

유전 알고리즘을 이용한 지하철 객차 내 승객의 최적대피경로 탐색

최재혁¹ · 박지혜¹ · 최수현¹ · 김남문¹
광운대학교 전자공학과¹

Research finding optimized evacuation route of people in subway passenger cars using genetic algorithm

Jae Hyuk Choi¹, Ji Hye Park¹, Su Hyeon Choi¹, Nam Moon Kim¹
¹ Dept. of Electronic Engineering, Kwangwoon Univ.

ABSTRACT:

There have been subway conflagrations such as Daegu subway conflagration at 2003, Washington D.C conflagration last month and so on. Compared to that, proper evacuation route is far from satisfactory. So this paper suggests optimized route when subway's passengers evacuate from passenger cars. For conducting our experiment, We made temporarily a model of subway station which is made up with 8 passenger cars and 3 exits. Using genetic algorithm, we found the optimized route that first and second passenger cars are optimized to first exit and third, fourth, fifth and sixth passengers cars are optimized to second exit and finally seventh, eighth passengers cars are optimized to third exit. It is expected that real subway station is applied to our experiment by developing passenger distribution algorithm.

Key Words: Genetic algorithm, Optimized evacuation route, Subway distribution evacuation

1. 서론

지난달 10 일에 발생하여 130 명의 사상자가 발생한 의정부 화재 사건을 비롯해 요즘 지구촌 곳곳에서 화재와 같은 재난으로 인한 사건사고가 끊이지 않고 있다. 지상에서 발생한 경우는 건물 옥상이나 화장실, 혹은 밖으로 대피할 수 있는 경우가 있는 반면에 지하철과 같은 지하에서 발생한 경우는 지상으로 올라가는 방법밖에는 없기 때문에 더 큰 피해가 생길 수 있다. 그러나 현재 우리 나라에서의 대피경로 방안이나 연구는 미비한 실정이다. 따라서 지하철 내에서 대피할 수 있는 최적경로를 산출하는 것을 목적으로 연구를 수행하였다.

본 논문에서는 2003 년 대구 지하철 화재참사와 지난 12 일 발생한 워싱턴 DC 지하철 사고와 같이 역사 내에서 대피해야 하는 경우일 때 각 객

차마다 최적화된 경로를 제안한다. 총 8 개의 객차와 3 개의 출구를 가상공간에 형상화 하여 각각의 객차에서 어느 출구로 빠져나가야 최단 거리가 걸리는지 유전 알고리즘을 통해 산출해 냈다. 지하철 역사 내를 도면으로 나타내어 각각의 객차에서 세 개의 출구까지의 직선거리를 측정하여 가장 짧은 거리의 출구를 알려주는 것이다. 아직까지는 가상공간을 만들어서 연구를 수행하였지만 향후에는 현재 운영되고 있는 지하철의 구조에 적용하여 실제 각각의 객차마다 최적의 대피경로를 산출해 낼 수 있을 전망이다.

2. 선행연구 고찰

2.1 기존의 지하철 분산대피 연구

기존의 승객위치정보 기반 지하철 분산대피 안내 시스템[1] 연구에 의하면 지하철내의 AP와 승객들이 사용하고 있는 스마트폰을 이용하여 객차 별 승객의 수를 추정하였으며, 추정된 승객의 수를 이용하여 특정 AP를 사용하고 있는 그룹의 승객마다 어느 출구로 대피해야 하는지 스마트폰으로 안내해주는 방식을 통해 객차별 승객을 분산 대피시킬 수 있는 시스템을 제안하였다. 이것에 적용할 수 있는 방안을 생각한 것이 본 논문의 내용이다. 지하철 객차 별로 최적화된 출구를 알려주면 훨씬 효율적으로 대피할 수 있기 때문이다.

2.2 유전 알고리즘(GA) 및 GA를 활용한 분산 대피 연구

유전 알고리즘(Genetic Algorithm)[2]은 1975년에 존 홀랜드(John Holland)에 의해서 개발된 생물의 진화를 모방한 진화 연산의 대표적인 기법으로, 최적화 문제를 해결하는 기법 중의 하나이다. 문자열 형태의 유전자로 문제에 대한 해를 표현하고 이 해가 얼마나 적합한지를 적합도 함수를 통해 계산하며 선택, 교차, 변이 등의 연산을 사용한다.

유전자 알고리즘을 이용한 실내 보행자 최적 대피모형[3]의 연구에서는 실내공간에서 사람들의 분포에 따라 여러 개의 출구가 존재할 때 출입문 혹은 계단 등에서 사람들이 몰리는 병목현상이 발생하는 경우에 대피시간을 최소화하고자 하였고, 이것을 유전자 알고리즘을 적용하여 해결하고자 하였다. 이 연구에서의 모형은 건물내 사람들의 분포에 따라서 최소 대피시간을 나타내도록 설계하였으며, 각 방 안의 사람들이 가장 빠른 시간에 대피할 수 있는 경로를 알 수 있다. 이것을 토대로 하여 지하철 객차 내에서 여러 개의 출구로 대피하는 경우에 병목현상을 고려하여 최소 시간을 구하려 하였으나 수행하는데 어려움이 있어 단순히 출구까지의 거리만을 생각하여 본 논문을 작성하였다.

3. 연구 개요

3.1 대피모형

대피모형을 만들기 위해 임의의 지하철 객차와 출구를 만들었다. 2cm x 2cm의 정사각형 총 60개(10 x 6)로 구성된 임의의 공간을 만들었으

며 공간에는 객차는 총 8개, 출구는 총 3개가 구성되어있다. 각 객차 별 출구까지의 거리는 객차 출입구에서 직선거리로 산정하였으며 객차별 인원은 7명으로 할당하였다. 이 때 각 객차별로 각 출구까지의 거리는 아래 Table 1과 같으며 지하철 가상공간을 시각화하면 아래 Fig.1과 같다.

Table 1 table of the distance between passenger cars and exit (unit : cm)

	1	2	3	4	5	6	7	8
Exit 1	8.2	8.9	10	11.3	12.8	14.4	16.1	17.9
Exit 2	10.6	9.4	8.5	8	8	8.5	9.4	10.6
Exit 3	17.9	16.1	14.4	12.8	11.3	10	8.9	8.2

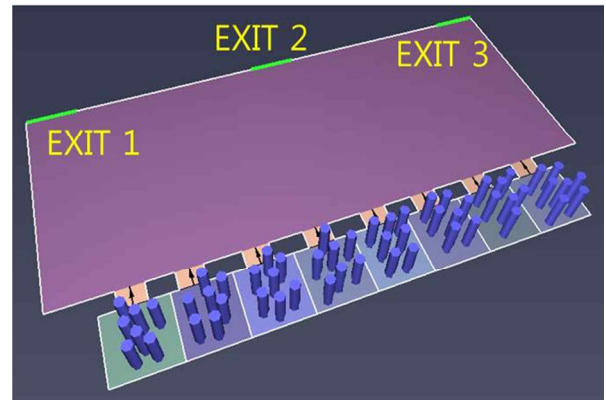


Fig. 1 virtual space for experiment

3.2 목적함수 설정

목적함수는 아래와 식과 같이 각 객차 별로 각 출구까지의 거리의 합으로 표현 될 수 있다.

$$d = \sum_{i=0}^7 \sum_{j=0}^2 distance [i][j] \quad (1)$$

distance[i][j]는 i 번째 객차에서 j 번째 출구까지의 거리를 의미한다.

4. 실험 환경설정 및 결과

4.1 실험을 위한 설정

C code 상에서 table 1 각 객차 별로 각 출구까지의 거리를 2 차원 배열로 입력한다. EDISON에서 제공하는 플랫폼의 testary[]는 1~8까지의 각 객차를 의미하며 testary[]의 요소에는 exit 1~exit3이 들어간다. EDISON 플랫폼에서 제공하

는 Real Mode 의 경우 Testary[]의 요소들에는 실수 값이 들어가기 때문에 Fig.2 에서 실수를 정수로 변환해주는 작업을 수행해준다. 이렇게 되면 Output 으로 실수가 나오더라도 반올림 혹은 반내림 함으로써 결과를 평가할 수 있다. 초기 개체 수는 10, 반복 회수는 500, mutation 확률은 0.08, crossover 확률은 0.8 로 각각 설정했을 때 최적의 결과가 나타남을 실험을 통하여 확인하였다.

```
#include <stdio.h>
#include <math.h>
#define cell_number 8 // 지하철 객차의 총 개수
#define exit_number 3 // 지하철 종구의 총 개수
float objfunc(float testary[])
{
    float sum = 0;
    float result = 0;
    for (int a = 0; a < cell_number; ++a) // Edison testary[]에 들어오는 실수들을 정수로 변환해서 각 종구로 나타내줌
        if (1.0f <= testary[a] && testary[a] < 1.5f)
            testary[a] = Floor(testary[a]);
        else if (1.5f <= testary[a] && testary[a] < 2.0f)
            testary[a] = ceil(testary[a]);
        else if (2.0f <= testary[a] && testary[a] < 2.5f)
            testary[a] = Floor(testary[a]);
        else if (2.5f <= testary[a] && testary[a] < 3.0f)
            testary[a] = ceil(testary[a]);
    }
    float distancel[8][3] = { // distancel[][]의 행은 객차 수, 열은 종구 수이며 배열의 요소는 각 객차에서 종구까지의 거리
        { 9.2, 10.6, 17.9 },
        { 8.9, 9.4, 16.1 },
        { 10, 8.5, 14.4 },
        { 11.9, 9, 12.8 },
        { 12.8, 8, 11.3 },
        { 14.4, 8.5, 10 },
        { 16.1, 9.4, 8.9 },
        { 17.9, 10.6, 8.2 },
    };
    for (int i = 0; i < cell_number; ++i)
        if (testary[i] == 1.0f) // 1번째 객차가 1번종구로 탈출할 경우
            sum += distancel[i][0];
        else if (testary[i] == 2.0f) // 1번째 객차가 2번종구로 탈출할 경우
            sum += distancel[i][1];
        else if (testary[i] == 3.0f) // 1번째 객차가 3번종구로 탈출할 경우
            sum += distancel[i][2];
    }
    result = -sum; // EDISON 플랫폼에서 최소값을 구하기 위해 - 부호를 붙여줌
    return result;
}
```

Fig. 2 C source code

4.2 순서도

전체 시스템 순서도 흐름은 아래의 Fig.3 과 같다.

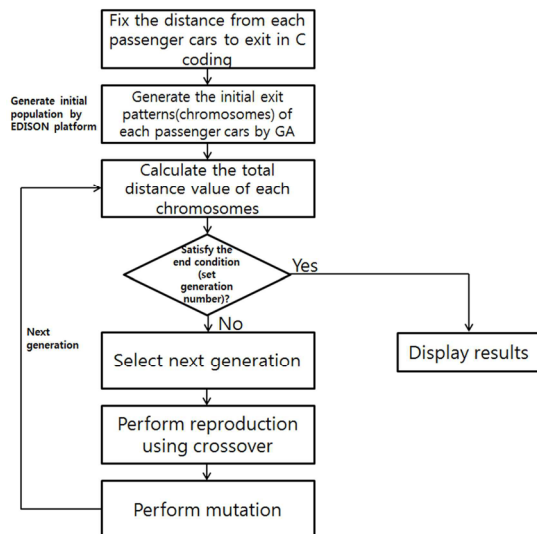


Fig. 3 flow chart for experiment

4.3 실험 결과

Putty 에서 결과를 살펴본 결과 각각 객차에 7 명이 탑승하고 있을 경우 최단 시간 내에 탈출하는 결과가 아래 Fig.4 와 같이 도출되었다.

```
Real coded genetic algorithm
This program finds the maximum value in the function
with the constraints
1 <= x0 <= 3
1 <= x1 <= 3
1 <= x2 <= 3
1 <= x3 <= 3
1 <= x4 <= 3
1 <= x5 <= 3
1 <= x6 <= 3
1 <= x7 <= 3
the ga found an optimum at the point (1.01663, 1.10418, 2.4895, 1.67863, 1.66317, 1.7358, 2.96453, 2.99966)
best of generation data are in 'output.txt'
```

Fig. 4 result of experiment

C code 에서 설정하였던 반올림, 반내림을 고려하여 이를 표로 정리하면 아래와 같다.

Table 2 result of experiment

	1	2	3	4	5	6	7	8
Exit	1	1	2	2	2	2	3	3

Fig.5 는 대피프로그램 중 하나인 Pathfinder 를 사용하여 도출된 최적해를 시각화한 모습이다.

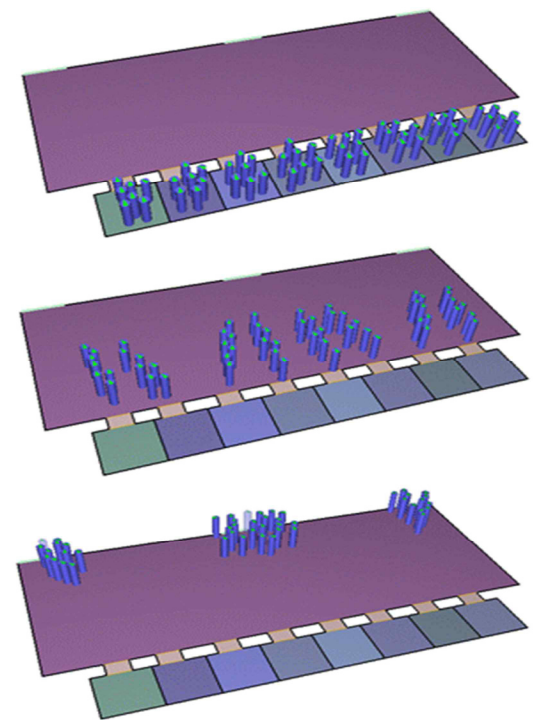


Fig. 5 Visualization of experiment

3. 결 론

본 논문에서 지하철 내에서 대피해야 하는 경우 일 때 각 객차마다 최단 거리의 경로를 제안하고, EDISON 에서 제공하는 플랫폼을 통해 유전 알고리즘을 이용하여 결과를 확인하였다. 이 연구를 바탕으로 하여 실제적인 대피구현을 위해 같은 출구로 많은 인원이 몰렸을 때 병목현상을 고려하여 인구를 분산시키는 알고리즘을 개발하였으나 플랫폼을 사용하여 결과값을 산출하는 과정은 향후 연구해야 할 과제이다.

감사의글

프로젝트를 진행하는 동안 여러모로 조언해주신 김영익 교수님을 비롯하여 연구실 선배님들께 감사의 말씀을 드립니다. 점심시간이나 틈틈이 시간이 되실 때마다 많은 조언과 격언 덕분에 이렇게 논문을 쓸 수 있었던 것 같습니다. 앞으로도 더 많이 배워서 좋은 결과를 얻을 수 있도록 노력하겠습니다.

참고문헌

- [1] 김남문 외 1 명, 2014, 승객 위치정보 기반 지하철 분산대피 안내 시스템, 한국산학기술학회 논문지, 15(7) pp. 4528-4535
- [2] 유전 알고리즘, 네이버 위키백과
- [3] 광수영, 2012, 유전자 알고리즘을 이용한 실내 보행자 최적 대피모형, 서울시립대학교, pp.1-65