

A*による迷路の最短経路探索

2019 年 12 月 18 日

とうぐー(@twu_go)

1. 問題設定

次のような迷路の最短経路を求める問題を考える。迷路にはところどころ

泥などがあり、場所によって通りやすかったり通りにくかったりする。

[illegible]

マスに書かれている文字について、S はスタート、G はゴール、-1 は壁、正の数字はその場所を通る際のコストをそれぞれ表す。

今回は、縦方向 1 マスか横方向 1 マスにしか進めず、斜め移動はできないものとした。

迷路は次のような CSV ファイルの形式で与えられる。

(ステージの横幅 W)、(ステージの縦幅 H)

(スタートの x 座標)、(スタートの y 座標)

(ゴールの x 座標)、(ゴールの y 座標)

(ステージ情報 (壁とコスト) (H×W))

2. 実装

x 座標と y 座標をひとまとめに扱うための構造体を作成した。

CSV ファイルを読み込み、ステージ情報は二次元配列に格納した。

探索済み情報は、探索にかかる実コストを格納する二次元配列と、一つ前の場所を保持する二次元配列に分割した。

結果の表示について、壁は'@'、スタートは'S'、ゴールは'G'、経路は'+'の文字で表示している。

図 1 マンハッタン距離で実行した場合

```
Ubuntu
(1, 1), (1, 2), (1, 3), (1, 4), (2, 4), (3, 4), (1, 5), (4, 4), (1, 6),
(3, 3), (4, 3), (1, 7), (3, 2), (1, 8), (3, 1), (4, 1), (4, 2), (5, 1),
(6, 1), (1, 9), (2, 9), (3, 9), (4, 9), (5, 9), (6, 9), (7, 9), (7, 1),
(8, 1), (5, 2), (6, 8), (7, 8), (8, 8), (6, 2), (1, 10), (9, 1), (6,
3), (9, 2), (6, 4), (9, 8), (7, 2), (7, 4), (7, 5), (10, 1), (2, 10),
(7, 6), (3, 10), (4, 10), (5, 10), (6, 10), (10, 2), (7, 10), (10, 3),
(10, 8), (10, 4), (10, 9), (10, 5), (10, 10), goal
rooptime = 57
cost = 24
@@@@@@@@@@@@@@@@
@S@               @
@+@               @
@+@  @ @@@@ @
@+  @             @
@+@@@@@           @
@+@             @
@+@  @  @@@@
@+@@@@@ +++++@
@+++++++  +@
@          G@
@@@@@@@@@@@@@@@@
```

図 2 ユークリッド距離で実行した場合

今回の問題設定から外れるが、8 近傍距離で実装した場合、ユークリッド距離のループ回数の方がマンハッタン距離のループ回数より少なくなった。また、マンハッタン距離では最短経路は求められなかった。

```
Ubuntu
(1, 1), (1, 2), (1, 3), (2, 4), (3, 4), (4, 4), (1, 4), (4, 3), (3, 3),
(1, 5), (1, 6), (1, 7), (3, 2), (1, 8), (2, 9), (3, 9), (4, 9), (5, 9),
(6, 9), (7, 9), (5, 2), (6, 3), (7, 4), (8, 5), (9, 6), (10, 6), (8,
6), (10, 5), (9, 5), (7, 5), (8, 4), (7, 7), (9, 4), (7, 6), (6, 4), (1
0, 4), (10, 3), (6, 6), (8, 8), (3, 10), (9, 8), (10, 9), (10, 10), goa
|
rooptime = 43
cost = 20
@@@@@@@@@@@@@@@@
@S@               @
@+@               @
@+@  @ @@@@ @
@ +  @           @
@+@@@@          @
@+@             @
@+@  @  @@@@
@+@@@@  ++ @
@ ++++++  +@
@          G@
@@@@@@@@@@@@@@@@
```

図 3 マンハッタン距離（8近傍）

```
Ubuntu
(1, 1), (1, 2), (1, 3), (2, 4), (3, 4), (4, 4), (1, 4), (3, 3), (4, 3),
(1, 5), (3, 2), (1, 6), (4, 1), (4, 2), (5, 1), (3, 1), (5, 2), (6, 3),
(7, 4), (8, 5), (9, 6), (8, 6), (7, 5), (7, 6), (10, 6), (9, 5), (6,
4), (7, 7), (8, 8), (9, 8), (10, 9), (10, 10), goal
rooptime = 32
cost = 17
@@@@@@@@@@@@@@@@
@S@               @
@+@  +           @
@+@ +@@@@@ @
@ ++ @ +       @
@ @@@@  +      @
@ @      +      @
@ @  @ + @@@@
@ @@@@  ++ @
@          +@
@          G@
@@@@@@@@@@@@@@@@
```

図 4 ユークリッド距離（8近傍）

4. 感想

コストを与えた場合でも、マンハッタン距離やユークリッド距離のヒューリスティック関数を使用して適切な経路を求めることができた。だがまだ妥当性の証明を行っていないので、妥当性を確かめたい。