International Journal of Smart Home
Vol.8, No.1 (2014), pp.197-206
http://dx.doi.org/10.14257/ijsh.2014.8.1.21

The Path Search Be Applied Algorithm for Spread of Flame and Smoke in Fire Simulation

Eunju Kim[1], Yongzhe Xu[1], KyunjooLee[2], Jaesug Ki[1,*] and Byungsoo Lee[1,*]

[1]Department of Computer Engineering, Incheon National University, Korea
[2]MaxOn Soft, Co., Ltd, Korea
leebone28@hanmail.net,yongzhexu@hotmail.com, lisun@maxonsoft.com,
{skyblueki, bsl}@incheon.ac.kr

Abstract

Fire is one of the disasters that often occur in everyday life. There has been much research done to build a fire Simulator. In this paper, we propose a fire Simulator that is applied optimal path search algorithm to arrange the quick evacuation. This simulator considers the structure of a building and the location where the fire has initially occurred, applying it to the simulator. Especially when implementing the training simulation for firefighters, optimized evacuation path for the required virtual rescuer awaiting rescue was informed to implement more realistic firefighting training simulator. For the simulation for the test, fixed data was set to apply the shortest path search algorithm and flame smoke spread algorithm. As a result of the test, the optimized path was derived through avoiding the flame occurred from flame smoke spread algorithm and following the shortest path search algorithm.

Keyword: *Fire Simulator, optimal path, Greedy method, Binary programming*

1. Introduction

At present, the modern buildings consist of large-scale, multifunctional and complex and automatic alarm system cannot afford to reach the demand of evacuation. Because of the people’s limitation, to get the information from different parts in a fire will be difficult. If the evacuation can’t be effectively finished in the preliminary stage, it will result the unnecessary casualties. Therefore, setting up a safe and effective evacuation method is important in a fire.To escape the safe and efficient in fire, the study of algorithms have been developed.

In this paper, the implemented fire training simulation was applied the optimal path search algorithm and the fire smoke diffusion algorithm.For the existing optimized path search algorithm, there are Greedy method, Linear Programming Simplex Method, Binary Programming, Network Simplex Method, and for flame smoke spread algorithm, many algorithms are under study such as Jordan Method, Geometric Equation, etc. Applying these algorithms informs the optimized evacuation path for the required rescuer awaiting rescue especially when implementing the training simulation for firefighters, providing implementation of more realistic firefighting training simulation. To enhance responsive ability of first-line firefighters with many emergency situations in real fire situation, this will be used as the training simulator for the firefighters.

In Section 2 of this paper examines related work, and in Section 3, the proposed the fire Simulator is tested to evaluate its method. Finally, Section 4 concludes.

ISSN: 1975-4094 IJSH

2. Related Work

2.1. The Optimal Path Search Algorithm

2.1.1. Greedy Method

In case the burning point of option finds in the building, to avoid the ignition point, it searches for the shortest distance at the current position. And then, the people provide guidance so that they can into the building. The Optimal path search algorithm is implemented avoiding ignition point from the current position. There is several algorithm for the optimal path searching, but as associated algorithms with the simulation program implementation, there are the following things.

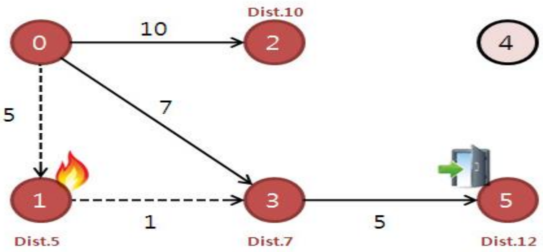


Figure 1. The Result of Shortest Path Search

As shown in Figure 1, shortest path algorithm can calculate the shortest distance from the starting point to all peak points. First of all, the closest distance of peak point from the starting point is found, and the distance (weight) to the peak point found from the starting point is memorized separately (distance). Then from all the peak points found, peak points with the lowest weight is continued to be found to explore all optimized path. The key is that the weight is accumulated, and the lowest weight of peak point is continued to be found, in other words, Greedy method (Greedy exploration method)[9].As shown in Figure 1, before ignition, 0-1-3-5 is 11 to be the shortest path, but after ignition, since this path is not possible, a new exploration of shortest path is required. In other words, the optimized path considering the weight is 0-3-5[1].

2.1.2. Linear Programming Simplex Method

Linear Programming has the objective function that expressed as a linear function, and the constraints that expressed as a linear inequality. It is the best techniques to obtain best solution for appropriate distribution of limited resources.

$$Minimize(Maximize)\, Z = \sum_{i=1}^{n} c_i x_i (1)$$

constraint:

$$kst - equation\text{:} \sum_{i=1}^{n} s_{ik} x_i \le S_k (2)$$

$$\sum_{i=1}^{n} s_i x_i \ge S_k \Rightarrow \sum_{i=1}^{n} (-s_i x_i) \le (-S_k),$$

$$\sum_{i=1}^{n} s_i x_i = S_k \Rightarrow \sum_{i=1}^{n} s_i\, x_i \le S_k \text{과} \sum_{i=1}^{n} (-s_i x_i) \le (-S_k)$$

Equation(1) is converted as a formula that satisfies the constraint by dividing into two formula(2).The equation (1), (2) search the path of the optimized conditions by including several conditions.

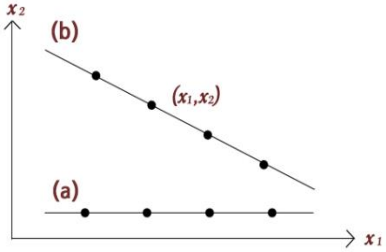


Figure 2. Best Solution in Linear Programming Simplex Method

Figure 2 shows the optimal path to the punctual coordinates that exit is the best path to meet. In case (a) is no the ignition point around the exit, and in case(b) find other the exit to starts the fire. Therefore, it is a set of coordinates to find the optimal path [2].

2.1.3. Binary Programming

The problem of shortest path exploration is included in the optimization problem between peak points. Binary Programming is a method of exploring the optimized path by planning minimum cost problem with the variable 0 or 1. It is defined in Formula (3), (4).

$$Minimize(Maximize)Z = \sum_{i=1}^{n} c_i x_i (3)$$

constraint:

$$kst - equation: \sum_{i=1}^{n} s_i x_i \leq S_k (4)$$

$$x_i \leq 0, -x_i \leq 0, x_i \leq 1, -x_i \leq -1$$

Binary Programming can be thought as a concept of a tree, and a tree with the depth of two parts in all nodes not differing 1 or more is called balanced tree, and binary search tree includes searchable record according to the key value. For the average time take to find the key in the binary search tree to be minimum, it is in well arranging the key [3].

2.1.4. Network Simplex Method

From the point of view of the minimum cost problem either maximum flow or the minimum cost, this method is to search the optimal path or optimization using the value or a variable that take the lowest cost.

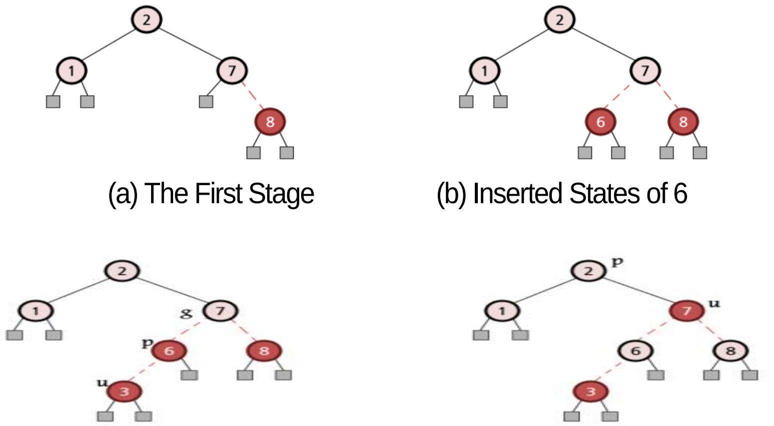
In general concepts, while $i$node and j node is related, c_{ij} is defined as the flow or the costs by $i$and$j$.Here, x_{ij} is defined by assigning of I and j, and has a value 0 or 1. Generalizing this, it can be defined as equation (5) and (6).

$$Minimize(Maximize)\mathrm{Z} = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} (5)$$

constraint:

$$kst - equation: \sum_{i=1}^{n} x_{ik} - \sum_{j=1}^{n} x_{kj} = S_k (6)$$

$$x_{ij} \leq 0, -x_{ij} \leq 0, x_{ij} \leq 1, -x_{ij} \leq -1$$



(a) The First Stage (b) Inserted States of 6

(a) Inserted States of 3 (d) The Results of Optimal Path Search

Figure 3. Optimal Path Search in Binary Tree

Binary Programming and Network Simplex Method can deduce from the optimization using binary tree. It is a matter of choice searching a path to guide the exit to avoid the ignition point. In other words, the path selection is added a new node as binary tree. When a new node is added, the new node has 0 or 1 and is search minimum costs. Also it is possible to calculate the optimal value to ensure minimum costs. Figure 3 shows the optimal path search in binary tree, the result of search is 2-7-6-3[4].

2.2. The Fire Smoke Diffusion Algorithm

2.2.1. Linear Algebra Gause-Jordan Method

Gause-Jordan elimination is made the coefficient matrix with only 0 or 1using the row operations and is made its column that has all 0 the other elements in column that has 1.This is the process of creating so that the above does not have 1 than have one line. The equation (7) is shown in the following.

$$\begin{gathered} \mathrm{AX} = \mathrm{C} \\ \mathrm{X} = A^{-1}C \end{gathered} \tag{7}$$

ForA^{-1}obtaining, the Gause-Jordan Method is to convert to A from I(Identity matrix) after existing A and I are listed. When the flames and smoke spread, it is the way that fire and smoke fills to remove then one by one in matrix by many different paths which is possible spread [5, 6].

2.2.2. Geometric Equation

When fire and smoke spreads, it is a common way that implement to spread in the form of a circle or a rectangle. In equation $(x-x_0)^2+(y-y_0)^2=r^2$, r is the radius, and (x_0, y_0) is the location of the center. As shown in Figure 4, the flames spread out of the circle in flash point.

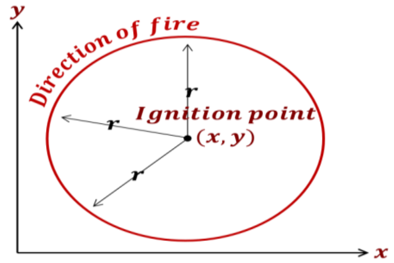


Figure 4. Geometric Equation Method

In addition, another expression, If fire spreading implemented as the form of a square spreads, 4 can be define. Here, the rectangular is half of a horizontal, b is the length of the half, 4 means the position of the center. When a fire occurs usually, we need to consider the two such as when will spread the diameter of the circle, when spread on direction like corner hanmo rectangular. So [4, 5] of algorithm introduced for implement as the similar form with the real world.

3. Simulation

3.1. The Implementation of Optimal Path Search Algorithm

Even if how to burn, we have to implement the algorithm that can guide the shortest path so that it can avoid ignition point. Therefore, in this paper, optimized path search algorithm was developed from the current location, avoiding the ignition point, to the exit possible to evacuate in the shortest distance. In design and implementation, Binary Programing method is applied, and it is implemented in greedy method.

1. The distance of the node is found excluding the current node.
2. The shortest distance node is brought in.
3. Node passed before is excluded.
4. 1 ~ 3 method is repeated.

```
intmin_index = -1;
unsigned pos = start;
m_paths[pos].distance = 0;
while(duplicate[pos] == false)
{
duplicate[pos] = true;
std::vector<__int64>& row = m_graph[pos];
for(i = 0; i<arr_size; i++)
{
        if(row[i] + m_paths[pos].distance
<m_paths[i].distance)
        {
                m_paths[i].distance
= row[i] + m_paths[pos].distance;
                m_paths[i].header = pos;
        }
}
```

```
    min_index = -1;
    for(i = 0; i<arr_size; i++)
    {
            if(duplicate[i]) continue;
            if(min_index == -1 || m_paths[i].distance<m_paths[min_index].distance)
                    min_index = i;
    }
    pos = min_index;
    if(min_index == -1) break;
}
```

Figure 5. Program Source for Shortest Path Search Algorithm of Simulation Program

3.2. The Implementation of Fire Smoke Diffusion Algorithm

When a fire occurs practically in a room, the flame and smoke always spread to different directions depending on the situation in the building. Because the direction of spread and the rate of spread is most of the conditions, it must set artificially as the similar form with the real world. Thus, it is essential to use optimization algorithms of the fire smoke spread.

In this paper, the simulation of fire smoke diffusion algorithm implemented the simulation by using spread of smoke and fire as start the ignition of fire. The order of implementation can be as follows.

1. The value is obtained by condition.
2. Increase the value.
3. It also increases when overlapping area is included.
4. The other area is found neighboring area.
5. 1 ~ 4 method is repeated.

```
static boolpt_in_circle(POINT pt, POINT center, unsigned r)
{
double r2 = (pt.x - center.x) * (pt.x - center.x) +
(pt.y - center.y) * (pt.y - center.y);
if(r2 <= r * r) return true;
return false;
}

int&fire_value = g_sensor_status_fire[object.mobject];
int&smoke_value = g_sensor_status_smoke[object.mobject];

if(fire &&fire_value> 0)
{
__int64 prev = g_sensor_status_firetime[object.mobject];
if((__int64)GetTickCount() - prev>g_level_fire * 1000)
{
        fire_value++;
        g_sensor_status_firetime[object.mobject] = GetTickCount();
}
}
```

```
if(!fire &&smoke_value> 0)
{
__int64 prev = g_sensor_status_smoketime[object.mobject];
if((__int64)GetTickCount() - prev>g_level_smoke * 1000)
{
            smoke_value++;
            g_sensor_status_smoketime[object.mobject]
= GetTickCount();
}
}
```

Figure 6. Program Source for Smoke Flame Diffusion Algorithm of Simulation Program

3.3. Result of Simulation

Experiments which include the collection of basic data for the simulation were conducted in fire training simulation. However, In the case of fire training simulation, the result is completely different as the terms of the simulation. Because the map will be different the results by the optimal path search and the fire smoke diffusion algorithm, this part was conducted several times. Regarding the default value setting for the test, by assigning various values on the environment setting screen on the simulation program, the time taken to explore the optimized path of the applied algorithm was measured. The Table 1 shows the test data setting table, and because transmission is a value transmitted as soon as it is detected regardless of the ignition situation, the default value of 10 seconds was maintained in all test cases. As the case become larger, it applies to a larger fire, and in this case, the shortest distance must be explored quickly.

Table 1.The Experimental Data Set

CASE	transmission (sec)	Speed of fire (pixel/sec)	Level of fire (sec)	Speed of smoke (pixel/sec)	Level of smoke (sec)
CASE1	10	5	30	8	30
CASE2	10	10	35	13	35
CASE3	10	15	40	18	40

Table 2 is the results of experiments using data set of Table 1. In the table of results, the meaning of each of the values is the amount of time that the shortest distance can calculate.

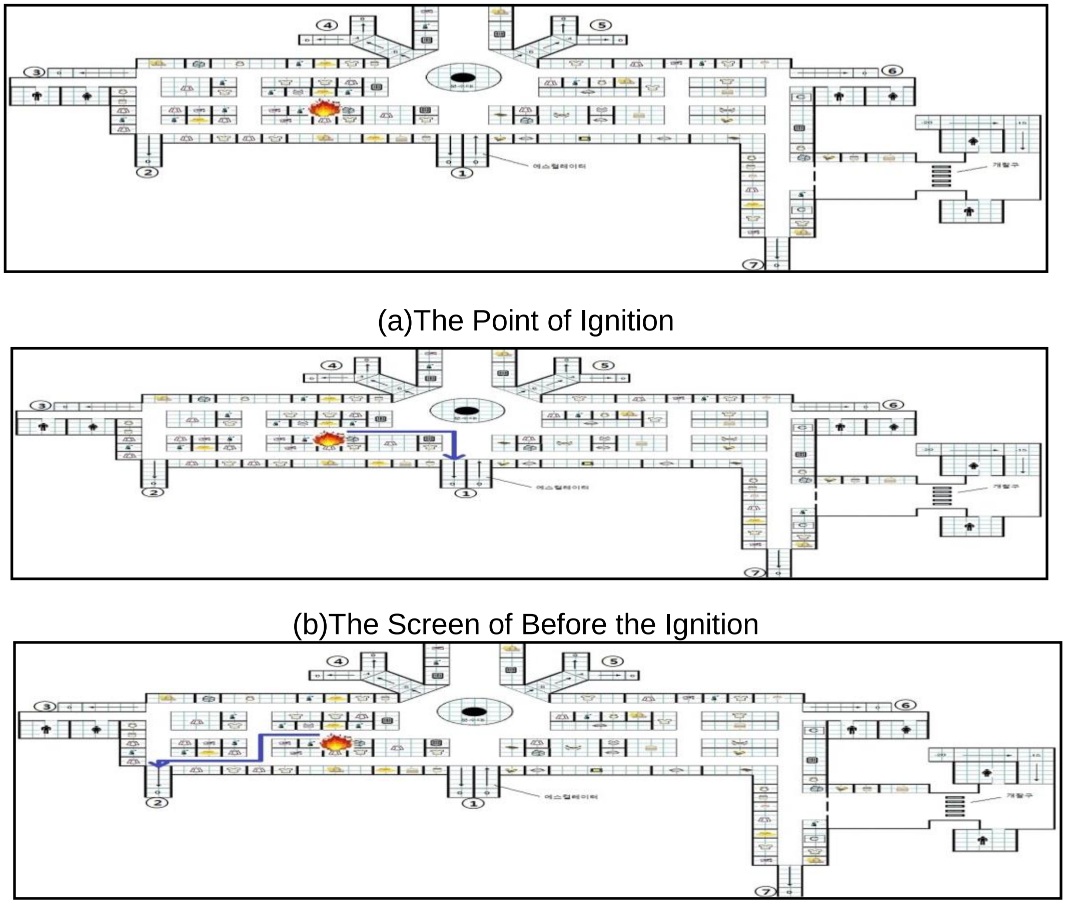
Table 2. Experiment Result for Automatic Disaster Information Broadcast System

CASE	1 path	2 path	3 path	4 path	5 path	6 path
CASE1	0.94	0.93	0.72	0.72	0.72	0.72
CASE2	0.89	0.87	0.62	0.61	0.61	0.58
CASE3	0.79	0.77	0.57	0.54	0.75	0.71

Table 2 shows the values of experiment and the values of experiment was expressed only two decimal places to normalize. Looking at the numerical result, the ignition point is avoiding the relevant ignition point to guide to the nearest emergency exit (Number 2 passage). Also, when the ignition point is relatively far (Number 4 passage, number 5 passage, number 7 passage), it can be guided to the nearest emergency exit regardless of the ignition point, therefore, we can see that the result is derived in similar level to each other.

As shown in Table 1, CASE means a large fire so that rise and the results of search the shortest path so that fast speeds of fire.

When the ignition occurs near the emergency exit (Number 1 passage), it must be guided to another emergency exit, therefore some time is required to search the emergency exit. Also, the closer the ignition point is (Number 1 passage), the flame spreads irregularly requiring more time, because it must be measure to find out to which emergency exit it must guide.



(a)The Point of Ignition

(b)The Screen of Before the Ignition

(c)The Screen of After the Ignition

Figure 7. Experiment Result of Simulation Program

Figure 7 shows the experiment result of simulation program. Figure 3 shows the screen of before the ignition. The screen which displayed the result of the experiments.

As shown in Figure 7, the exit of the shortest path was chosen as the nearest exit. However, after the fire progress, the direction of fire was changed by fire smoke diffusion algorithm. Therefore, the exit of the shortest path was chosen by optimal path search algorithm again.

5. Conclusion

In this paper, the training fire simulation was implemented to firefighters. When implementing the training simulation for firefighters, Binary Programing method is applied, and it is implemented in greedy method. Especially when implementing the training simulation for firefighters, providing implementation of more realistic firefighting training simulation. For the simulation for the test, fixed data was set to apply the shortest route search algorithm and flame smoke spread algorithm. As a result of the test, the optimized route was derived through avoiding the flame occurred from flame smoke spread algorithm and following the shortest route search algorithm.

Acknowledgement

This paper was researched (2012~2014) The Next Generation Fire protection & Safety Core Technology Development program of National Emergency Management Agency (NEMA) of Republic Korea.

References

[1] H.-S. Lee and J.-D. Kim, "A Method to determine Search Space of Hierarchical Path Algorithm for Finding Optimal Path", The Korean Institute of Maritime Information and Commucation Sciences, **(2007)**, pp. 565-569.

[2] C.-S. Sim and J.-S. Kim, "Performance Evaluation for One-to-One Shortest Path Algorithms", KIISE, vol. 29, no. 11-12, **(2002)**, pp. 634-639.

[3] J.-S. Han and K.-S. Jeon, "Search for an Optimal-Path Considering Various Attributes", Journal of Korean Society of Transportation, vol. 26, no. 1, **(2008)**, pp. 145-153.

[4] L. Bass, P. Clements and R. Kazman, "Software Evaluation", ADDISON-Wesley, **(2007)**.

[5] G. Shmueli, N. R. Patel and P. C. Bruce, "The DataMining for Business Intelligence", WILEY, **(2009)**.

[6] P.-N. Tan, M. Steinbach and V. Kumar, "Introduction to Data Mining", Addison Wesley Longman, **(2005)**.

[7] T. Whangbo, "Efficient Bidirectional Search Algorithm for Optimal Route", Korea Multimedia Society, vol. 5, no. 6, **(2002)**, pp. 745-752.

[8] Y.-G. Jung and C.-H. Park, "Development of the Shortest Route Search Algorithm Using Fuzzy Theory", Journal of Korean Society of Transportation, vol. 23, no. 8, **(2005)**, pp. 171-179.

[9] B.-M. Lee, J.-I. Park and W.-G. Park, "Design and Implementation of a Real-time Automatic Disaster and Information Broadcasting System", The Society of Digital Policy & Management, vol. 10, no. 7, **(2012)**, pp. 141-152.

[10] P.-N. Tan, M. Steinbach and V. Kumar, "Introduction to Data Mining", Addison Wesley Longman, **(2005)**.

[11] L. Bass, P. Clements and R. Kazman, "Software Architecture in Practice Second Edition", ADDISON-Wesley, **(2007)**.

Authors

EunJu Kim, received the PhD in computer science and engineering from Incheon National University, Korea, in 2013. Since September 2013, she has been a senior member of research staff with the High-Tech S/W Research Center in Incheon National University, Korea. Her research interests include computer vision, image processing, and Artificial intelligence.

Yongzhe Xu, received his MS in Computer science and engineering from Incheon National University, Korea, in 2009.

He is currently working toward a PhD in computer science and engineering at the same university. His research interests include computer graphic, FDS, virtualization, and Pattern Recognization, Machine Learning.

KyunJuLee, received his bachelor's degree in Department of Philosophyfrom korea university in 1991, 1993-1999 和像音 CEO, 1999-2001 Oz Intern Media project chief, 2001-2002 Muhon virtual Society Game team leader, 2002-2013 MaxonSoft CTO. His interesting about multi-platform game, fire protection training game, 3D Game engine.

Jaesug Ki, received his doctor of Department of industrial engineering from HanYang university in 1993.1993-2000 associate professor in Kangwon National University. 2000-2006 a laboratory chief in KCI. His interesting about fire protection training simulation, sports training simulation, National Defense weapon simulation.

Byungsoo Lee, received his MS in MBA from University of Dongguk, Korea, 1980. Received his Doctor of Science from University of KyongGi, Korea, 1998. He is an Professor of Department of Computer Engineering at the Incheon National University, Korea. His research interests include software design, decision making system, eCRM, RFID/USN, IT convergence.