4L-06

災害シミュレーションにおける目的地まで最小曲がり回数で到達する避難経路を生成するアルゴリズム

尾崎翔*，朝倉宏一 (大同大学大学院)

An Algorithm for Calculating Simple Evacuation Routes in Disaster Simulation Systems

Sho Ozaki and Koichi Asakura (Daido University)

1 はじめに

大地震が発生したとき，携帯端末同士を経由して通信を行うことができるため地震災害によって通信インフラが破壊されても通信を行えるアドホック・ネットワークによる情報共有が注目を浴びている [1]．我々は被災地におけるアドホック・ネットワークの有効性をシミュレーションを用いて検証を行うことを目標として避難者エージェントの行動アルゴリズムを開発している [2]．現在移動アルゴリズムの開発として協調活動の一つである避難誘導に着目している．避難経路として一般的に使用されるのは最短経路であるが避難誘導を行う際には道路接続状況を知らない被避難誘導者にとっては何回も右左折する複雑な経路になり得る最短経路よりも多少距離が長くても右左折する回数の少ない単純な経路の方が避難経路として適していると考えられる．本稿では，周辺を熟知している避難者が他の避難者を誘導するとき避難所までの右左折の回数を最小とするための単純な避難経路を計算するためのアルゴリズムを提案する．

2 アルゴリズム

提案アルゴリズムでは各交差点の曲がり回数の計算を行う．しかし曲がり回数を計算する際，通常のやり方では全ての交差点に対して計算対象の交差点から直進し続けると特定の曲がり回数に到達できるかという判定で曲がり回数を判断する必要があるため隣接していない複数の交差点を繰り返し参照し続けるというやり方で行わなければならない．例えば Fig.1 の図のような場合．交差点 a を目的地とし交差点の左上にある数字は目的地までの右左折の回数とすると交差点 c,d,e,g の右左折の回数は 1 回だが交差点の右左折の回数を 1 回だと確定するためには c,d,e,g から直進すると 1 より 1 小さい曲がり回数 0 回に到達することを確かめなければならない．交差点 c,g は曲がり回数 0 回の交差点が隣接した交差点であるためすぐに参照できるが交差点 d,e の場合は d の場合交差点 c,b の交差点を参照し e の場合は d,c,b と曲がり回数 0 回の交差点から離れるほど複数の交差点を何度も参照しなければならないことがわかる．

しかしこのようなやり方では複数の交差点を何度も繰り返し参照する必要があるため計算量が膨大となってしまう．このような繰り返しの参照を防ぐために隣接した交差点の曲がり回数を参照のみで曲がり回数を計算できるような前処理が必要になる．

本稿ではダイクストラ法と異なり出発地点を中心に目的地までの経路を計算するのではなく目的地を中心に逆算して全ての交差点の曲がり回数を計算する方法を行う．そのため，前処理として直進先の交差点である到達先交差点と直進するとき経由した交差点，経由交差点から直進する前の出発交差点のリストを返すデータ構造を作成する．例えば Fig.2 に示す道路ネットワークにおいて，交差点 e から交差点 b に向かうことを考える．このとき交差点 g または交差点 h から出発する場合は直進して交差点 b に到達できるが交差点 d,f から出発する場合は e で右左折する必要がある. このときベクトル<b,e>に対して交差点のリスト [g,h] を対応付けるデータを保持する. このような前処理を行うことによって隣接した交差点を参照するだけで曲がり回数を計算することが可能となる．例えば h から交差点 e を経由して直進すると交差点 b に到達できるということは交差点 h を経由して直進すると交差点 e に到達できる交差点すなわちベクトル<e,h>が返すリストの交差点は直進し続けると交差点 b にたどり着くといえる．このような連鎖的なつながりを利用すれば曲がり回数の計算を行うことが可能となる．Fig.1 の交差点 b を使って説明すると，曲がり回数 1 回の交差点を求めるには直進し続けると交差点 b の交差点に到達できる交差点を計算すればよいため最初にベクトル< b, 交差点 b に隣接した交差点 c>が返すリストの交差点を曲がり回数 1 回の交差点とする．つまりベクトル<b,c>が返すリスト [d] を曲がり回数 1 回と設定する．そこから< c,d>が返したリストの交差点 e を曲がり回数 1 回に設定<d,e>が返したリストの交差点の曲がり回数を 1 回に設定 ··· とベクトル<b,c>,<c,d>,<d,e>と連鎖的に曲がり回数の計算を繰り返すことで隣接していない交差点を参照することなく隣接した交差点を参照するのみで曲がり回数を確定することが可能であるため隣接した交差点を参照するだけで曲がり回数の計算を行うことが

Copyright ©2015 Information Processing Society of Japan.
All Rights Reserved.

できる．すべての交差点に対して先ほど述べた前処理によるデータを生成後以下の処理を行う．また，本稿では交差点の直進と曲がるについては，交差点において現在の進行方向とのなす角を計算し，なす角が一番大きい道路を「直進道路」と定義し，それ以外の道路に進む場合「曲がる」と定義する．以下で述べるアルゴリズムの処理では曲がり回数の計算を行う計算対象となる交差点を格納する StartPair とループ終了後次の曲がり回数を計算するための計算対象となる交差点を格納する NextNodes の２つのキューを用いる．

1. 曲がり回数 $n \leftarrow 0$
2. 避難所と避難所に隣接した交差点の曲がり回数を n に設定
3. StartPair.push(<避難所, 避難所に隣接した交差点>)
4. <x,y> ← StartPair.pop()
5. <x,y> が対応するリストに格納されている出発交差点に曲がり回数を n に設定（ただし曲がり回数が n 以上か *null* であれば)
6. StartPair.push(<y, 5 で曲がり回数 n に設定した出発交差点 >)
7. NextNodes.push(y)
8. StartPair が空になるまで 4〜7 を繰り返す
9. NextNodes が空なら終了
10. $n \leftarrow n+1$
11. y ←NextNodes.pop()
12. StartPair.push(<y, y と隣接した交差点 (ただし曲がり回数が n か *null*)>)
13. NextNodes が空になるまで 12〜13 を繰り返す
14. 4 へ

3 評価実験

本学周辺 $3km \times 2km$ の道路ネットワークにおいて 50 点の交差点をランダムに選択し，その交差点から避難所までの最短経路，最小回数の右左折で避難所に向かった単純な経路を計算した．実験結果を Table.1 に示す，それぞれの経路の平均距離と平均曲がり回数を示している．この結果から単純な経路による移動距離は少し長くなるが曲がり回数が大幅に下がることがわかる．

4 おわりに

本稿では避難者エージェントの移動アルゴリズムのために単純な避難経路探索アルゴリズムを提案した．最短経路に比べて約 $19m$ 分わずかに移動距離が長くなるが，曲がり回数が 9.00 回から 2.44 回と大幅に少なくなり，適切な避難経路を計算できることが分かった．

提案したアルゴリズムの計算量は交差点と隣接した交差点のペアによる処理を行うことから O(VE) となりダイクストラ法の O(E+V *Log*V) と比較すると計算量が多い．よって計算量を削減するようアルゴリズムを改良することが今後の課題である．

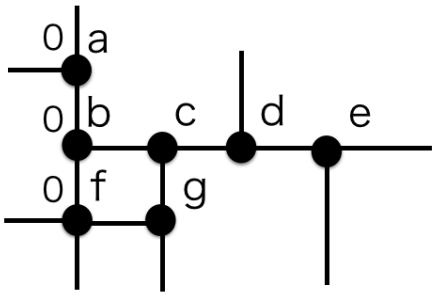

Fig. 1: 曲がり回数付き道路ネットワークの例

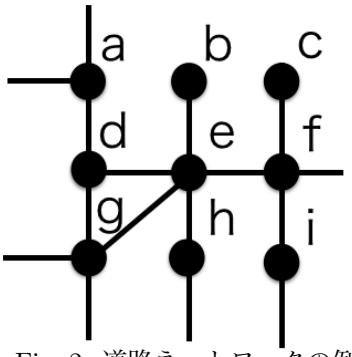

Fig. 2: 道路ネットワークの例

Table. 1: 最短と単純な経路の平均距離と曲がり回数

	平均距離 (m)	平均曲がり回数 (回)
最短経路	378.72	9.00
単純な経路	397.42	2.44

文献

[1] 間瀬 憲一：”大規模災害時の情報収集・伝達の命綱―アドホックネットワークの可能性”，電学誌，133 巻，1 号，pp.28-31 (2013).

[2] K.Asakura, H.Aoyama and T.Watanabe: “Movement Algorithms for Refugee Agents for Virtual Disaster Simulation Systems”，the 5th KESInternational Conference on Agent and Multi-Agent Systems：Technologies and Applications, pp.583-591 (2011).

Copyright ©2015 Information Processing Society of Japan.
All Rights Reserved.