

屋内火災における危険度を考慮した避難誘導手法の構築

○加藤 利哉 (東京大学), 山下 淳 (東京大学), 浅間 一 (東京大学)

Construction of Evacuation Guidance Method Taking Account of the Degree of Hazard to an Evacuee in the Situation of a Building Fire

○Toshiya KATO (Univ. of Tokyo), Atsushi YAMASHITA (Univ. of Tokyo), and Hajime Asama (Univ. of Tokyo)

Abstract— In the situation of a fire, an evacuation system, which guides pedestrians in a building to exist safely, is required. In this research, the spread of fire and smoke is assumed to be predictable beforehand. Moreover, the method of evaluating evacuee's suffered danger and evacuation time is proposed. Experiments show that our proposed method enables a pedestrian to evacuate safely.

1. 緒言

1.1 背景

建築物で火災が発生した場合、避難という観点からその建物の安全性を検討することは非常に重要である。現在では館内一斉放送や避難誘導灯などの、避難者全員に同じ指示を出す方法が採用されている。しかし、Fig.1のように、火災では普段使える通路が使用不可能になる可能性があり、個々人がどの時刻のどこにいるかによって適切な避難経路は変わる可能性がある。

1.2 先行研究

そのような事情から、建物内から人を迅速かつ安全に避難させる方法として、個々の避難者を適切な方向に誘導して避難させる方法が考案され、それを実現する手段として瀧本ら¹などは避難者を適切に誘導するの装置を開発した。

また、その誘導を評価する指標としては、Kooら²や Saeed Osmanら³のように屋内の全避難者の避難完了までにかかった時間（避難完了時間）や、Saadatsereshtら⁴や Guanquanら⁵のように避難が完了した時の生存者数を用いるのが一般的である。

しかし、これらの研究はいかに避難完了時間の短縮を実現するかや、実際に火災が発生した場合にどの程度の人が避難が可能かという意味での、建物の安全性に対する言及のみである。つまり、避難中に避難者が遭遇する可能性のある、個々の避難者の危険に対する評価が十分になされてはいない。安全な避難路を選択しているかを確認するには、避難者がどれだけの危険を避難中に受けるのかを定量的に把握する必要がある。

そこで、新居ら⁶は火災時での屋内の煙の危険性に着目し、危険ポテンシャルという呼ばれる量を定義した。そして、その危険ポテンシャルが少ない方に避難者を誘導することにより、避難者が危険に遭遇する可能性を下げることに成功した。しかし、この研究では煙以外の危険、例えば炎などを考慮しておらず、その避難誘導路が安全性と避難完了時間の面から最適かは判断できない。

また、Yan-yanら⁷は火災が発生した屋内の温度と一酸化炭素濃度と二酸化炭素濃度を評価指標として、屋内の危険な区画を定量的に割り出した。そして、その区画を避けるように避難者を誘導することで、避難者を危険から遠

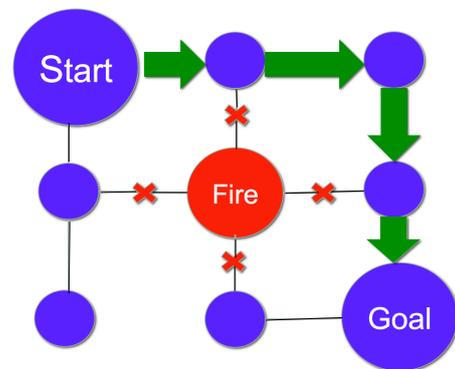


Fig.1 Evacuation Route

ざげられることを示した。だが、Yan-yanらの論文では、様々な危険となりうる要因を考慮しているが、提案した避難誘導路について、安全性だけで避難完了時間の面から評価をしていない。

1.3 目的

そこで本稿では、避難の誘導路が安全であることを評価する必要性から、避難中に個人が受ける可能性のある危険を定量的に評価する手法を確立する。そして、その値を用いて、一人の避難者を、安全でかつ短い時間で避難を完了させることが可能な誘導路を発見することを目的とする。

避難者がどれだけの危険を避難中に受けうるのかを定量的に把握するために、本稿では危険度という概念を導入する。そして、本稿で対象とする危険は温度と一酸化炭素濃度である。

また、火災が発生して屋内各所の温度と一酸化炭素濃度が上昇し、屋内の各所で危険が高まる中で、どのように避難者を誘導すべきかを、避難者の受ける危険と避難完了時間の短縮の観点から評価する手法を確立する。その際、避難者のいる屋内の環境は動的に変化するので、LPA*(Lifelong Planning A*)⁸と呼ばれるアルゴリズムを用いることで、避難者にとって最適な避難路を発見するアプローチを取る。そして、その手法をコンピュータシミュレーションで評価する。

2. 評価手法

2.1 危険度の定義

本稿では危険度を、避難者が避難中に受ける可能性のある身体的損傷や生命活動へのダメージ、と定義する。具体的には、高い温度による火傷や熱中症、そして高濃度の一酸化炭素に暴露し続けることで発生する、めまいや呼吸困難である。避難者の受けた危険度がある値を超えると、その避難者は生存出来ないものとする。

2.2 定式化

避難者が時刻 t_0 [s] における (x_0, y_0) から、出口 $G(x_g, y_g)$ に到着したするとき、避難のすべての軌跡の集合を、 Q とする。このとき、出口 G に到着した時刻 t_g [s] は軌跡によりそれぞれ違うものとする。

避難者が軌跡 $q \in Q$ に沿って、出口に到着したときの累積の危険度 $R(q)$ について考える。 $r_j(q)$ を避難完了時の避難者が受けた、種類 j の危険とする。また、すべての j について $r_j(q) \in [0, 1]$ とする。そのとき、 $R(q)$ は以下のように定義する。

$$R(q) = 1 - \prod_{j=1}^m (1 - r_j(q)) \quad (1)$$

ここで、 m は危険の種類総数。

それぞれの j について、 $1 - r_j(q)$ は j の種類の危険にさらされて、避難者がどの程度安全だったかを表現している。そして、それぞれの種類の危険はすべて独立である。よって、軌跡 $q \in Q$ に沿って出口に到着した時、避難者は $A(q) = \prod_{j=1}^m (1 - r_j(q))$ の程度で安全だったと表現できる。

ここで、 $A(q)$ と $R(q)$ は互いに背反の関係にある。つまり、 $A(q) + R(q) = 1$ 。よって、式 (1) を得る。このことは、 $0 \leq A(q) \leq 1$ と $0 \leq R(q) \leq 1$ であることを満たす。

この $R(q)$ の数値が高くなるほど避難者が受けた危険度は高く、 $r_j(q) = 1$ はその避難者が種類 j の生存が不可能な危険度を負ったことを意味するものとする。

本稿では、温度と一酸化炭素濃度の2種類の危険を扱うので、 $m = 2$ とする。軌跡 q に沿って避難したときに、 $r_1(q)$ を避難者が避難完了時に受けたの危険度と定義する。また、 $r_2(q)$ は避難者が避難完了時まで受けた一酸化炭素中毒の危険度とする。

2.3 温度の危険度

時刻 t [s] における、点 (x, y) の温度を $T(x, y, t)$ [K] とする。避難者のいる地点の温度が高ければ高いほど、避難者は火傷や熱中症などの温度の危険を負うことになると考えられる。そこで、避難者の温度による危険度は、避難者のいる地点の温度に比例すると仮定する。

しかし、常温程度では温度による、人体の影響は見ることができない。つまり、人体に悪影響が起こる温度が必要となる。そこで、 T_0 [K] を人の身体に影響が出始める温度とする。文献⁹によると、人は温度が $317.15K$ を超えると、人体に火傷が出始めると知られている。そこで、避難者に影響が出始める温度を $T_0 = 317.15$ と定める。

そして、文献⁹によると、温度が $357.15K$ を超えると短時間でも重大な火傷を負うことが確認されている。よって、 T_{limit} [K] を人が生存不可能な温度として、 $T_{limit} = 357.15$

とする。

次に、避難者が温度による危険を負った場合、危険を負った温度よりも温度が低ければ、 T_0 よりも温度が高かったとしても、危険をそれ以上負わないものと仮定する。つまり、避難者 p の温度に対する危険度は、避難者 p が通過した地点の最も高い温度によって決定される。

そこで、これらの定数を元にして、温度の危険度の定式化を考える。地点 (x, y) の時刻 t における温度の危険度を以下の通りに定める。

$$r_1(q) = \max_{(x, y, t) \in q} f_1(x, y, t) \quad (2)$$

ここで、

$$f_1(x, y, t) = \begin{cases} 1 & \text{if } T(x, y, t) \geq T_{limit} \\ a(T(x, y, t) - T_0) & \text{if } T_{limit} > T(x, y, t) > T_0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$q \in Q$ は避難者 p の避難時の軌跡。

また、 $a[K^{-1}]$ は定数で、式 (3) が成り立つように定める。つまり、

$$1 = a(T_{limit} - T_0) \quad (4)$$

$$\Rightarrow 1 = a(357.15 - 317.15)$$

$$\Rightarrow a = 0.025 \quad (5)$$

2.4 一酸化炭素中毒の危険度

文献¹⁰によると、一酸化炭素中毒の重篤の程度は、主に一酸化炭素の濃度と暴露されていた時間によって決まる。

そこで、一酸化炭素の濃度を c [ppm]、一酸化炭素に暴露されていた時間を τ [s] とする。そのとき、文献¹¹によると避難者は、 c と τ を用いて以下のような一酸化炭素中毒の危険を負う。

$$c \cdot \tau \begin{cases} < 1,080,000, \text{ 影響なし} \\ < 2,160,000, \text{ 軽度の作用} \\ < 3,240,000, \text{ 中度ないし高度の影響} \\ = 5,400,000, \text{ 致死} \end{cases} \quad (6)$$

つまり、一酸化炭素の濃度が高ければ高いほど、そして一酸化炭素に暴露されていた時間が長ければ長いほど、避難者は一酸化炭素中毒の危険を負う。そこで、避難者の一酸化炭素中毒の限界許容量を $L_s[\text{ppm} \cdot \text{s}] = 5,400,000$ とする。

次に、 $\rho(x, y, t)$ [ppm] を時刻 t で点 (x, y) における一酸化炭素の濃度とする。普段生活する中で、火災やガス漏れなどの事故に遭遇することがなければ、人間は一酸化炭素中毒になることはない。つまり、ある程度以下の濃度の一酸化炭素に暴露され続けても影響がないことから、人体に影響の出ない最大の一酸化炭素の濃度を把握する必要がある。

そこで、そのような人体に影響が出始める一酸化炭素の濃度を ρ_0 [ppm] とする。文献¹²によると、日本産業衛生学会の許容一酸化炭素濃度は 50ppm と定められている。そして、本稿ではその濃度が人体に影響が出始める一酸化

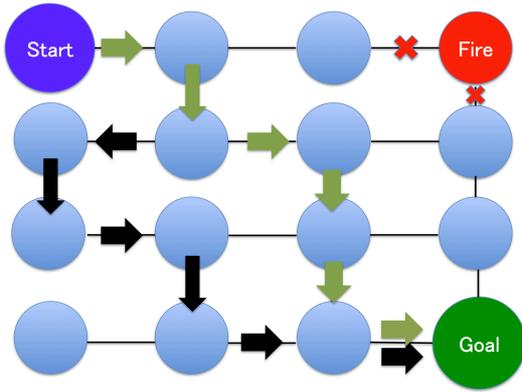


Fig.2 Green Arrows: Shorter Evacuation Route;
Black Arrows: Longer Evacuation Route

炭素濃度として、 $\rho_0 = 50$ とする。つまり、 $\rho(x, y, t) < \rho_0$ では一酸化炭素中毒の危険性がないものと考えられる。

そこで、一酸化炭素の濃度に着目し、一酸化炭素の濃度が ρ_0 未満の時とそれ以上の場合に分けると以下のように定式化ができる。

$$f_1(x, y, t) = \begin{cases} \rho(x, y, t) & \text{if } \rho(x, y, t) \geq \rho_0 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

式 (6) の一酸化炭素の濃度と暴露された時間の関係から、一酸化炭素中毒の危険度の定式化をする。まず、避難者が時刻 t に地点 (x, y) にいたときに、瞬間的に $dt[s]$ の短い時間だけ一酸化炭素に暴露されたとする。そのとき、その避難者は $f_1(x, y, t)dt$ だけの一酸化炭素中毒の危険を負うと考えられる。

これらのことから、軌跡 q に沿って避難者が移動したとすると、式 (6) から一酸化炭素中毒の危険性について以下の式を得る。

$$s(q) = \int_q f_2(x, y, t) dt \quad (8)$$

ここで、一酸化炭素中毒の危険度 $r_2(q)$ は $s(q)$ に比例すると仮定する。そして、 $s(q)$ が一酸化炭素中毒の限界許容量 L_s 以上の場合、その避難者は生存できないので、 $r_2(q) = 1$ となる。よって、一酸化炭素中毒の危険度は以下のように定式化できる。

$$r_2(q) = \begin{cases} 1 & \text{if } s(q) > L_s \\ \frac{1}{L_s} s(q) & \text{otherwise.} \end{cases} \quad (9)$$

2.5 危険度のまとめ

q を避難者の軌跡とする。そのとき、式 (1) に従い温度の危険度 $r_1(q)$ と一酸化炭素中毒の危険度 $r_2(q)$ をまとめると、累積の危険度 $R(q)$ を以下のように表現できる。

$$\begin{aligned} R(q) &= 1 - (1 - r_1(q))(1 - r_2(q)) \\ &= r_1(q) + r_2(q) - r_1(q)r_2(q) \end{aligned} \quad (10)$$

2.6 評価関数

本稿では、避難路を前述の温度と一酸化炭素中毒の危険度と、避難者の避難完了時間で評価する手法を提案する。避難路の評価に危険度だけでなく避難完了時間を考慮する理由は、実際には避難が完了する時間は出来るだけ短い方が避難者にとって望ましいと考えられるからである。

避難者が受ける危険度が小さい避難路を選択するのは、当然必須である。そして、それを達成するためには、温度が低い場所と一酸化炭素濃度が低い場所を通過すれば、低い危険度で避難者は避難を終えることで可能である。しかし、Fig.2 のように危険度が低い場所を通過し続ける避難路を選択した場合に、その避難路での避難完了時間は長くなってしまふ可能性が存在する。

そこで本稿では、危険度と避難完了時間を合わせて評価可能な関数 $E(q)$ を提案する。ここで、 $q \in Q$ は出口までの避難の軌跡で、 Q は避難者の避難開始から避難完了までのすべての軌跡の集合とする。 $E(q)$ を以下のように定義する。

$$E(q) = \alpha R(q) + (1 - \alpha) \frac{\tau_q}{\tau_{\max}} \quad (11)$$

ここで、 $R(q)$ は軌跡 q の危険度で式 (10) から求める、また、 $\tau_q[s]$ は軌跡 q で避難したときに、避難完了にかかった時間。そして、 $\tau_{\max}[s]$ は避難の限界時間。また、 α は $\alpha \in [0, 1]$ の定数である。

そして、 $E(q) \in [0, 1]$ で、0 に近いほどその軌跡 q は受けた危険度が小さく、避難完了時間も短い軌跡である。

3. 誘導手法

前述の評価関数 $E(q)$ を用いて、屋内で火災が発生した場合にどのような軌道で避難者を出口まで誘導するかを考える。

本稿の想定する状況では、動的に温度や一酸化炭素の濃度が時間と場所によって変化する。よって、危険度の時間発展と避難者のすべての軌道を考えて評価関数を計算すると、計算量が增大することが考えられる。

そこで、本稿はそのように動的に変化する環境で経路を探索するアルゴリズム、Lifelong Planning A*(LPA*) アルゴリズムを今回の問題に用いることを考える。

3.1 仮定

本稿では、問題を単純にするため以下の仮定をおく。

- 環境を離散化して格子で表現する。
- 格子点をノード、格子点と格子点をつなぐ線をエッジとする。
- ノードにのみ避難者が存在する。
- ノードの場所の温度と一酸化炭素濃度だけを考慮する。
- あらかじめ、火と煙がどのように拡散するかは既知のものとする。
- 避難者は一人のみで、所持している携帯端末が表示する避難誘導の指示に従う。
- 避難者の速さは一定。

3.2 定式化

以上の仮定から、以下の条件を満たす軌道 q を求めることになる。

$$\begin{aligned} &\text{Find } q \in Q \text{ such that } \min E(q) \\ &\text{subject to} \end{aligned} \quad (12)$$

The pseudocode uses the following functions to manage the priority queue: $U.Top()$ returns a vertex with the smallest priority of all vertices in priority queue U . $U.TopKey()$ returns the smallest priority of all vertices in priority queue U . (If U is empty, then $U.TopKey()$ returns $[\infty; \infty]$.) $U.Pop()$ deletes the vertex with the smallest priority in priority queue U and returns the vertex. $U.Insert(s, k)$ inserts vertex s into priority queue U with priority k . $U.Update(s, k)$ changes the priority of vertex s in priority queue U to k . (It does nothing if the current priority of vertex s already equals k .) Finally, $U.Remove(s)$ removes vertex s from priority queue U .

```

procedure CalculateKey( $s$ )
{01} return [ $\min(g(s), rhs(s)) + h(s, s_{goal}); \min(g(s), rhs(s))$ ];

procedure Initialize()
{02}  $U = \emptyset$ ;
{03} for all  $s \in S$   $rhs(s) = g(s) = \infty$ ;
{04}  $rhs(s_{start}) = 0$ ;
{05}  $U.Insert(s_{start}, CalculateKey(s_{start}))$ ;

procedure UpdateVertex( $u$ )
{06} if ( $u \neq s_{start}$ )  $rhs(u) = \min_{s' \in Pred(u)} (g(s') + c(s', u))$ ;
{07} if ( $u \in U$ )  $U.Remove(u)$ ;
{08} if ( $g(u) \neq rhs(u)$ )  $U.Insert(u, CalculateKey(u))$ ;

procedure ComputeShortestPath()
{09} while ( $U.TopKey() < CalculateKey(s_{goal})$  OR  $rhs(s_{goal}) \neq g(s_{goal})$ )
{10}  $u = U.Pop()$ ;
{11} if ( $g(u) > rhs(u)$ )
{12}    $g(u) = rhs(u)$ ;
{13}   for all  $s \in Succ(u)$   $UpdateVertex(s)$ ;
{14} else
{15}    $g(u) = \infty$ ;
{16}   for all  $s \in Succ(u) \cup \{u\}$   $UpdateVertex(s)$ ;

procedure Main()
{17} Initialize();
{18} forever
{19}   ComputeShortestPath();
{20}   Wait for changes in edge costs;
{21}   for all directed edges  $(u, v)$  with changed edge costs
{22}     Update the edge cost  $c(u, v)$ ;
{23}     UpdateVertex( $v$ );

```

Fig.3 LPA* Algorithm Retrieved from D* Lite¹³

$$R(q) < 1. \quad (13)$$

3.3 Lifelong Planning A*

ここでは、LPA*アルゴリズムの概要について説明する。文献¹³によると、LPA*アルゴリズムはA*アルゴリズムを改良したものである。有限グラフのエッジの重みが時間で変化する場合や、ノードが増えたり減ったりする場合の、グラフ探索の問題に応用可能である¹⁴。

LPA*アルゴリズムはFig.3のような流れで実行される。以下にLPA*アルゴリズムについて、文献¹³に基づいて説明を行う。

S を対象とする有限グラフのノードの集合とする。また、 $Succ(s) \subseteq S$ はノード $s \in S$ の移動可能な次のノードの集合を表す。同様に、 $Pred(s) \subseteq S$ はノード $s \in S$ につながる移動可能だった前のノードの集合とする。 $0 < c(s, s') \leq \infty$ はノード s から $s' \in Succ(s)$ に移動するときにかかる重みを表現する。

グラフの位置関係と現在のエッジの重みが既知の場合、LPA*は開始のノード $s_{start} \in S$ から終点のノード $s_{goal} \in S$ までの最短距離を決定する。ここで、 $g^*(s)$ をノード $s \in S$ から s_{start} から s までの最短距離とする。A*と同様に、LPA*はヒューリスティクス $h(s, s_{goal})$ で $s \in S$ から s_{goal} までの最短距離を推定する。そして、 $h(s, s_{goal})$ は以下の三角不等式を満たす。

$h(s_{goal}, s_{goal}) = 0$.
そして、すべての $s \in S$ と $s' \in Succ(s)$ について、 $h(s, s_{goal}) \leq c(s, s') + h(s', s_{goal})$.
ただし、ここで $s \neq s_{goal}$.

$g(s)$ は $g^*(s)$ の推定値とする。そして、LPA*は開始

地点から現在地までの、二つ目の推定値 $rhs(s)$ を用いる。 $rhs(s)$ の定義は下記の通りである。

$$rhs(s) = \begin{cases} 0 & \text{if } s = s_{start} \\ \min_{s' \in Pred(s)} (g(s') + c(s', s)) & \text{otherwise} \end{cases} \quad (14)$$

また、LPA*は優先度付きキューを用いる。あるノード $s \in S$ について、 $g(s) \neq rhs(s)$ なら s はこのキューに追加される。このキューは $g(s)$ と関連していて、どのノードに移動すればいいかの目安に用いる。そして、そのキューは鍵と呼ばれる2つの要素を持つベクトルで表現され、 $k(s) = [k_1(s); k_2(s)]$ とする。ここで、 $k_1(s) = \min(g(s), rhs(s)) + h(s, s_{goal})$, $k_2(s) = \min(g(s), rhs(s))$ {01}。なお、{01}はFig.3の行番号を意味している。

その鍵は辞書式順序で大小関係が決定される。つまり、以下の関係を満たす。

$$\begin{aligned} k(s) &\leq k(s') \\ \Leftrightarrow k_1(s) &< k_1(s') \\ &\text{or} \\ \Leftrightarrow (k_1(s) = k_1(s') \text{ and } k_2(s) &\leq k_2(s')) \end{aligned}$$

LPA*は最小の鍵に対応するノードを次々に探索していく。これはFig.3の{10-16}に対応している。

LPA*のメイン関数が最初にInitialize()を呼び出し、この探索問題を初期化する。これはFig.3の{17}である。そして、Initialize()はすべてのノードの $g(s)$ を無限大に設定し、式(20)に従って $rhs(s')$ を設定する{03-04}。よって、 s_{start} のみが優先度付きキューに追加される{05}。この時点で、A*と同じなので、ComputeShortestPath()はA*探索と同様な働きをする。

次に、エッジの重みが変化するのを待つ{20}。そして、エッジの重みが変化した場合、その変化をUpdateVertex(){23}で更新する。その後、再度最短距離を計算し、 s_{goal} に到達するまでこれを繰り返す。もし、 $g(s_{goal}) = \infty$ ならば、そのグラフでは s_{start} から s_{goal} に到達不可能ということの意味する。

3.4 提案手法

前述のLPA*を用いて、式(12)を式(13)の条件のもとで解く手法を考案する。

LPA*はエッジの重みが変化するとき、目的地までの最短経路を求めるアルゴリズムである。また、本稿の目的は式(16)を最小化するような軌道を求めることである。そこで、エッジの重みを式(11)で表現する。ここで、避難者の移動速度 v は $v = v_{const}$ で一定とする。

ある軌道 $q \in Q$ を、時間が等間隔になるように g 個に分割する。 $1 \leq i \leq g \in \mathbb{N}$ として、軌跡 q 上において、始点を $p_{q_0}(x_0, y_0, t_0) \in q$ 、終点を $p_{q_g}(x_g, y_g, t_g) \in q$ とする。ここで、 $1 \leq i \leq g$ において点 $p_{q_{i-1}}(x_{i-1}, y_{i-1}, t_{i-1}) \in q$ から点 $p_{q_i}(x_i, y_i, t_i) \in q$ まで移動した軌跡を q_i とする。

つまり、 $q = \cup_{i=1}^g q_i$ 。ただし、すべての $i \in [1, g]$ において、 $t_i - t_{i-1} = const$ となるように分割する。ここで、 $const$ は正の実定数。そして、 Q に属する軌跡 q 上のすべての点は格子点、つまりノードとする。

ここからは3.3章の記号を使い、 $q = S$ とする。つまり、

$s = p_{q_j} \in q$ となる $j \in [0, g]$ が存在する.

避難者が q の軌跡に沿って避難するとき, 開始ノード $s_{start} = p_{q_0}(x_0, y_0, t_0)$ からノード $s_{q_i} = p_{q_i}(x_i, y_i, t_i)$ まで移動したときの軌道の評価を, 式 (11) を用いて便宜的に以下のように表現する.

$$E(q_{0,i}) = \alpha R(q_{0,i}) + (1 - \alpha) \frac{t_i - t_0}{\tau_{max}} \quad (15)$$

ここで, $q_{0,i}$ は開始ノード s_{start} からノード s_{q_i} まで移動したときの軌道. つまり, $q_{0,i} = \cup_{j=1}^i q_j$, かつ $q = q_{0,g} = \cup_{j=1}^g q_j$.

よって, ノード $p_{q_{i-1}}(x_{i-1}, y_{i-1}, t_{i-1})$ からノード $p_{q_i}(x_i, y_i, t_i)$ まで移動したときの, エッジの重み $c(p_{q_{i-1}}, p_{q_i})$ は以下ようになる.

$$\begin{aligned} c(p_{q_{i-1}}, p_{q_i}) &= E(q_{0,i}) - E(q_{0,i-1}) \\ &= \alpha(R(q_{0,i}) - R(q_{0,i-1})) + \frac{1 - \alpha}{\tau_{max}}(\Delta t_i - \Delta t_{i-1}) \\ &= \alpha(R(q_{0,i}) - R(q_{0,i-1})) + \frac{1 - \alpha}{\tau_{max}}(t_i - t_{i-1}) \end{aligned} \quad (16)$$

ここで, $\Delta t_i = t_i - t_0$, $\Delta t_{i-1} = t_{i-1} - t_0$ とする.

$R(q_{0,i}) = 1$ ならば, そのエッジを選ばないようにすれば, 式 (13) の条件を満たした軌道を探索可能である.

また, 3.3 章で $s_{start} \in S$ から $s \in S$ までの最短距離を表す関数として, $g(s)$ が用いられた. ここでは, 式 (11) を最小にするために, $s_{start} \in S$ から $s \in S$ までの危険度を $R(s)$ として, $R(s) = 0$. そして, s_{start} から s までの最短時間を Δt とすると,

$$\Delta t = \frac{d(s_{start}, s)}{v} \quad (17)$$

となる. また, $d(s_{start}, s)$ はノード s_{start} とノード s のマンハッタン距離とする. よって,

$$\begin{aligned} g(s) &= (1 - \alpha) \frac{\Delta t}{\tau_{max}} \\ &= (1 - \alpha) \frac{1}{\tau_{max}} \frac{d(s_{start}, s)}{v} \end{aligned} \quad (18)$$

同様に, $h(s, s_{goal})$ についても, 式 (11) が最小となるように, $s \in S$ から $s_{goal} \in S$ までの危険度を $R_s(s_{goal})$ として, $R_s(s_{goal}) = 0$ とする. そして,

$$h(s, s_{goal}) = (1 - \alpha) \frac{1}{\tau_{max}} \frac{d(s, s_{goal})}{v} \quad (19)$$

と定める.

これらの定式化により, LPA* をこの問題に適用可能である.

4. シミュレーション

4.1 環境設定

本稿では, 以下のように環境を設定した.

- 温度, 一酸化炭素の濃度はその各々のノードで一定.
- 温度, 一酸化炭素, そして避難者の移動方向は現在存在するノードの前後左右の4方向のみ.

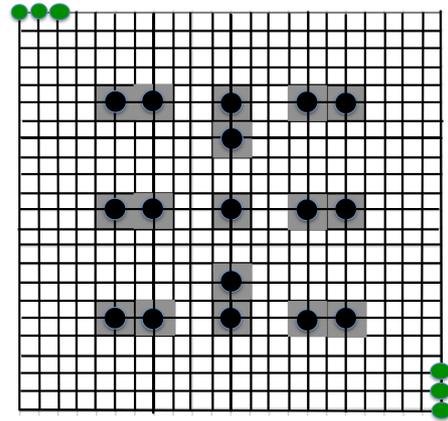


Fig.4 Map. Black circles: block nodes; Green circles; exit nodes

- 温度, 一酸化炭素, そして避難者は壁を通り抜けることは不可能.
- 1つの格子は 60cm × 60cm.
- 出口のノードは図の左上方と右下方部に Fig.4 のように設置.
- 1ステップは 0.5 秒.
- 避難者の移動速度は 1.2m/s.
- 避難者の避難は, 火元の温度が 573.15K を超えてから 5 秒後に始まる.
- 火災はランダムに 1つのノードから発生し, その地点を中心に温度と一酸化炭素が拡散する

まず, それぞれの出口に向かったときの, 式 (12) を満たす軌道を得る. その後, その中で式 (11) がもっとも小さくなる軌道を採用する. また, 文献¹⁵によると, 火元は時間の 2 乗に比例して熱放射は増加していくことから, 火元の火の成長 $H[W]$ を以下のように表現できる.

$$H = 0.0469t^2 \quad (20)$$

そして, 火元の温度は式 (20) に従って上昇していくものとする.

4.2 比較手法

提案手法の有効性を示すために, 式 (12-13) を満たす軌道だけでなく他の軌道と比較する.

$q' \in Q$ を, 可能な限り受ける危険度が小さくなるように避難させた軌道とする. そのとき, この軌道は下の式を最小化するような軌道となる.

$$\min_{q' \in Q} R(q') \quad (21)$$

また, $q'' \in Q$ を, 可能な限り受ける避難完了時間が短くなるように避難させた軌道とする. そのとき, この軌道は下の式を最小化するような軌道となる.

$$\min_{q'' \in Q} \frac{\tau_{q''}}{\tau_{max}} \quad (22)$$

$$\text{subject to } R(q') < 1$$

そして, 式 (26-27) で探索した軌道と式 (17-18) を満たす軌道を, 式 (16) と危険度, 避難完了時間の 3つの観点から比較する. このとき, $\alpha = 0.5$, $\tau_{max} = 360$ とした.

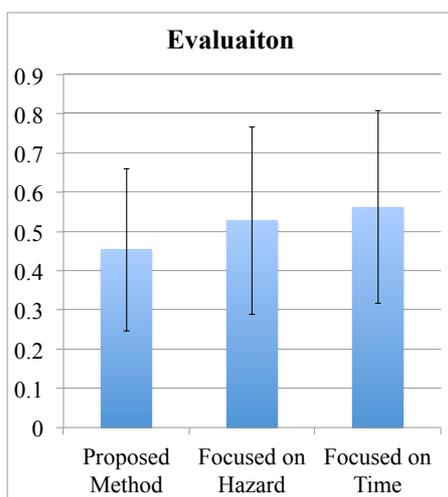


Fig.5 Result.

4.3 結果・分析

シミュレーションの結果は Fig.5 のようになった。

Proposed Method は式 (12-13) を満たす軌道の場合の、式 (11) の値。Focused on Hazard は式 (21) を満たす軌道のときの、式 (11) の値。そして、Focused on Time は式 (22) を満たす軌道のときの、式 (11) の値。Fig.5 の結果は、それぞれシミュレーションを 50 回行った結果を平均して算出したものである。

評価関数の危険度と避難時間の観点から、危険度を考慮した避難路の方が、安全性と避難時間の両面から優れていることが示された。

また、Focused on Time の結果が悪かった理由は、危険度を考慮せずに出口まで向かうため、他の軌道に比べてより多くの危険に遭遇したためだと考えられる。

5. 結言

本稿ではコンピュータシミュレーションにより、提案した手法の方がより避難者を安全で避難時間も短いように、バランスよく避難させることが可能だということを示した。

しかし、本稿では火と煙がどのように拡散するかあらかじめ把握している上で、経路を決定していたので、将来的にはそれらの拡散方向が不確定の場合を加味した手法を提案する必要がある。また、避難者が複数いる場合での状況を想定した、避難誘導路の計画の必要性が挙げられる。

参考文献

- [1] 瀧本浩一, 三浦房紀, 松元隆博, 関原弦, 組田良則, 山本真吾. "自律分散協調による避難誘導システムの開発", 社会技術研究論文集, 1349-0184 社会技術研究会, 8, pp. 82-90, 2011.
- [2] Koo, J., Kim, S. Y., Kim, B.I. "Estimating the impact of residents with disabilities on the evacuation in a high-rise building: A simulation study", Simulation Modelling Practice and Theory, 24, pp. 71-83, 2012.
- [3] Saeed Osman, M.F., Ram, B. "Evacuation route scheduling using discrete time-based Capacity-Constrained model," Proceedings of the 2011 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), pp.161-165, 2011.

- [4] Saadatseresht, M., Mansourian, A., Taleai, M. "Evacuation planning using multiobjective evolutionary optimization approach", European Journal of Operational Research, 198(1), pp. 305-314, 2009.
- [5] Guanquan, C., Jinhua, S. "Quantitative Assessment of Building Fire Risk to Life Safety", Risk Analysis, 28(3), pp. 615-625, 2008.
- [6] 新居 康彦, 渡辺 良信, 吉田 登美男. "方位コードを用いたルート決定アルゴリズムによる積極誘導ビル避難シミュレーション", シミュレーション, 17(2), pp. 134-140, 1998.
- [7] Yan-yan, C., Dong, L., Hui, Z. "Multi-factor Risk Analysis in a Building Fire by Two Step Cluster", Procedia Engineering, 11, pp. 658-665, 2011.
- [8] Koenig, S., Likhachev, M. "Incremental A*", Advances in Neural Information Processing Systems 14 (NIPS), MIT Press, Cambridge, MA, 2002.
- [9] Guogao, Z. "Biology and Health in High Temperature". ShangHai: ShangHai, Science Technology Pr, 1989
- [10] Naian, L., Qiong, L., Zhihua, Z. et al. "Burn-out time data analysis on interaction effects among multiple fires in fire arrays". P Combust Inst(2), pp. 2589-2597, 2007.
- [11] 財) 日本中毒情報センター,
[http://www.j-poison-ic.or.jp/homepage.nsf/7bf3955830f37ccf49256502001b614f/aa11e24792343d9e49256fe0000f9472/\\$FILE/一酸化炭素\(概要\).pdf](http://www.j-poison-ic.or.jp/homepage.nsf/7bf3955830f37ccf49256502001b614f/aa11e24792343d9e49256fe0000f9472/$FILE/一酸化炭素(概要).pdf).
- [12] 竹内 亨. "一酸化炭素中毒の予防について". 産業衛生学雑誌, 51(5), pp. 71-73, 2009.
- [13] Koenig, S, Likhachev, M. "D* Lite". In Proceedings of the AAAI Conference of Artificial Intelligence (AAAI), pp. 476-483, 2002.
- [14] Koenig, S, Likhachev, M and Furcy, D. "Lifelong Planning A*". Artificial Intelligence Journal, 155, (1-2), pp. 93-146, 2004.
- [15] Poon, L. "Modeling Fire Development for Fire Engineering Design", Proceedings of 4th International Conference on Performance-Based Codes and Fire Safety Design Methods, pp.264-275, 2002