 2009; Wang and Liu, 2012). 예를 들어 통과하는데 1초가 걸리는 출구의 너비가 3m이고 보행자 한명이 점유하는 폭이 1m라면 f_i 는 3명/s가 된다.

Table 1. Nomenclature

E_i	Number of pedestrians waiting for evacuation through exit i
x_{ij}	Number of pedestrians in node j allocated in exit i
m_i	Number of nodes allocated in exit i
f_i	Pedestrian flow at exit i
t_{i1}	Evacuation start time at exit i
t_{max}	Maximum of arrival time from node to assigned exit
l_{ij}	Shortest path from node j to exit i
l_{max}	Maximum of shortest path from node to assigned exit
v	Average walking speed of pedestrians

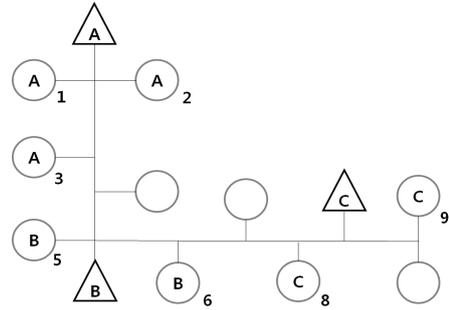


Figure 12. Evacuation route assignment status

Table 2. Shortest path from node to exit

N	A		B		C	
	l_{ij}	N	l_{ij}	N	l_{ij}	
1	10	5	10	8	10	
2	10	6	15	9	15	
3	20					

현재 계산시점은 지금까지 할당이 완료된 모든 노드의 재실인원들이 해당 출구까지 도달하는데 걸리는 시간을 의미한다. 따라서 이는 현재 할당이 완료된 노드들 중 해당 출구로부터 가장 멀리 떨어진 노드의 재실인원들이 출구에 도달하는 시간과도 같다. 즉, $t_c = t_{max}$ 이고 $t_{max} = l_{max}/v$ 가 성립된다. 여기서 최단 거리 l 은 노드의 출입구부터 출구까지의 거리에 해당한다. 각 출구별로 대피가 시작되는 시점은 각 출구의 가장 근접한 노드가 해당 출구에 도달하는 시간과 같다. 예를 들어, 현재 할당 상태가 Fig. 12과 같고 할당된 노드들의 각 출구까지의 최단거리가 Table 2와 같다면, 현재 계산시점 t_c , 즉, t_{max} 는 N3이 Exit A에 도달하는데 걸리는 시간이다. 그리고 각 출구의 대피 시작 시점은 $10/v$ 로 모두 동일하다.

결과적으로 특정 출구의 최근접 노드를 k 라 했을 때 최종 대기인원 산출 식은 Eq. (2)와 같다.

$$E_i = \sum_{j=1}^{m_i} x_{ij} - f_i(t_{max} - t_{i1}) = \sum_{j=1}^{m_i} x_{ij} - f_i\left(\frac{l_{max} - l_{ik}}{v}\right) \quad (2)$$

4. 적용 테스트

4.1 CA 기반 대피 시뮬레이터

대피경로 할당 알고리즘의 최종적인 흐름은 Fig. 13

과 같은 알고리즘의 검증을 위해 CA 기반 대피 시뮬레이터를 이용하였다. CA 기반 대피 시뮬레이터는 개별 보행자의 이동을 묘사하기 때문에 알고리즘에 의한 대피 결과를 보다 미시적으로 확인할 수 있는 장점이 있다. 시뮬레이션의 실행은 각 노드의 재실인원과 CA 보행모델에 대한 파라미터를 입력한 후 진행된다. 본 연구에서는 총 대피시간 단축을 위한 알고리즘 개발에 중점을 두었기 때문에 CA 보행모델(Nam et al., 2014)에 대한 자세한 설명은 생략하였다.

4.2 실험결과 및 분석

알고리즘에 대한 실험은 크게 두 부분으로 나누어서 진행하였다. 첫 번째 부분에서는 알고리즘에 의한 대피 결과를 검증하기 위해 최단거리에 있는 출구로 재실인원을 대피시켰을 경우와 비교, 분석해 보았다. 유사 건물 구조 및 실제 건물 구조에서 실험을 진행하였고 Table 3 and 4가 그 결과에 해당한다. 두 번째 부분에서는 알고리즘의 연산시간을 분석하였다. Intel X5650 2.67GHz CPU, 4.00GB RAM의 PC 사양에서 실험을 진행하였으며 재난현장에 적용가능한지 검토하기 위해

Wang and Liu(2012)에서 제안된 실시간 대피 알고리즘과 연산시간을 비교해보았다. 대피경로 할당 알고리즘은 편의상 ERAA(Evacuation Route Assignment Algorithm)로 표기하였다.

4.2.1 대피결과 분석

Table 3은 Fig. 1과 같은 공간 구조에서 1, 2, 7, 8번 방, 즉 출구 A쪽에 많은 재실인원을 배치했을 때의 결과이다. 대피시간은 시뮬레이터의 단위시간인 tick을 이용하였고 대피인원 계산을 위한 각 출구에서의 보행 흐름은 2명/tick으로 모두 동일하게 설정하였다.

대피경로를 비교해보면, 2번방과 8번방이 각각 출구 B, D에 할당되어 재실인원들이 우회하였고 4, 10번방은 출구 C에 할당되어 재실인원들이 출구 C쪽으로 대피하였다. 즉, 기존의 A에 걸리던 부하를 방지하였고 B와 D에 걸릴 수 있는 부하도 방지하였다. 이로 인해 총 대피시간이 100tick 이상 빨라진 것을 확인할 수 있다.

각 출구의 누적 대피인원 그래프를 살펴보면 알고리즘을 적용한 후의 변화를 극명하게 확인할 수 있다. 누적 대피인원 그래프의 x축은 대피시간이며 y축은 누적

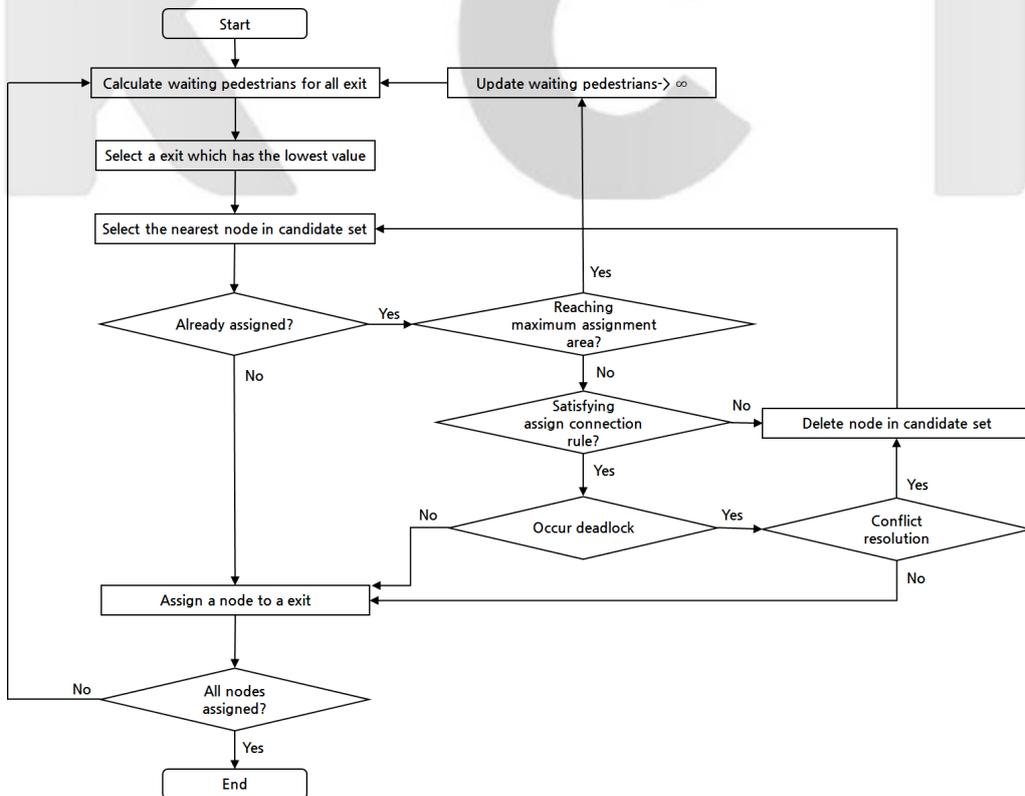


Figure 13. Whole process of evacuation route assignment algorithm

Table 3. Test results I

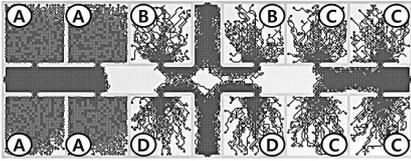
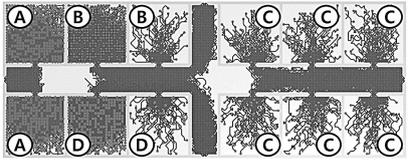
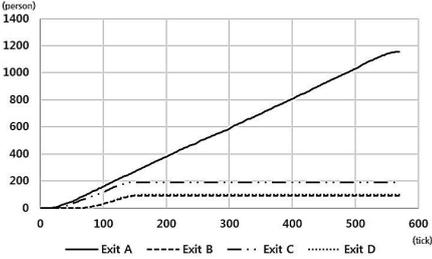
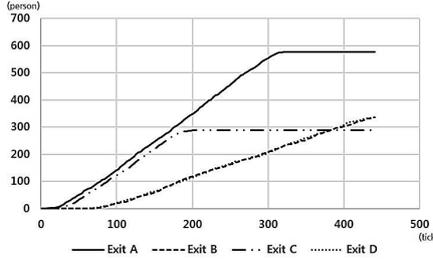
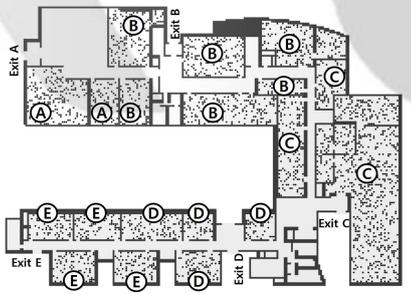
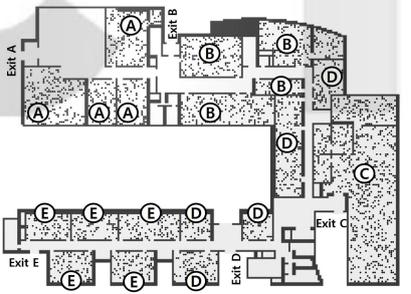
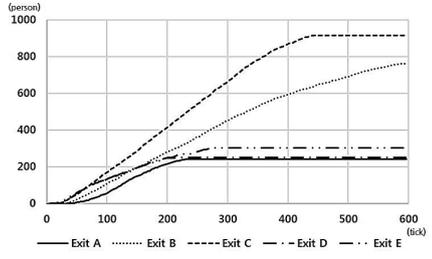
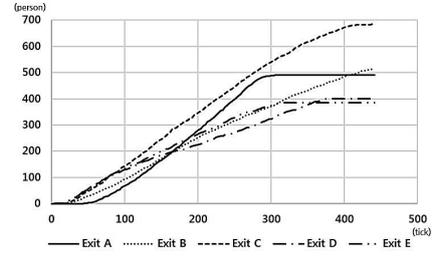
	The nearest exit guide	ERAA																														
Evacuation route																																
Cumulative number of evacuee																																
Compare each exit	<table border="1"> <thead> <tr> <th>Exit</th> <th>Evacuation time</th> <th>Evacuee</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>570</td> <td>1156</td> </tr> <tr> <td>B</td> <td>159</td> <td>93</td> </tr> <tr> <td>C</td> <td>165</td> <td>192</td> </tr> <tr> <td>D</td> <td>159</td> <td>99</td> </tr> </tbody> </table>	Exit	Evacuation time	Evacuee	A	570	1156	B	159	93	C	165	192	D	159	99	<table border="1"> <thead> <tr> <th>Exit</th> <th>Evacuation time</th> <th>Evacuee</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>324</td> <td>578</td> </tr> <tr> <td>B</td> <td>441</td> <td>337</td> </tr> <tr> <td>C</td> <td>201</td> <td>288</td> </tr> <tr> <td>D</td> <td>441</td> <td>337</td> </tr> </tbody> </table>	Exit	Evacuation time	Evacuee	A	324	578	B	441	337	C	201	288	D	441	337
Exit	Evacuation time	Evacuee																														
A	570	1156																														
B	159	93																														
C	165	192																														
D	159	99																														
Exit	Evacuation time	Evacuee																														
A	324	578																														
B	441	337																														
C	201	288																														
D	441	337																														

Table 4. Test results II

	The nearest exit guide	ERAA																																				
Evacuation route																																						
Cumulative number of occupants																																						
Compare each exit	<table border="1"> <thead> <tr> <th>Exit</th> <th>Evacuation time</th> <th>Evacuee</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>237</td> <td>242</td> </tr> <tr> <td>B</td> <td>597</td> <td>763</td> </tr> <tr> <td>C</td> <td>450</td> <td>916</td> </tr> <tr> <td>D</td> <td>219</td> <td>250</td> </tr> <tr> <td>E</td> <td>270</td> <td>303</td> </tr> </tbody> </table>	Exit	Evacuation time	Evacuee	A	237	242	B	597	763	C	450	916	D	219	250	E	270	303	<table border="1"> <thead> <tr> <th>Exit</th> <th>Evacuation time</th> <th>Evacuee</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>306</td> <td>490</td> </tr> <tr> <td>B</td> <td>441</td> <td>515</td> </tr> <tr> <td>C</td> <td>438</td> <td>684</td> </tr> <tr> <td>D</td> <td>375</td> <td>401</td> </tr> <tr> <td>E</td> <td>318</td> <td>384</td> </tr> </tbody> </table>	Exit	Evacuation time	Evacuee	A	306	490	B	441	515	C	438	684	D	375	401	E	318	384
Exit	Evacuation time	Evacuee																																				
A	237	242																																				
B	597	763																																				
C	450	916																																				
D	219	250																																				
E	270	303																																				
Exit	Evacuation time	Evacuee																																				
A	306	490																																				
B	441	515																																				
C	438	684																																				
D	375	401																																				
E	318	384																																				

대피인원이다. 최단거리 출구로 안내할 경우, 다른 출구들에 비해 A의 누적인원 증가가 압도적인 것을 볼 수 있다. 이는 다른 출구들은 이미 대피가 종료되었음에도 불구하고, A에서는 한참동안 대피가 지속된다는 것을 의미한다. 수치상으로는 대피시간은 약 2.5배, 대피인원은 6배에서 10배까지 차이가 난다.

하지만 알고리즘을 적용하고 나면 출구 A에 걸리는 부하가 상당히 해소되는 것을 확인할 수 있다. 그래프의 기울기가 0이 되는 시점, 즉 대피가 종료되는 시점은 오히려 B와 D보다 빨라진다. 이는 각 출구에 대피인원이 비교적 균형적으로 할당되었기 때문인 것으로 유추해볼 수 있다.

Table 4는 실제 대학 캠퍼스 건물 단층에 2304명을 복도를 제외한 각 방에 균일한 밀도로 배치한 후 실험한 결과이다. 각 출구에서의 보행흐름은 실제 출구의 폭을 고려하여 A부터 E순으로 4, 3, 5, 3, 3(명/tick)으로 설정하였다.

최단거리 출구로 안내할 경우, 출구 C 주변에 재실인원 용량이 큰 방이 존재하기 때문에 C에 많은 인원이 할당되게 된다. 출구 B 역시 인접한 방들이 많아 대피인원이 타 출구에 비해 많게 된다. 누적 대피인원 그래프를 보면 쉽게 확인할 수 있다. 특이사항으로는 C가 B

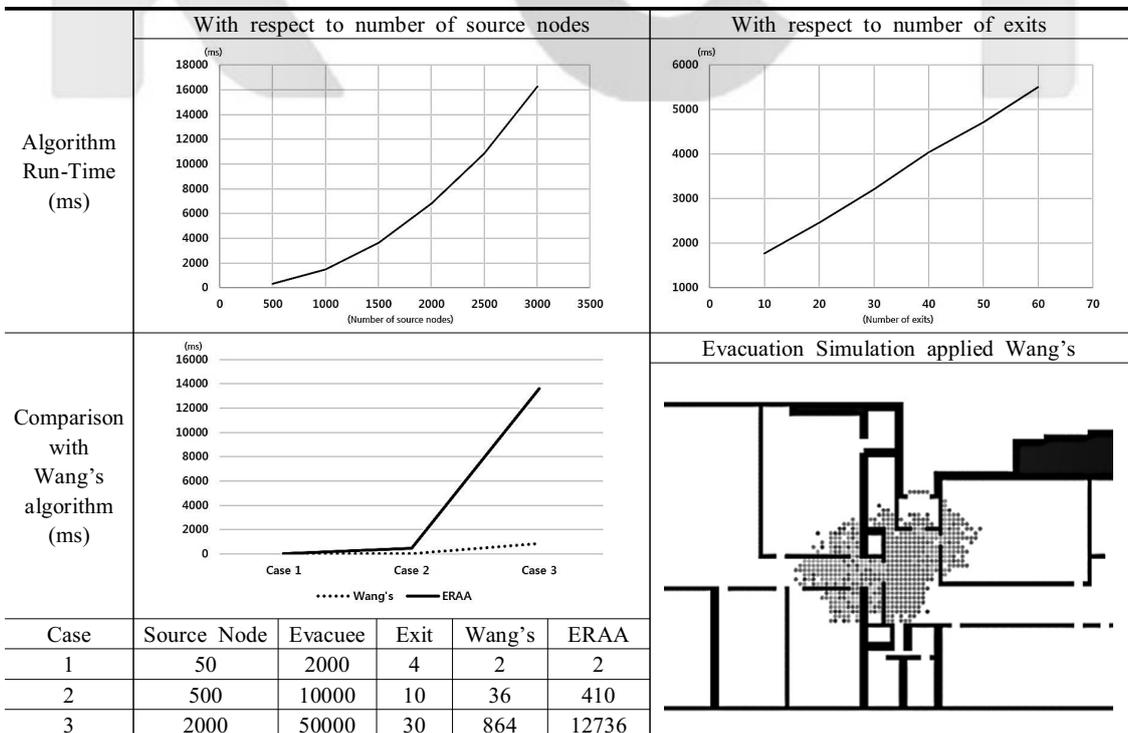
에 비해 할당된 대피인원이 많음에도 불구하고 대피종료시점이 더 빠르다는 점이 있다. 이는 C의 보행흐름이 B보다 커서 단위시간당 대피할 수 있는 인원이 많기 때문에 가능하다. 나머지 출구들은 할당된 대피인원이 적어 대피종료시점이 B와 C에 비해 월등히 빠르게 나타난다.

알고리즘을 적용한 후에는 각 출구의 대피시간 차이가 많이 줄어든 것을 확인할 수 있다. B와 C의 경우, 거의 동시에 대피가 종료되었으며 D의 대피시간이 많이 증가하였다. B의 대피시간이 감소된 이유는 A에 할당량을 분배했기 때문이며 C는 재실인원이 많은 하나의 방을 제외하고 나머지 방들을 D에 양보하여 대피시간을 단축시켰다.

누적 대피인원 그래프를 살펴보면 A는 대피가 시작된 시점부터 C와 유사한 기울기로 빠르게 대피인원이 누적되는 것을 확인할 수 있다. 이는 A의 보행흐름이 C를 제외한 나머지 출구보다 커서 동일한 경과시간이라도 더 많은 대피인원이 누적되기 때문이다. B, D, E의 경우, 보행흐름이 동일하여 유사한 기울기로 대피인원이 증가하고 있다.

최종적으로 각 출구의 할당량이 균형적으로 유지되어 총 대피시간은 150tick 감소된 결과를 얻었다. 또한

Table 5. Test results III



단 하나의 방만 할당받은 C의 대피종료시점이 총 대피 시간이 되었으므로 하나의 방의 재실인원들은 모두 동일한 경로로 대피한다는 가정 하에서는 이 결과가 최적 해라고 할 수 있다.

4.2.2 연산시간 분석

대피경로 할당 알고리즘은 대기인원이 최소인 출구가 후보군 중 최적이라고 판단되는 노드를 흡수하는 방식으로 전체 노드의 할당이 완료될 때까지 진행되기 때문에 전체 재실인원 수는 알고리즘 연산에 부담을 주지 않는다. 따라서 노드(source node) 수와 출구 수를 증가시켜가며 연산시간을 살펴보았다. Table 5가 그 결과에 해당한다.

노드 당 재실인원을 100명으로 하고 출구를 4개로 제한한 상태에서 노드 수를 증가시켜가며 연산시간을 살펴본 결과, 노드 500개의 연산에는 1초가 걸리지 않았고 3000개의 노드에서는 약 16초의 연산시간이 소요되었다. 재실인원은 그대로 적용하고 노드 수를 1000개로 제한한 상태에서 출구 수만 늘렸을 경우에는 출구 수를 10개씩 증가시킬 때마다 약 0.7초의 연산시간이 증가하는 것을 확인하였다.

기존의 짧은 연산시간으로 총 대피시간을 최소화하는 Wang's 알고리즘과 비교해 보았을 때는, Case 1과 같이 일반적인 경우에는 연산시간이 유사하게 나타났으나 극단적인 상황을 가정할수록 차이가 발생하였다. 하지만 Table 4의 실험과 동일한 조건에서 Wang's 알고리즘을 적용하여 대피시뮬레이션을 수행한 결과, Table 5의 이미지와 같이 좁은 통로에서 서로 다른 출구로 대피하려는 인원들이 밀집되어 병목현상이 심화되고 대피가 종료되지 않았다. Wang's 알고리즘은 산술적인 계산으로는 대피시간을 단축시키더라도 동일 공간의 인원들을 서로 다른 출구로 대피시키기 때문에 실제 재난상황에 적용할 경우, 대피의 혼란이 발생할 가능성이 크다. 따라서 Case 2와 3의 연산시간이 총 대피시간의 단축을 보장하기에는 한계가 있다.

5. 결 론

본 연구는 짧은 연산시간으로 총 대피시간을 최소화하기 위해 대피경로 할당 알고리즘을 개발하였다. 일반적으로 다수의 출구가 존재하는 상황에서, 총 대피시간을 최소화하는 최적 대피경로를 산출하기 위해서는 많은 연산량을 필요로 한다. 하지만 본 연구에서는 그리드 셀이 아닌 그래프 구조를 이용하였으며, 목적함수를 대피시간으로 설정하지 않고 각 출구의 할당량 균형적

으로 유지하는 방법으로 대피문제에 접근하여 보다 적은 연산량이 소요된다.

출구에 할당량을 정하는 방식은 탐욕 알고리즘 기반의 대피경로 할당 알고리즘을 통해 이루어졌다. 그래프 이론을 이용하여 실내 공간을 노드와 에지로 구조화한 뒤, 대기인원이 최소인 출구에 최적이라고 판단되는 노드를 할당하는 방식으로, 전체 노드의 할당이 완료될 때까지 진행하였다.

적용 테스트 결과, 최단거리 출구로 실내 인원들을 안내할 경우보다 총 대피시간이 많이 감소된 결과를 획득할 수 있었으며 각 출구에서의 대피시간도 균형적으로 나타났다. 또한 네트워크 사이즈가 작은 경우, 실시간으로 결과 획득이 가능하였으며 극단적인 상황에서도 비교적 짧은 연산시간이 소요되었다.

따라서 센서 네트워크가 구축되어 실내 공간의 재실인원 분포를 감지할 수 있는 건물이라면, 재난 발생 시, 본 연구에서 제안하는 알고리즘을 활용하여 총 대피시간 단축 및 실시간 대피 안내가 가능할 것으로 판단된다.

본 연구에서는 대기인원 계산을 위한 출구의 보행흐름을 출구의 폭만을 고려하였다. 하지만 항상 출구의 최대 폭만큼 인원들이 대피할 수 있는 것은 아니다. 따라서 대기인원 계산에 대한 보다 정교한 수식이 필요하다. 또한 알고리즘의 실험이 단층 구조에서만 수행되었으므로 다층 구조에 적용하였을 경우 발생할 수 있는 문제점 및 할당규칙 등을 보완하는 향후 연구가 필요할 것으로 판단된다.

감사의 글

본 연구는 국민안전처 재난안전기술개발기반구축사업 ("NEMA-기반-2014-116")의 연구비 지원으로 수행되었습니다.

References

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C., 1990, Introduction to Algorithms, MIT Press, p. 329.
2. Chalmet, L. G., Francis, R. L. and Saunders, P. B., 1982, Network models for building evacuation, Fire Technology, Vol. 18, No. 1, pp. 90-113.
3. Chen, P. H. and Feng, F., 2009, A fast flow control algorithm for real-time emergency evacuation in large indoor areas, Fire Safety Journal, Vol. 44, No. 5, pp. 732-740.

4. Fruin, J. J., 1992, Designing for pedestrians, Public Transportation United States, http://ntl.bts.gov/DOCS/11877/Chapter_8.html
5. Gwynne, S., Galea, E. R., Owen, M., Lawrence, P. J. and Filippidis, L., 1999, A review of the methodologies used in the computer simulation of evacuation from the built environment, *Building and Environment*, Vol. 34, No. 6, pp. 741–749.
6. Hamacher, H. W. and Tjandra, S. A., 2001, Mathematical modelling of evacuation problems: A state of art, *Fraunhofer–Institut für Techno–und Wirtschaftsmathematik, Fraunhofer (ITWM)*, pp. 1–44.
7. Kim, S., Shekhar, S. and Min, M., 2008, Contraflow transportation network reconfiguration for evacuation route planning, *Knowledge and Data Engineering, IEEE Transactions on*, Vol. 20, No. 8, pp. 1115–1129.
8. Kisko, T. M. and Francis, R. L., 1985, EVACNET+: a computer program to determine optimal building evacuation plans, *Fire Safety Journal*, Vol. 9, No. 2, pp. 211–220.
9. Kuligowski, E. D., Peacock, R. D. and Hoskins, B. L., 2005, A review of building evacuation models, MD: US Department of Commerce, National Institute of Standards and Technology, pp. 1–29.
10. Kwak, S. Y., Nam, H. W. and Jun, C. M., 2012, An Optimal Model for Indoor Pedestrian Evacuation considering the Entire Distribution of Building Pedestrians, *Journal of the Korean Society for Geospatial Information Science*, Vol. 20, No. 2, pp. 23–29.
11. Lovas, G. G., 1995, On performance measures for evacuation systems, *European Journal of Operational Research*, Vol. 85, No. 2, pp. 352–367
12. Li, Q., Feng, Z., Li, Q. and Zong, X., 2010, Multiobjective evacuation route assignment model based on genetic algorithm, *Proc. of the 18th International Conference, Geoinformatics*, pp. 1–5.
13. Lu, Q., George, B. and Shekhar, S., 2005, Capacity constrained routing algorithms for evacuation planning: A summary of results, *Advances in spatial and temporal databases, Springer Berlin Heidelberg*, pp. 291–307.
14. Nam, H. W., Kwak, S. Y. and Jun, C. M., 2014, Developing a Cellular Automata–based Pedestrian Model Incorporating Physical Characteristics of Pedestrians, *Journal of Korea Spatial Information Society*, Vol. 22, No. 2, pp. 53–62.
15. Pursals, S. C. and Garzón, F. G., 2009, Optimal building evacuation time considering evacuation routes, *European Journal of Operational Research*, Vol. 192, No. 2, pp. 692–699.
16. Pursals, S. C. and Garzón, F. G., 2013, Building evacuation: Principles for the analysis of basic structures through dynamic flow networks, *Journal of Industrial Engineering and Management*, Vol. 6, No. 4, pp. 831–859.
17. Wang, X. and Liu, H., 2012, An evacuation algorithm for large buildings, *Proc. of the 10th World Congress, Intelligent Control and Automation (WCICA)*, pp. 2497–2502
18. Xie, J., Li, Q., Wan, Q. and Li, X., 2014, Near optimal allocation strategy for making a staged evacuation plan with multiple exits, *Annals of GIS*, Vol. 20, No. 3, pp. 159–168.
19. Yusoff, M., Ariffin, J. and Mohamed, A., 2008, Optimization approaches for macroscopic emergency evacuation planning: a survey, *Proc. of the ITSim 2008, Information Technology*, Vol. 3, pp. 1–7.