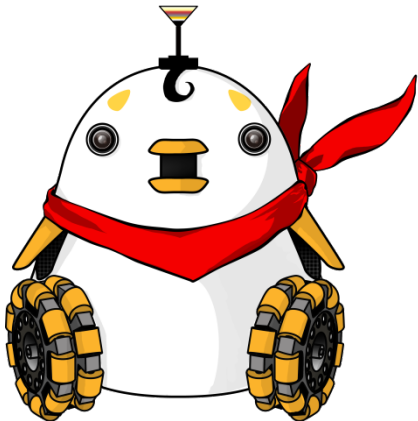


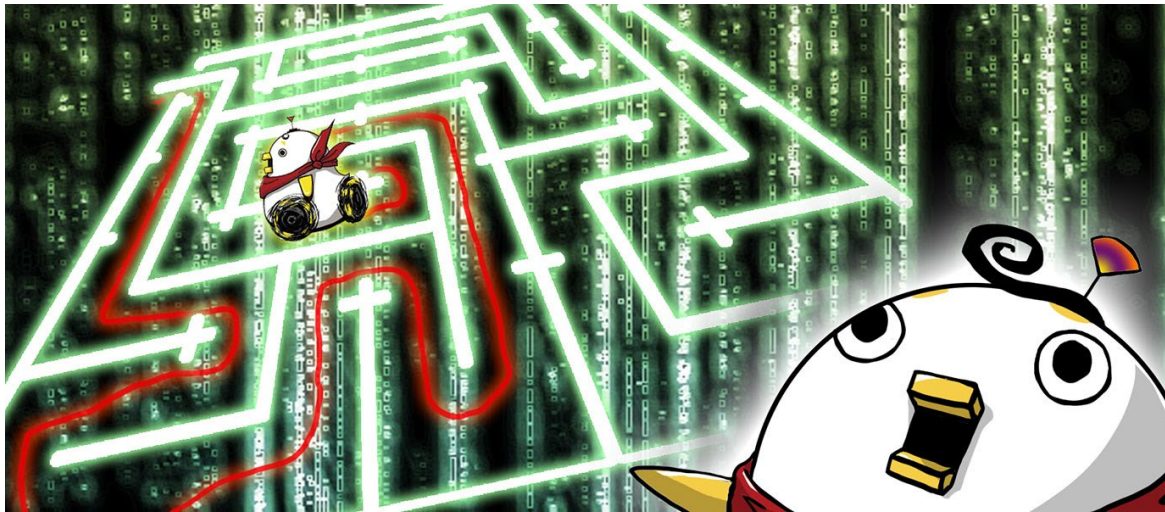
人工知能

第3章 探索(2)： 最適経路の探索



STORY 最適経路の探索

- ホイールダック2号は気づいた。深さ優先探索や幅優先探索ではゴールを見つけるまでにしばしば無駄に長い距離を移動しなければならないことに。ゴールがどこにあるのかわからないときは仕方ないが、ゴールの位置がわかっているときもある。わかっている場合にしても、できるだけ短い距離を移動してゴールに到達したい。しらみつぶしで探索するのではなく、効率的にゴールに向かう方法はないものだろうか、ホイールダック2号は思った。
- できるだけ短い時間、もしくは、短い距離でゴールにたどり着く経路を見つけたい。そのような最適経路をホイールダック2号はどうやって見つけることができるのだろうか。



仮定 最適経路の探索

- ホイールダック 2 号は迷路の完全な地図を持っているものとする.
- ホイールダック 2 号は迷路の中で自分がどこにいるか認識できるものとする.
- ホイールダック 2 号は連続的な迷路の空間から適切な離散状態空間を構成できるものとする.
- ホイールダック 2 号は各状態間の移動にかかるコストと状態の予測評価値の両方もしくはどちらかを知っているものとする.
- ホイールダック 2 号は物理的につながっている場所・状態には意図すれば確定的に移動することができるものとする.

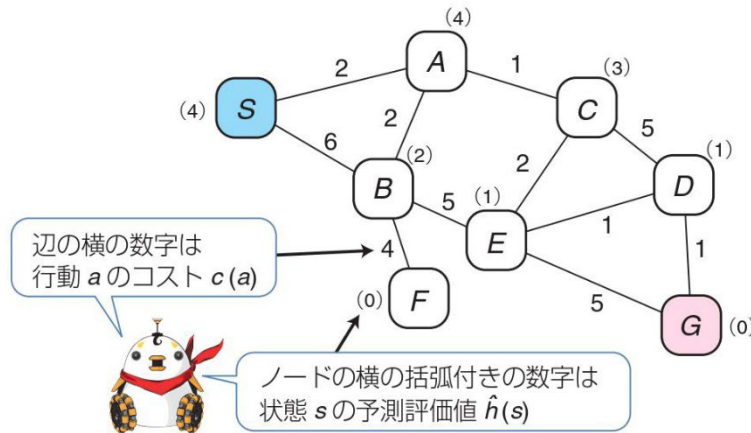


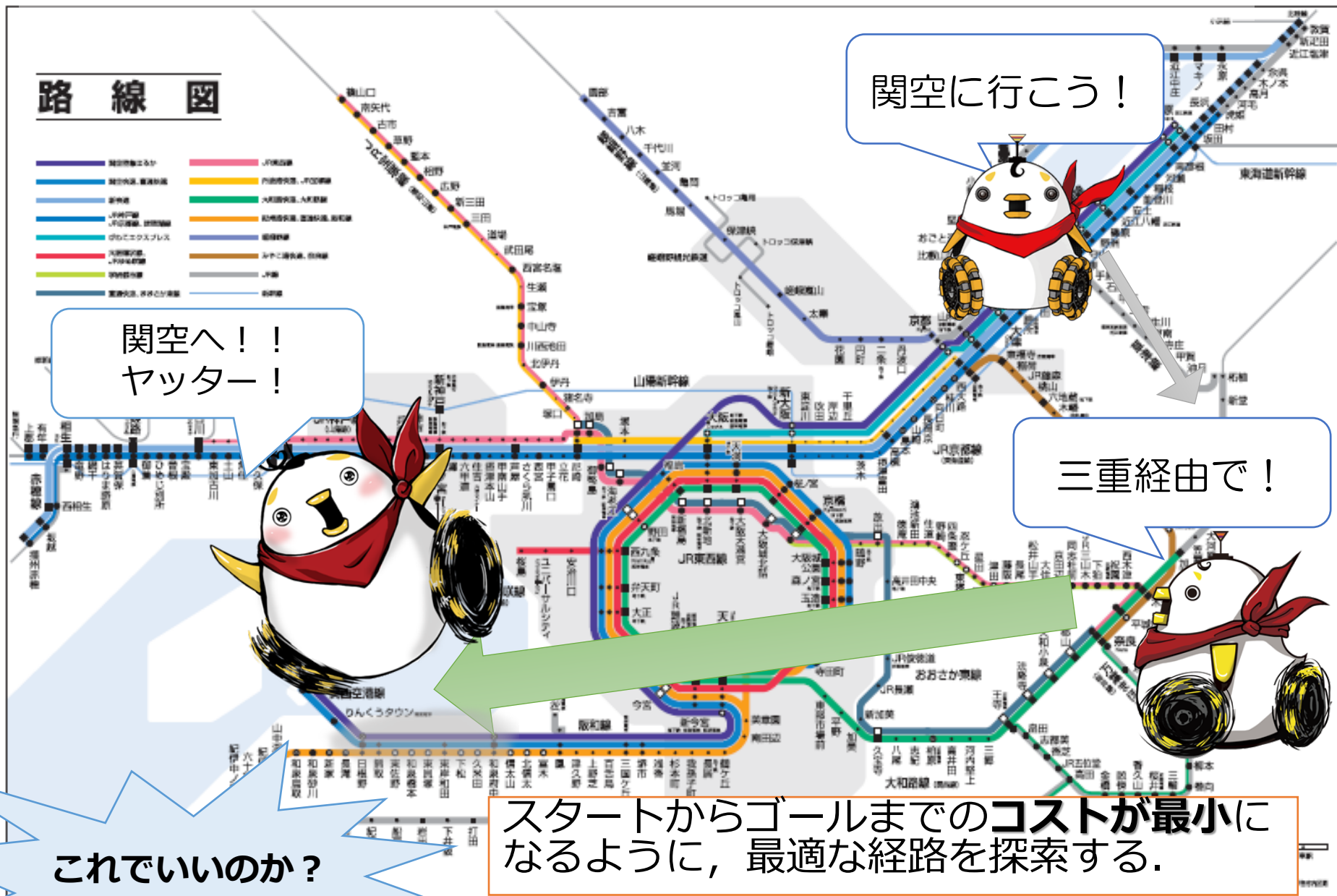
図 3.2 コストと予測評価値の記されたグラフ

Contents

- 3.1 最適経路の探索とヒューリスティックな知識
- 3.2 最適探索
- 3.3 最良優先探索
- 3.4 A*アルゴリズム
- 3.5 迷路を最適経路で抜けるホイールダック2号

経路探索問題

引用元：<https://www.jr-odekake.net/eki/pdf/ubn.pdf>



3.1.1 経路のコスト

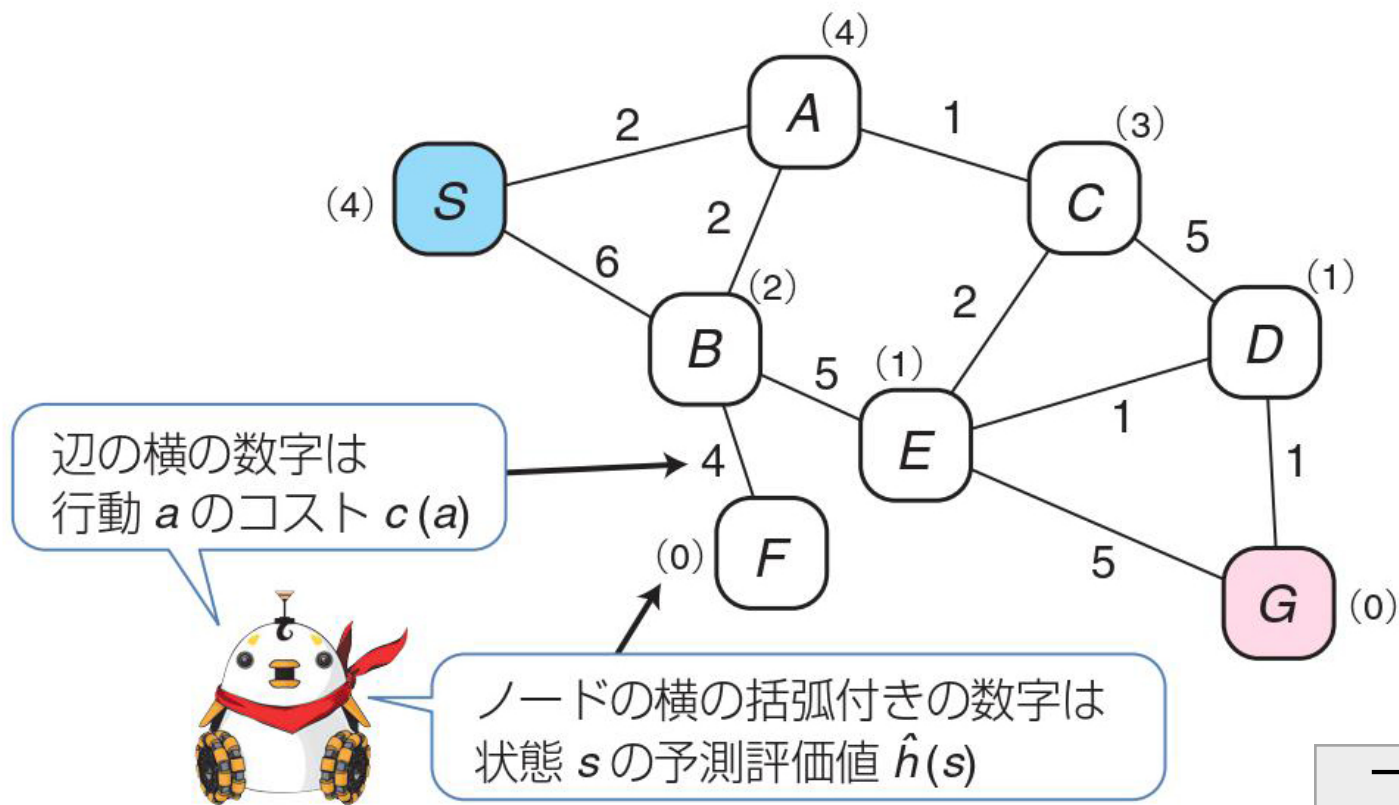


図 3.2

コストと予測評価値の記されたグラフ

コストの和を
最小化

3.1.2 ヒューリスティックな知識としての予測評価値

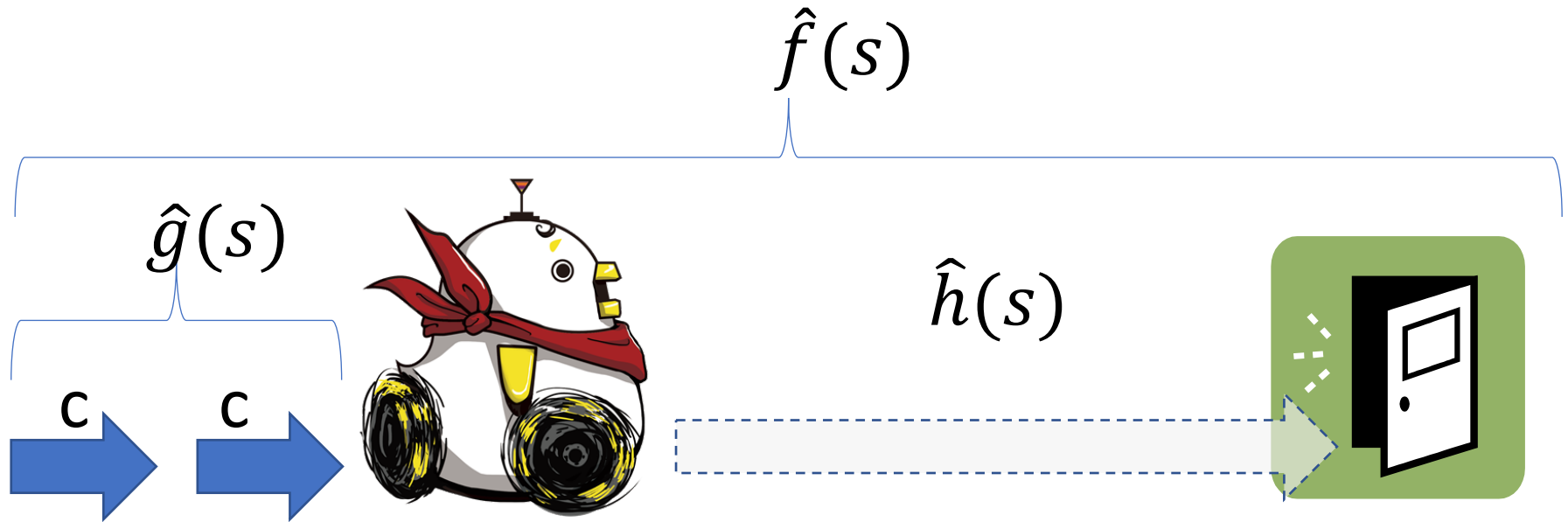


表 3.1 最適経路探索に用いる関数の定義

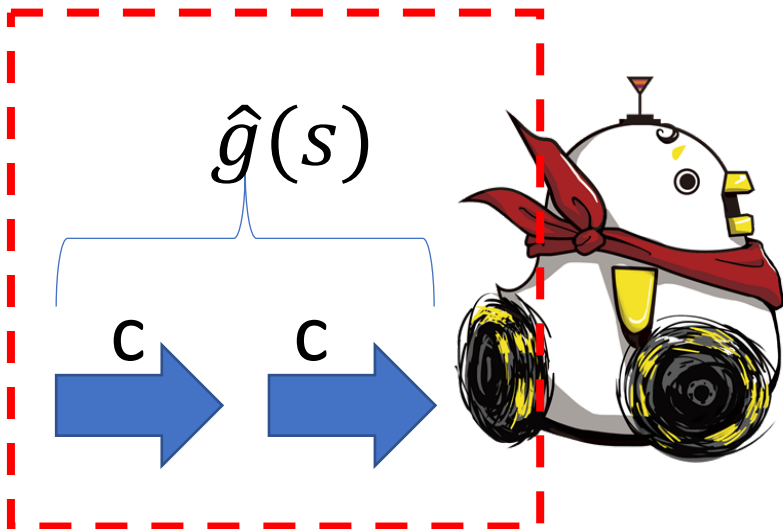
関数記号	説明
$g(s)$	初期状態から状態 s までの最適経路上のコストの総和
$h(s)$	状態 s からゴールまでの最適経路上のコストの総和
$f(s)$	s を経由した場合の最適経路のコスト ($f(s) = h(s) + g(s)$)
$\hat{h}(s)$	$h(s)$ の推定値, 状態 s の予測評価値として用いる
$\hat{g}(s)$	$g(s)$ の推定値
$\hat{f}(s)$	$f(s)$ の推定値 ($\hat{f}(s) = \hat{h}(s) + \hat{g}(s)$)

Contents

- 3.1 最適経路の探索とヒューリスティックな知識
- 3.2 最適探索
- 3.3 最良優先探索
- 3.4 A*アルゴリズム
- 3.5 迷路を最適経路で抜けるホイールダック2号

3.2 最適探索

- ヒューリスティックな知識（予測評価値）を用いず，コストの和を最小にする最適経路を確実に発見するための手法.



ここをモニタリングする！

※この手法は**ダイクストラ法**とも呼ばれる.

3.2 最適探索（アルゴリズム）

Algorithm 3.1 最適探索

- ① 初期状態のコスト値を 0 としてオープンリストに追加する. クローズドリストを空に初期化する.
- ② while オープンリストが空ではない. do
- ③ オープンリストから先頭の要素 s を取り出す. クローズドリストに s を追加する (s を探索することに相当).
- ④ s が目標状態ならば, 解は発見されたとして探索を終了.
- ⑤ s から接続していてまだ探索していない状態をすべてオープンリストに追加する. オープンリスト内の状態の累積コストの推定値 $\hat{g}(s)$ を再計算し, **累積コストの推定値が小さい順に並べ替える.**
- ⑥ end while 探索を終了.

3.2.2 最適探索の実行例

実行してみよう！

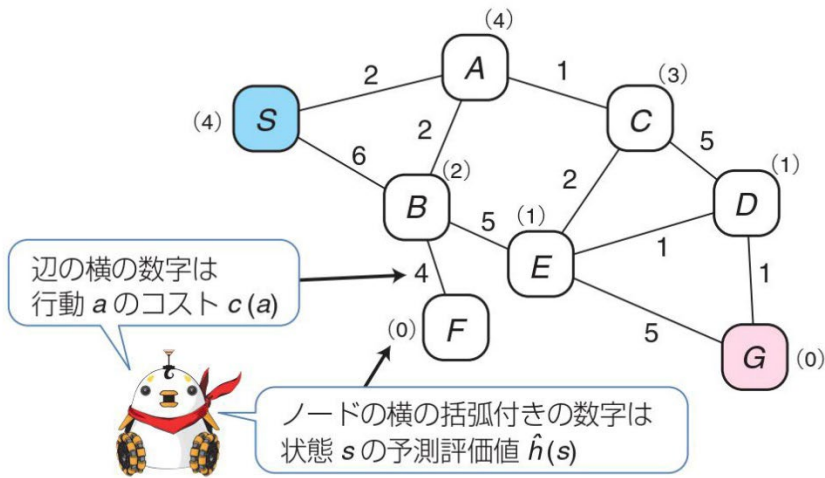


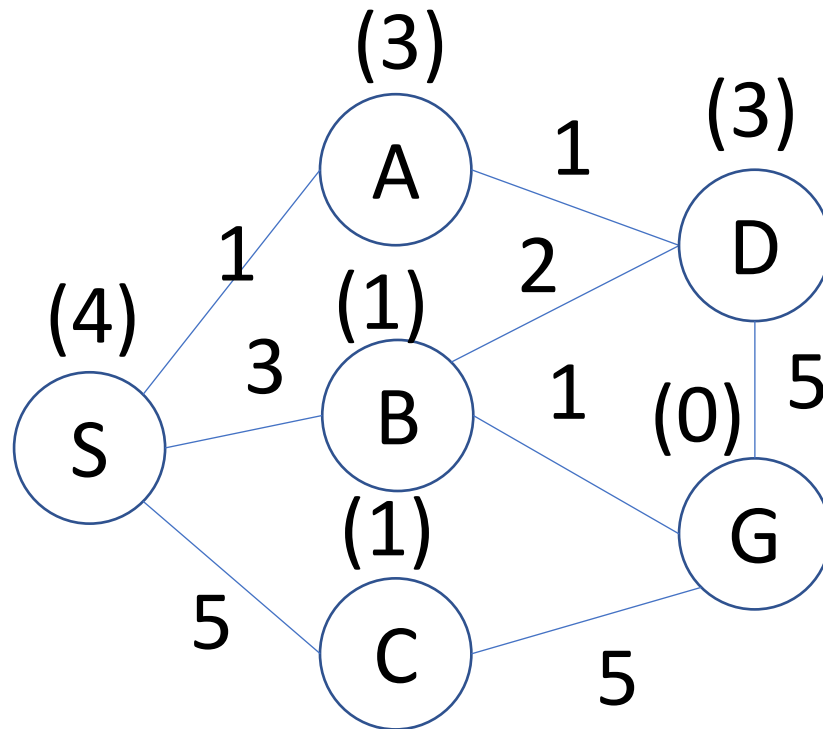
図 3.2 コストと予測評価値の記されたグラフ

表 3.2 最適探索におけるオープンリストとクローズドリストの変化

ステップ	オープンリスト	クローズドリスト
1	$S(0)$	
2	$A(2), B(6)$	$S(0)$
3	$C(3), B(4)$	$S(0), A(2)$
4	$B(4), E(5), D(8)$	$S(0), A(2), C(3)$
5	$E(5), D(8), F(8)$	$S(0), A(2), C(3), B(4)$
6	$D(6), F(8), G(10)$	$S(0), A(2), C(3), B(4), E(5)$
7	$G(7), F(8)$	$S(0), A(2), C(3), B(4), E(5), D(6)$
8	$F(8)$	$S(0), A(2), C(3), B(4), E(5), D(6), G(7)$ (解は発見され終了)

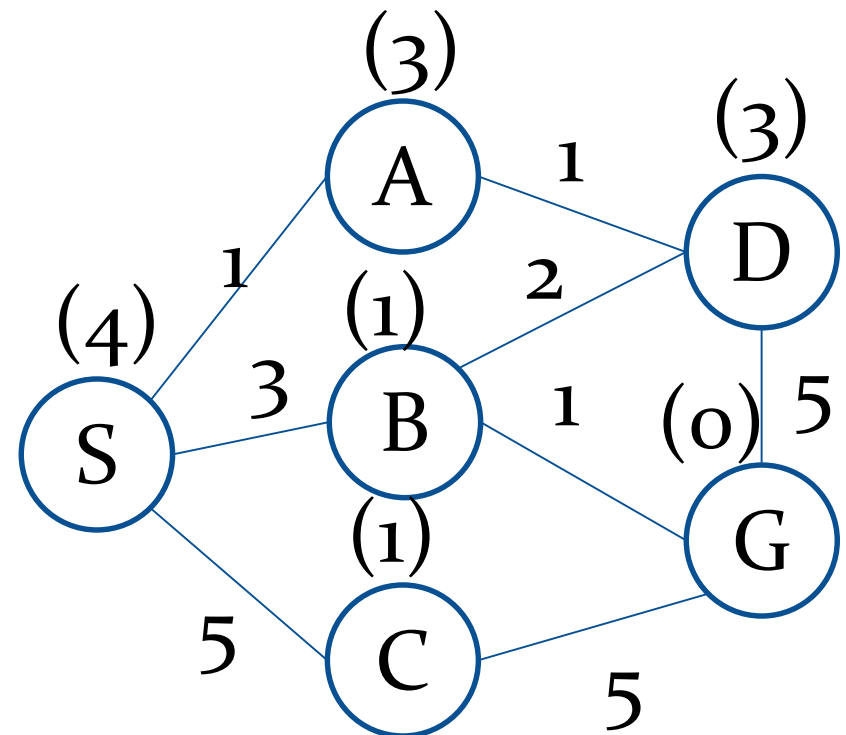
演習3-1 最適探索

- 下のグラフにおいてsからスタートして最適探索で探索せよ。探索中の様子をオープンリストで示し、最終的に得られる経路を示せ。



演習3-1 最適探索 解答

- S(0) |
- A(1) B(3) C(5) | S(0)
- D(2) B(3) C(5) | S(0) A(1)
- B(3) C(5) G(7) | S(0) A(1) D(2)
- G(4) C(5) | S(0) A(1) D(2) B(3)
- C(5) | S(0) A(1) D(2) B(3) G(4)

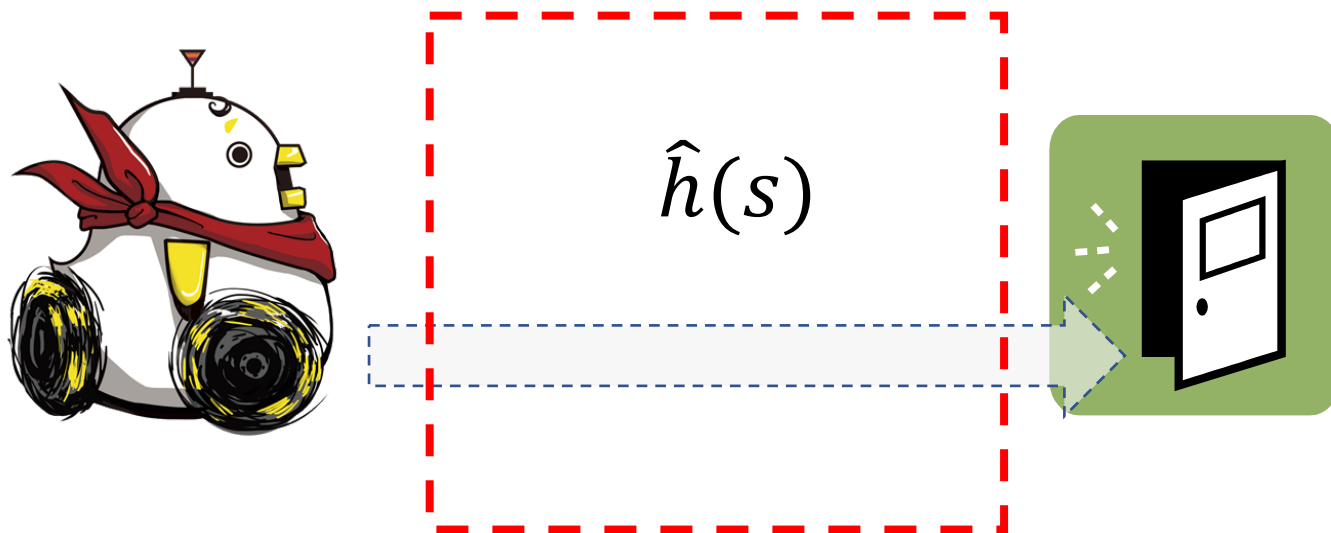


Contents

- 3.1 最適経路の探索とヒューリスティックな知識
- 3.2 最適探索
- 3.3 最良優先探索
- 3.4 A*アルゴリズム
- 3.5 迷路を最適経路で抜けるホイールダック2号

3.3.1 最良優先探索のアルゴリズム

- ヒューリスティックな知識としての予測評価値を頼りに探索を進めるのが**最良優先探索**(best-first search) である。



ここをモニタリングする！

3.3.1 最良優先探索のアルゴリズム

Algorithm 3.2 最良優先探索

- ① 初期状態を予測評価値とともにオープンリストに追加する. クローズドリストを空に初期化する.
- ② while オープンリストが空ではない. do
- ③ オープンリストから先頭の要素 s を取り出す. クローズドリストに s を追加する (s を探索することに相当).
- ④ s が目標状態ならば, 解は発見されたとして探索を終了.
- ⑤ s から接続していてまだ探索していない状態をすべてオープンリストに追加する. オープンリスト内の状態を, 予測評価値 $\hat{h}(s)$ が小さい順に並べ替える.
- ⑥ end while 探索を終了.

3.3.2 最良優先探索の実行例

実行してみよう！

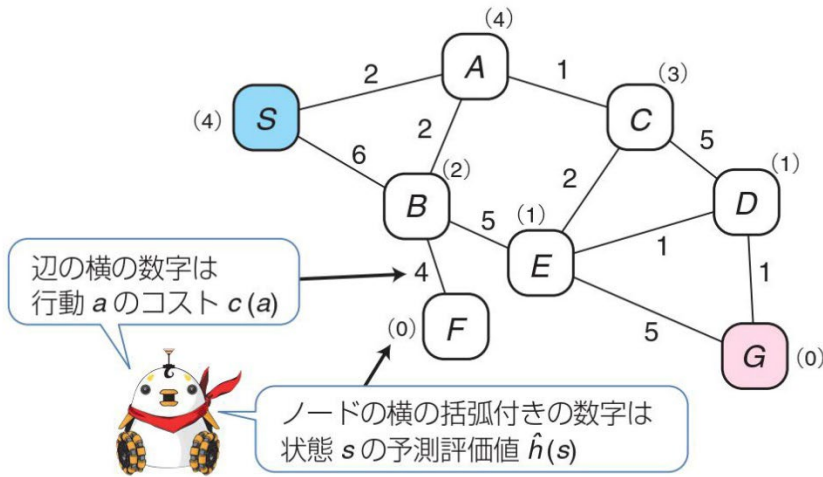


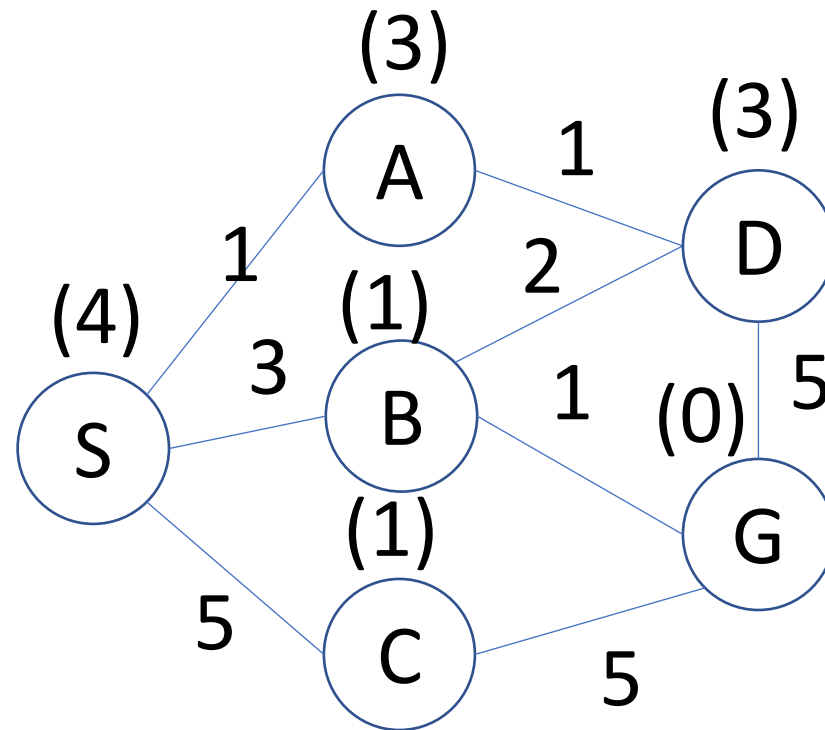
図 3.2 コストと予測評価値の記されたグラフ

表 3.3 最良優先探索におけるオープンリストとクローズドリストの変化

ステップ	オープンリスト	クローズドリスト
1	$S(4)$	
2	$B(2), A(4)$	$S(4)$
3	$F(0), E(1), A(4)$	$S(4), B(2)$
4	$E(1), A(4)$	$S(4), B(2), F(0)$
5	$G(0), D(1), C(3), A(4)$	$S(4), B(2), F(0), E(1)$
6	$D(1), C(3), A(4)$	$S(4), B(2), F(0), E(1), G(0)$ (解は発見され終了)

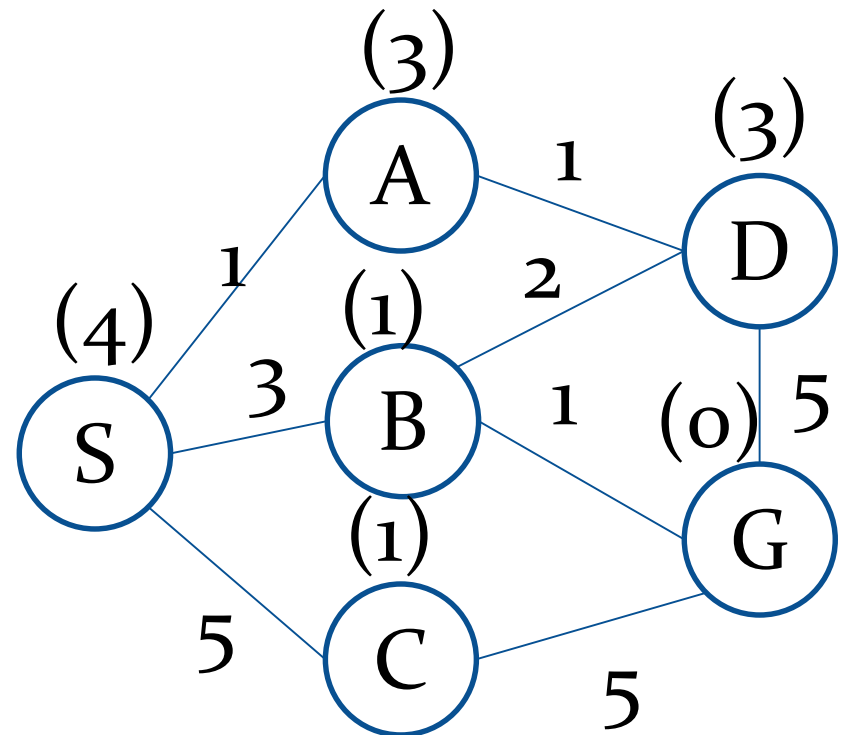
演習3-2 最良優先探索

- 下のグラフにおいてsからスタートして最良優先探索で探索せよ。
探索中の様子をオープンリストで示し、最終的に得られる経路を示せ。



演習3-2 最良優先探索 解答

- S(4) |
- B(1) C(1) A(3) | S(4)
- G(0) C(1) A(3) D(3) | S(4) B(1)
- C(1) A(3) D(3) | S(4) B(1) G(0)



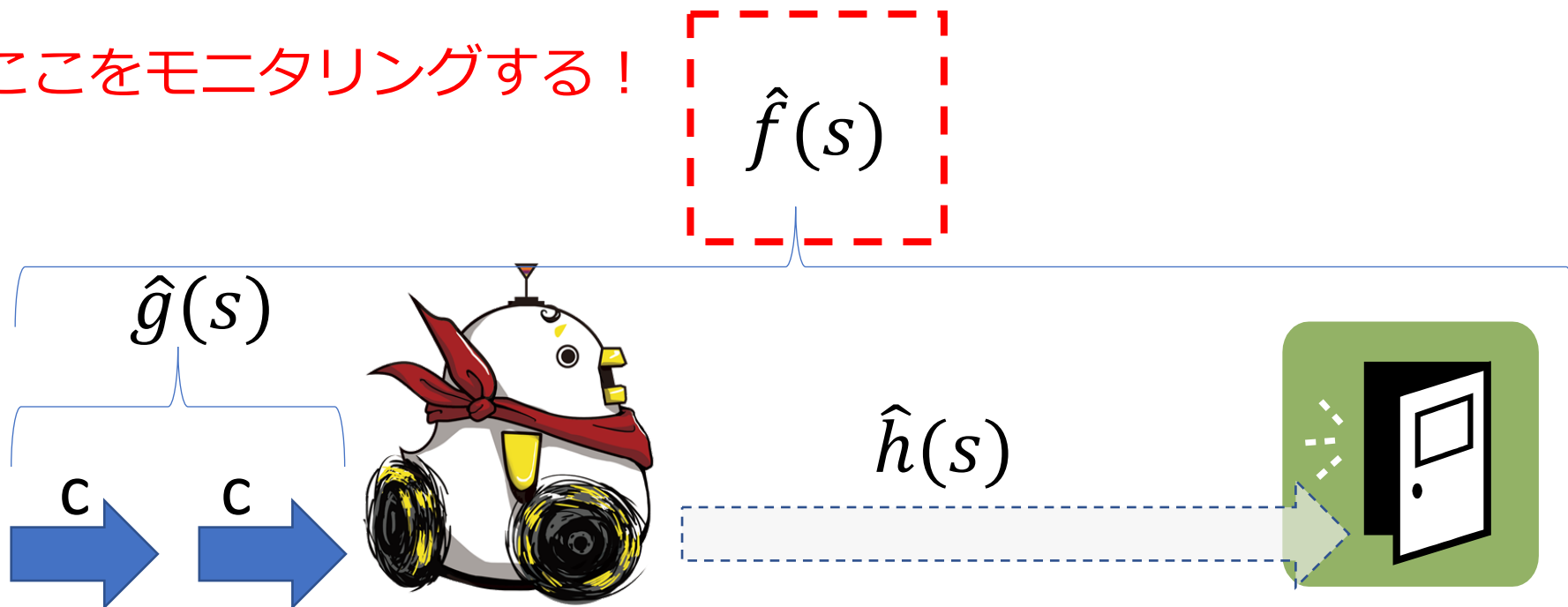
Contents

- 3.1 最適経路の探索とヒューリスティックな知識
- 3.2 最適探索
- 3.3 最良優先探索
- 3.4 A*アルゴリズム
- 3.5 迷路を最適経路で抜けるホイールダック2号

3.4.1 A*アルゴリズム

- 現在の状態までにかかったコスト $\hat{g}(s)$ と, ゴールまでに将来かかるであろう予測評価値 $\hat{h}(s)$ の二つをバランスよく用いて, 探索を効率化する手法

ここをモニタリングする！



3.4.1 A*アルゴリズム

Algorithm 3.3 A* アルゴリズム

- ① 初期状態を予測評価値とともにオープンリストに追加する。クローズドリストを空に初期化する。
- ② while オープンリストが空ではない. do
- ③ オープンリストから先頭の要素 s を取り出す。クローズドリストに s を追加する (s を探索することに相当)。
- ④ s が目標状態ならば、解は発見されたとして探索を終了。
- ⑤ s から接続されているすべての状態 s' に関して累積コストの推定値 $\hat{g}(s')$ と予測評価値 $\hat{h}(s')$ から $\hat{f}(s')$ を計算する。
- ⑥ ⑤ の状態のうちで、オープンリストにもクローズドリストにも含まれていないものは、オープンリストに加える。
- ⑦ ⑤ の状態のうちで、オープンリストかクローズドリストに含まれていたものについては、すでに入っているものより $\hat{f}(s)$ の値が小さければ、**元のものを消去し、新しいものをオープンリストに追加する。**
- ⑧ オープンリスト内の状態を、**累積コストの推定値と予測評価値の和 $\hat{f}(s)$ が小さい順に並べ替える。**
- ⑥ end while 探索を終了。

3.4.2 A*アルゴリズムの実行例

実行してみよう！

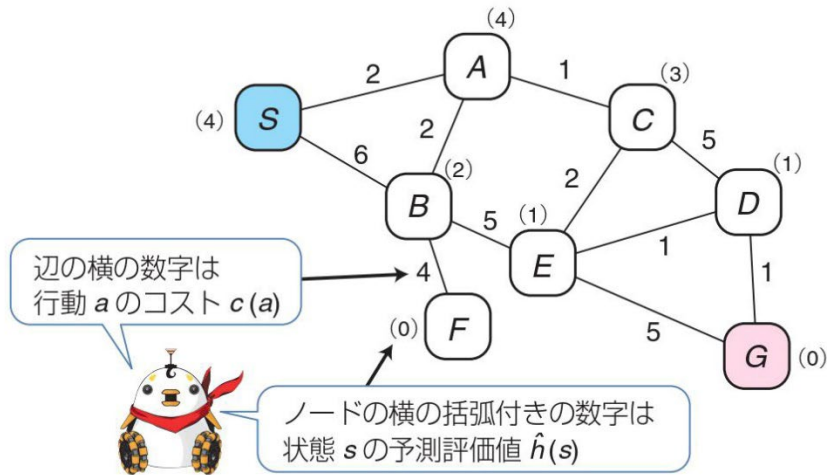


図 3.2 コストと予測評価値の記されたグラフ

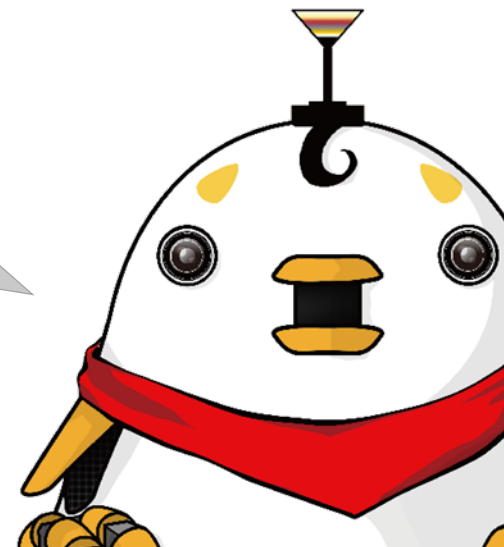
表 3.4 A* アルゴリズムにおけるオープンリストとクローズドリストの変化

ステップ	オープンリスト	クローズドリスト
1	$S(4)$	
2	$A(6), B(8)$	$S(4)$
3	$B(6), C(6)$	$S(4), A(6)$
4	$C(6), F(8), E(10)$	$S(4), A(6), B(6)$
5	$E(6), F(8), D(9)$	$S(4), A(6), B(6), C(6)$
6	$D(7), F(8), G(10)$	$S(4), A(6), B(6), C(6), E(6)$
7	$G(7), F(8)$	$S(4), A(6), B(6), C(6), E(6), D(7)$
8	$F(8)$	$S(4), A(6), B(6), C(6), E(6), D(7), G(7)$ (解は発見され終了)

3.4.3 A*アルゴリズムにおける 最適解の保証

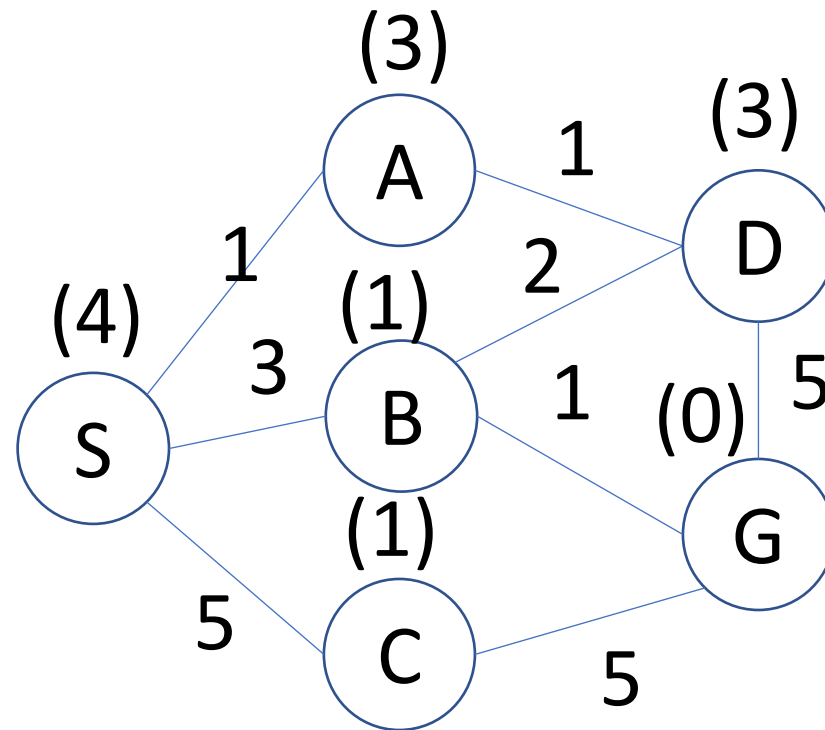
- A*アルゴリズムでは $\hat{h}(s) \leq h(s)$ の関係が成り立つときには、最適解が必ず得られることが保証されている。
- 逆に $\hat{h}(s) > h(s)$ の場合は最適解が得られることが保証されない。
- ゆえに、適切な予測評価値がわからない場合については、推定値は小さめに設定する方がよいと考えられる。

安心!!!!



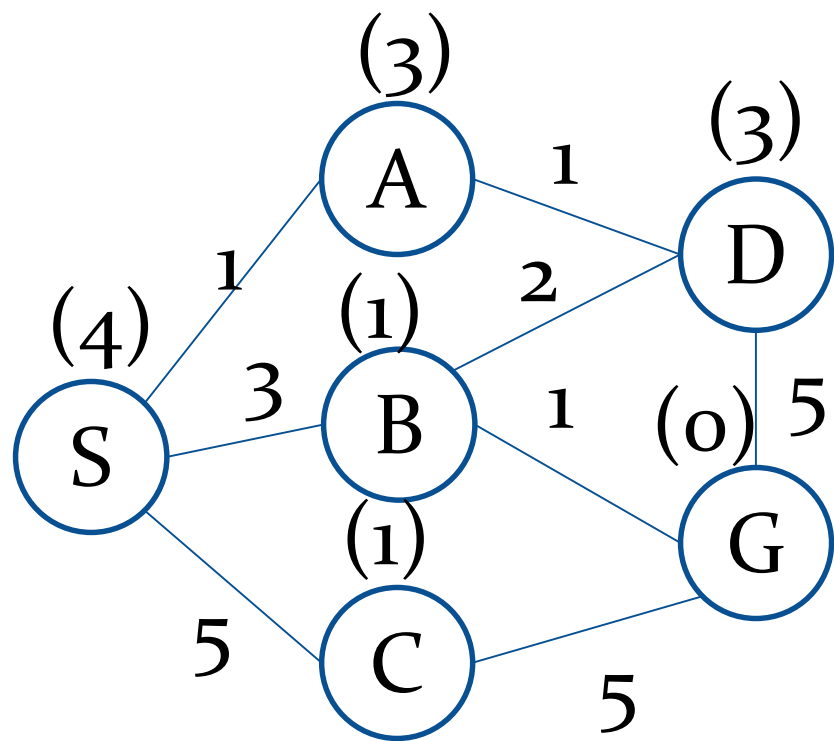
演習3-3 A*アルゴリズム

- 下のグラフにおいてsからスタートしてA*アルゴリズムで探索せよ。探索中の様子をオープンリストで示し、最終的に得られる経路を示せ。



演習3-3 A*アルゴリズム 解答

- S(4) |
- A(4) B(4) C(6) | S(4)
- B(4) D(5) C(6) | S(4) A(4)
- G(4) D(5) C(6) | S(4) A(4) B(4)
- D(5) C(6) | S(4) A(4) B(4) G(4)



Contents

- 3.1 最適経路の探索とヒューリスティックな知識
- 3.2 最適探索
- 3.3 最良優先探索
- 3.4 A*アルゴリズム
- 3.5 迷路を最適経路で抜けるホイールダック2号

3.5 迷路を最適経路で抜ける ホイールダック2号

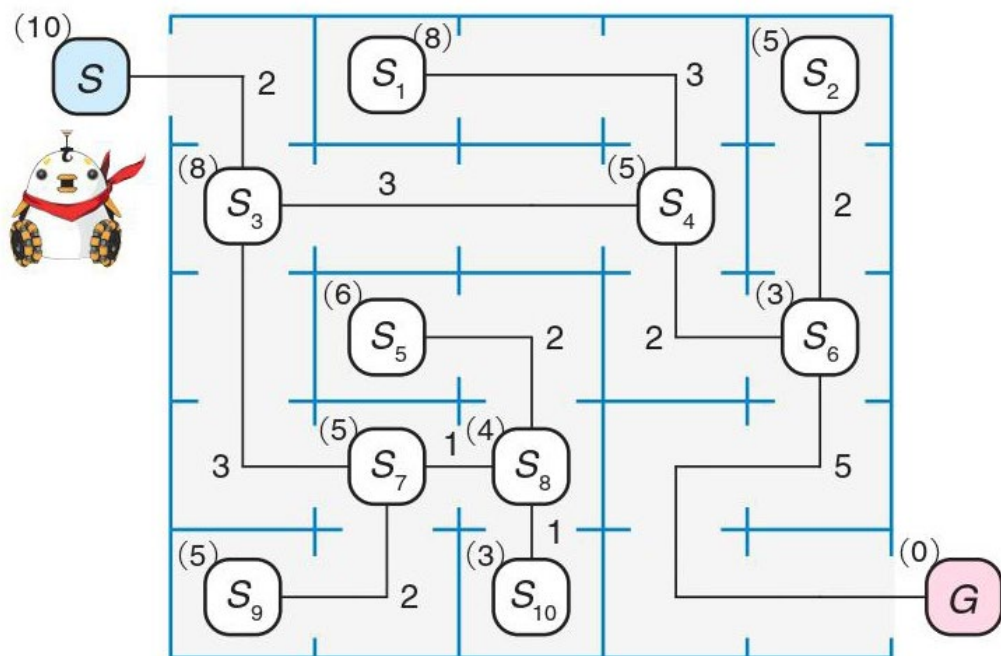


図 3.3 コスト付きのグラフ

- 辺にコスト $c(a)$ を置く
- 予測評価値 $\hat{h}(s)$ はゴールまでの壁を無視した最短距離
- A* アルゴリズムが最適解を出す条件を満たす.

OK

※一般的に上記のように探索のための「制約」を取り外すことで妥当な予測評価値 $\hat{h}(s)$ を作成することが出来る.

解いてみよう

ホイールダック 2 号による実行結果

表 3.5 ホイールダック 2 号による A* アルゴリズム

ステップ	オープンリスト	クローズドリスト
1	$S(10)$	
2	$S_3(10)$	$S(10)$
3	$S_4(10), S_7(10)$	$S(10), S_3(10)$
4	$S_7(10), S_6(10), S_1(16)$	$S(10), S_3(10), S_4(10)$
5	$S_6(10), S_8(10), S_9(12), S_1(16)$	$S(10), S_3(10), S_4(10), S_7(10)$
6	$S_8(10), G(12), S_9(12), S_2(14), S_1(16)$	$S(10), S_3(10), S_4(10), S_7(10), S_6(10)$
7	$S_{10}(10), G(12), S_9(12), S_2(14), S_5(14), S_1(16)$	$S(10), S_3(10), S_4(10), S_7(10), S_6(10), S_8(10)$
8	$G(12), S_9(12), S_2(14), S_5(14), S_1(16)$	$S(10), S_3(10), S_4(10), S_7(10), S_6(10), S_8(10), S_{10}(10)$
9	$S_9(12), S_2(14), S_5(14), S_1(16)$	$S(10), S_3(10), S_4(10), S_7(10), S_6(10), S_8(10), S_{10}(10), G(12)$ (解は発見され終了)

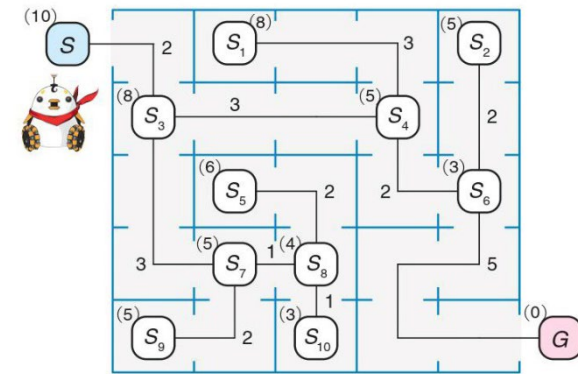


図 3.3 コスト付きのグラフ

それぞれに確認しておくこと

最適探索と最良探索, A*

- 最適探索
 - 経路のコストは明確にわかっている.
 - ノードの推定コストはわからない.
 - 最適な経路が必ず求められる.
- 最良優先探索
 - 経路のコストがわからない (事後的にしかわからない) .
 - ノードの推定コストがわかる.
 - 最適な経路が求められる保証はない.
- A*探索
 - 経路のコストがわかっている.
 - ノードの推定コストはわかっている.
 - 最適な経路は一定の条件の下で保証される.

	経路 コスト 情報	ノード 推定 コスト	最適性の 保証
最適 探索	○	×	○
最良 優先 探索	×	○	×
A*探索	○	○	△

演習3-4

- 深さ優先探索，幅優先探索，最適探索，最良優先探索，A*アルゴリズムはオープンリストにおける状態の管理手法の違いによって特徴付けられる．これらのオープンリストにおける状態の管理手法の違いを説明せよ．

演習3-4 解答

- 深さ優先探索 スタック
- 幅優先探索 キュー
- 最適探索
 - スタートからの最適経路の推定コスト和でソート
- 最良優先探索
 - ゴールまでの最適経路の推定コスト和でソート
- A*アルゴリズム
 - その点を通る最適経路の推定コスト和でソート

第3章のまとめ

- グラフに経路のコストを付与し最適経路探索問題の基礎について学んだ.
- 最適探索のアルゴリズムについて学んだ.
- 最良優先探索のアルゴリズムについて学んだ.
- A*アルゴリズムについて学び, アルゴリズムを実行した.