

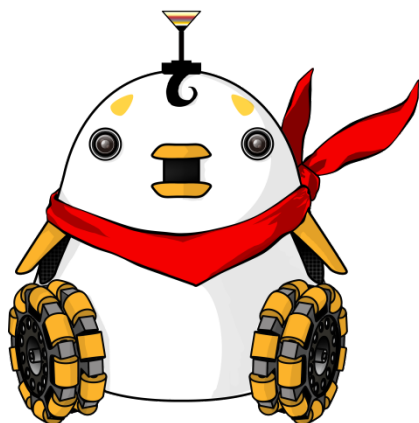
人工知能

第3章 探索(2)

最適経路の探索

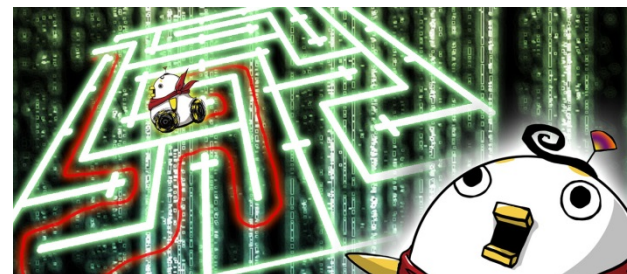
立命館大学 情報理工学部 知能情報学科

萩原良信



STORY 探索(2)

- 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**. わかっていない場合にしても, **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ってみようと, 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. どうやってそのような経路を見つけることができるのだろう



仮定 探索(2)

- 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評価推定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.

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

經路探索問題

路線圖



関空へ！！
ヤッター！

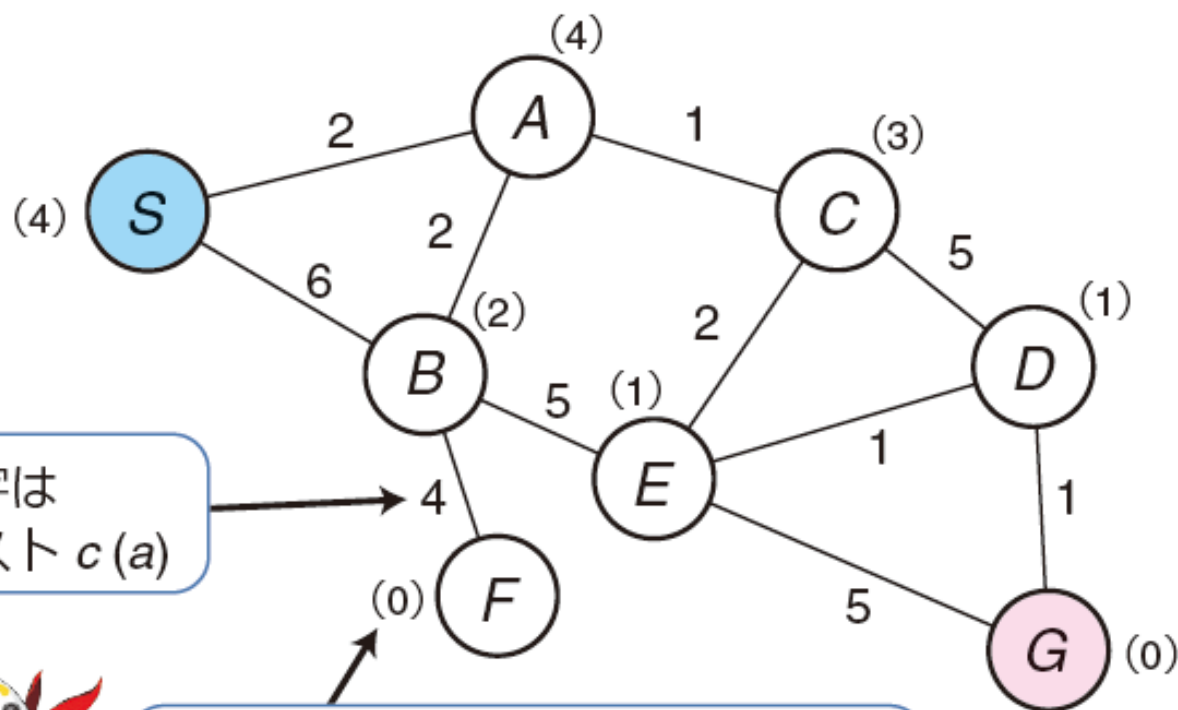
関空に行こう！

三重経由で！

これでいいのか？

スタートからゴールまでのコストが最小になるように、最適な経路を探索する.

3.1.1 経路のコスト



辺の横の数字は
行動 a のコスト $c(a)$



ノードの横の括弧付きの数字は
状態 s の予測評価値 $h(s)$

コストの和を
最小化

図 3.2

コストと予測評価値の記されたグラフ

3.1.2 ヒューリスティック(heuristic)な知識としての予測評価値

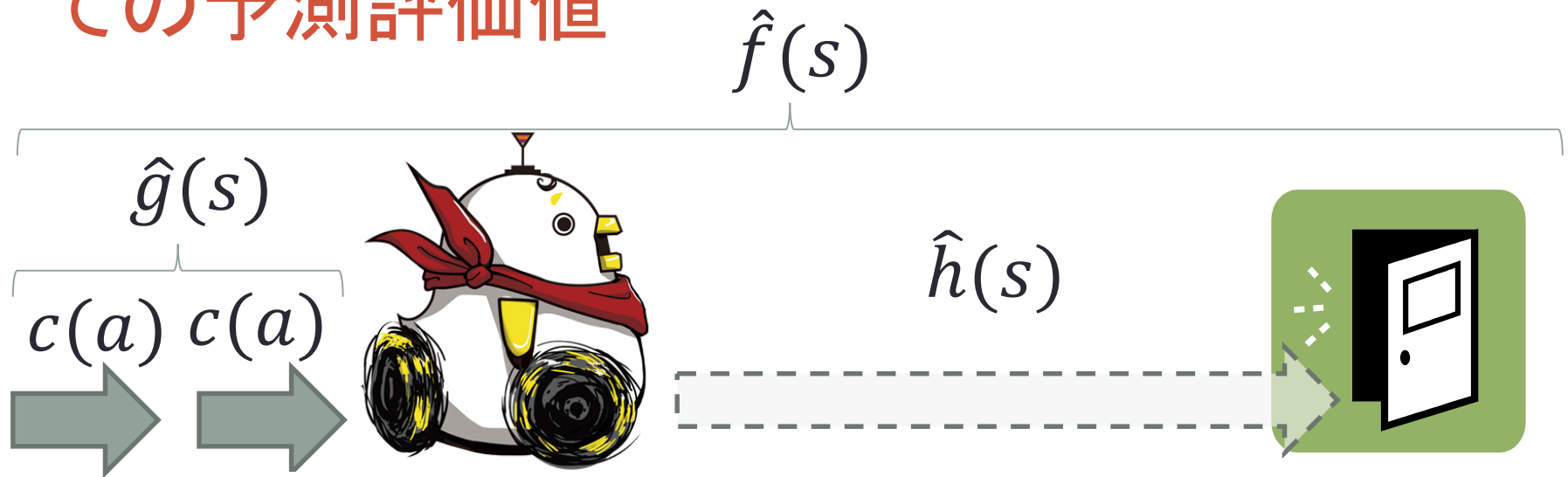


表 3.1 最適経路探索に用いる関数の定義

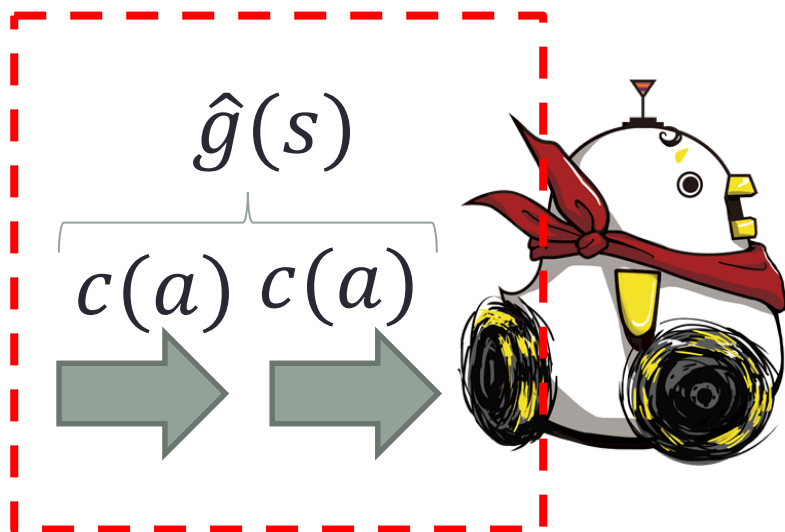
関数記号	説明
$g(s)$	初期状態から状態 s までの最適経路上のコストの総和
$h(s)$	状態 s からゴールまでの最適経路上のコストの総和
$f(s)$	s を経由した場合の最適経路のコスト ($f(s) = h(s) + g(s)$)
$\hat{h}(s)$	$h(s)$ の推定値, 状態 s の予測評価値として用いる
$\hat{g}(s)$	$g(s)$ の推定値
$\hat{f}(s)$	$f(s)$ の推定値 ($\hat{f}(s) = \hat{h}(s) + \hat{g}(s)$)

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！

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コスト値 $\hat{g}(s)$ を再計算し，コストが小さい順に並び替える．
- ⑥ end while 探索を終了．

3.2.2 最適探索の実行例 Dijkstra 算法

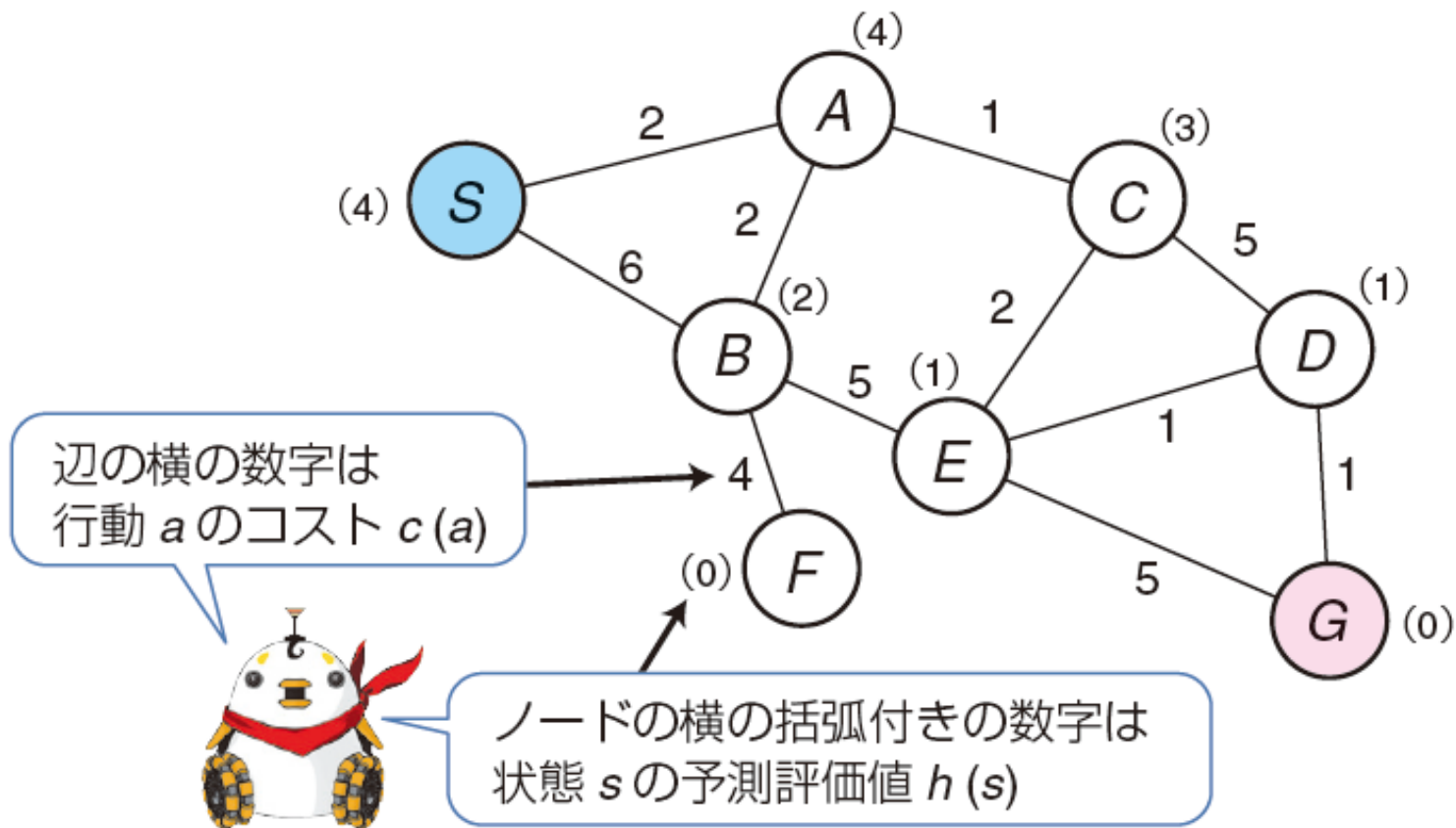


図 3.2

コストと予測評価値の記されたグラフ

実行してみましょう！

3.2.2 最適探索の実行結果

Open List | Close List

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. $F(8)$ | $S(0), A(2), C(3), B(4), E(5), D(6), G(7)$

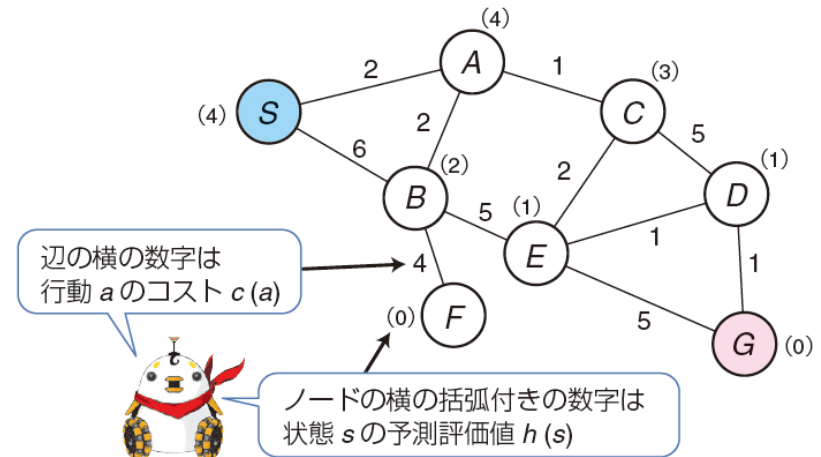
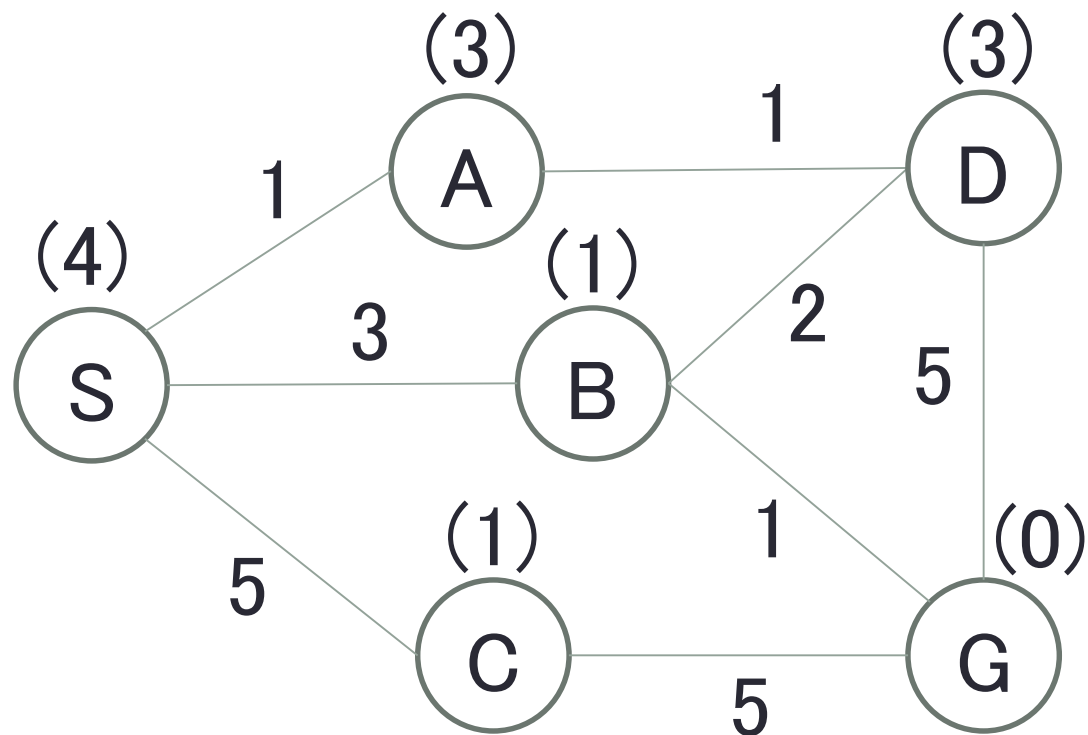


図 3.2 コストと予測評価値の記されたグラフ

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.

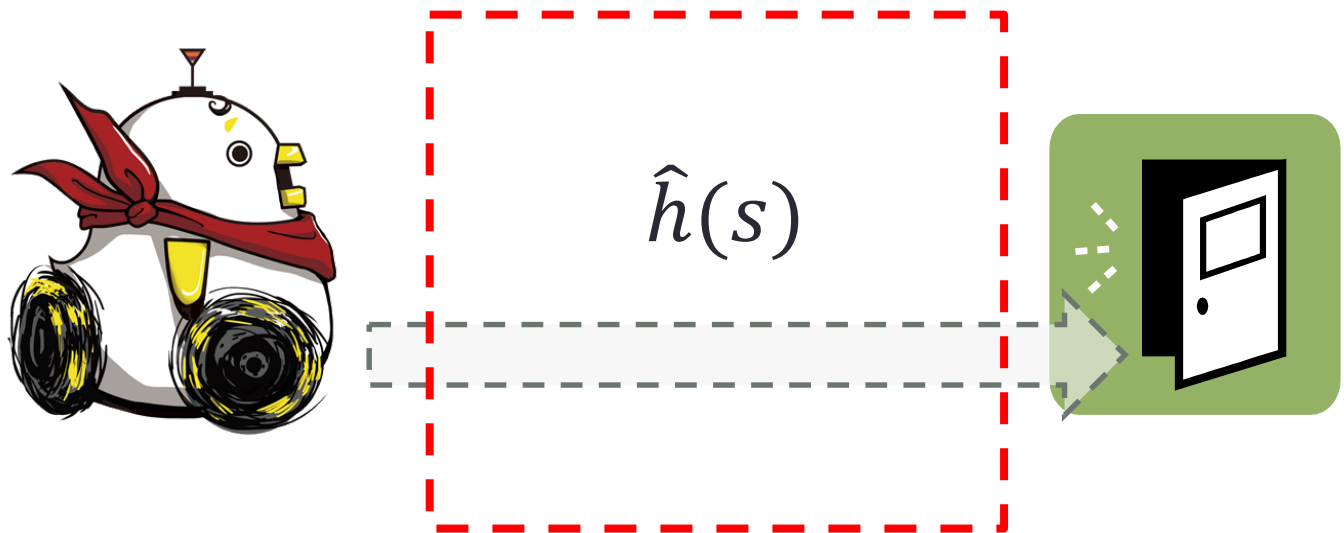


Contents

- 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 を探査することに相当)．
- 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査していないの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 $\hat{h}(s)$ が小さい順に並び替える**．
- ⑥ end while 探索を終了．

3.3.2 最良優先探索の実行例

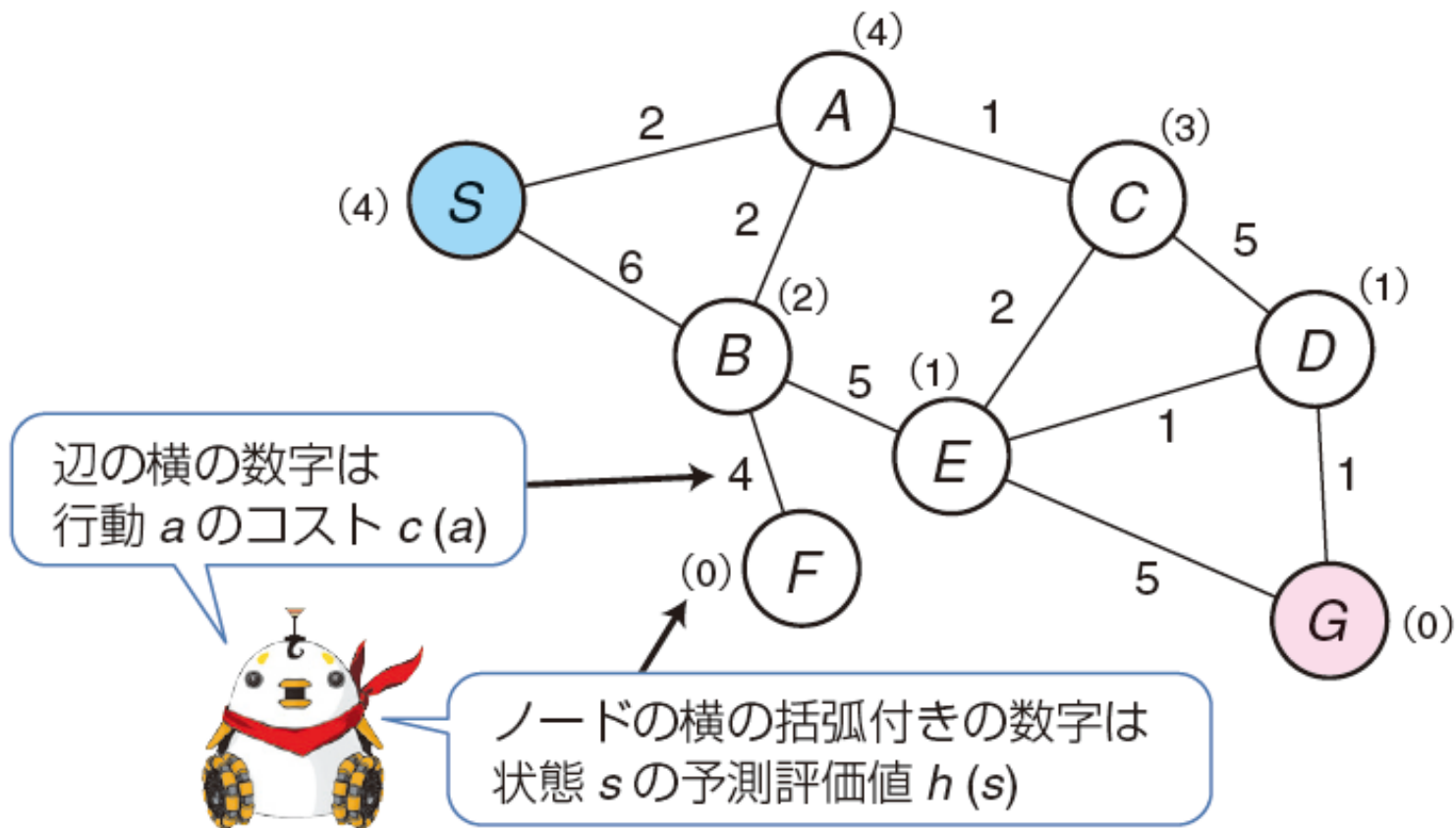


図 3.2

コストと予測評価値の記されたグラフ

実行してみましょう！

3.3.2 最良優先探索の実行結果

Open List | Close List

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)

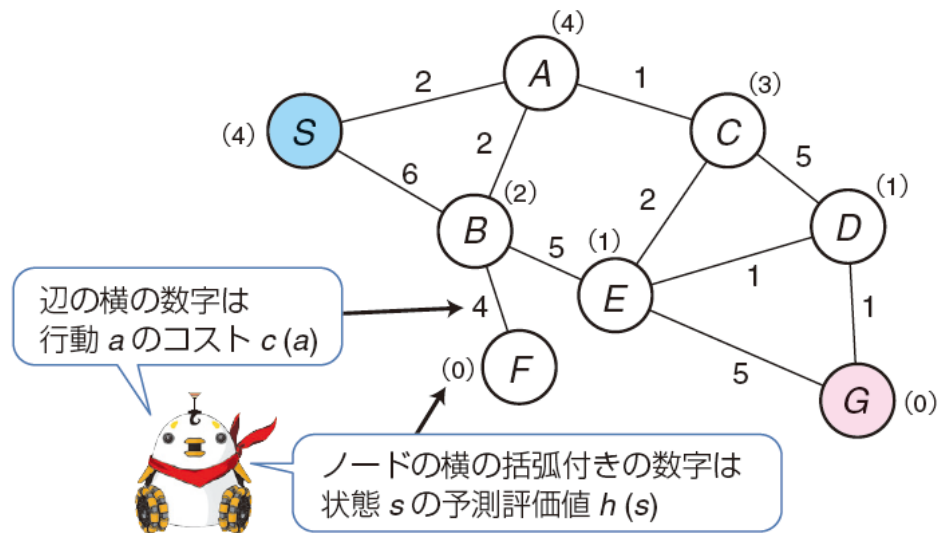
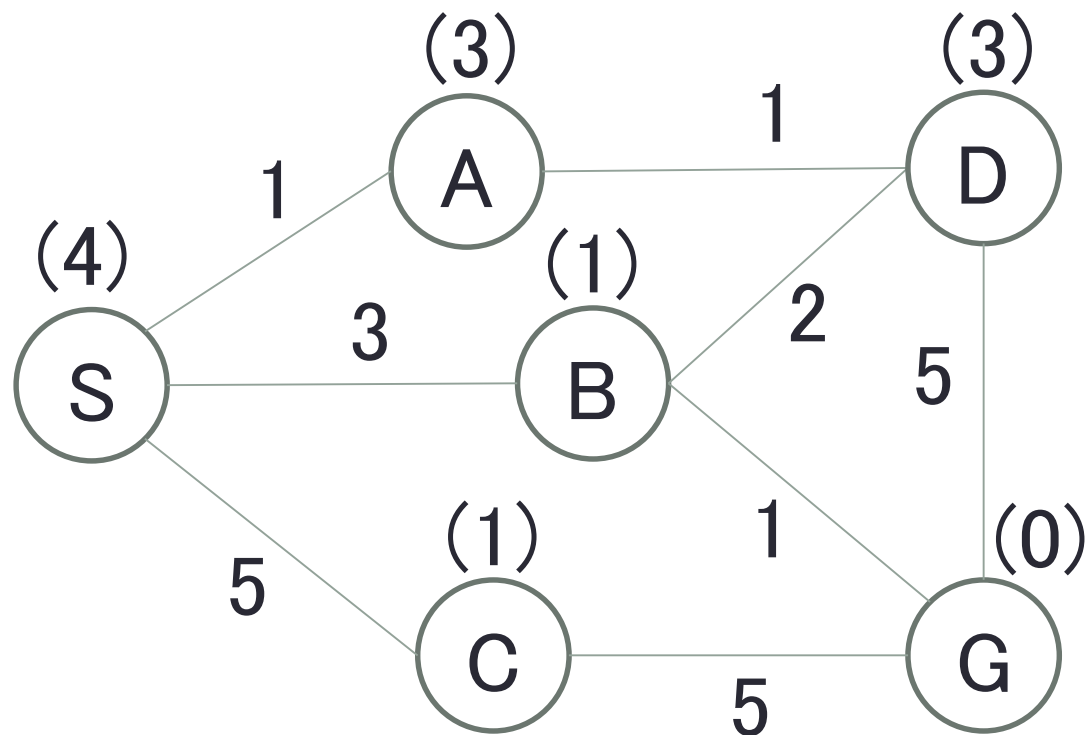


図 3.2 コストと予測評価値の記されたグラフ

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.



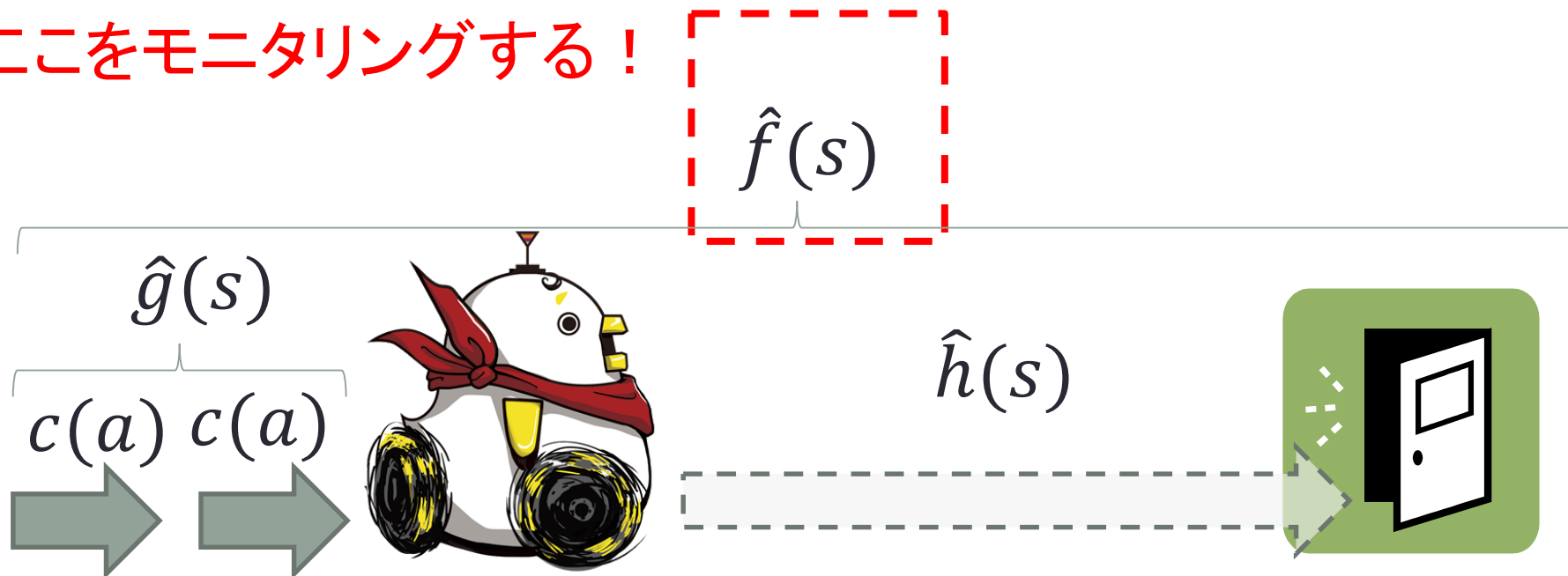
Contents

- 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 $g^{\wedge}(s)$ と, ゴールまでに将来かかるであろう予測評価値 $h^{\wedge}(s)$ 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 から接続されているすべての状態 s' に関してコスト値 $\hat{g}(s')$ 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、既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、**コスト値と予測評価値の和 $\hat{f}(s)$ が小さい順に並び替える。**
- ⑥ end while 探索を終了。

3.4.2 A*アルゴリズムの実行例

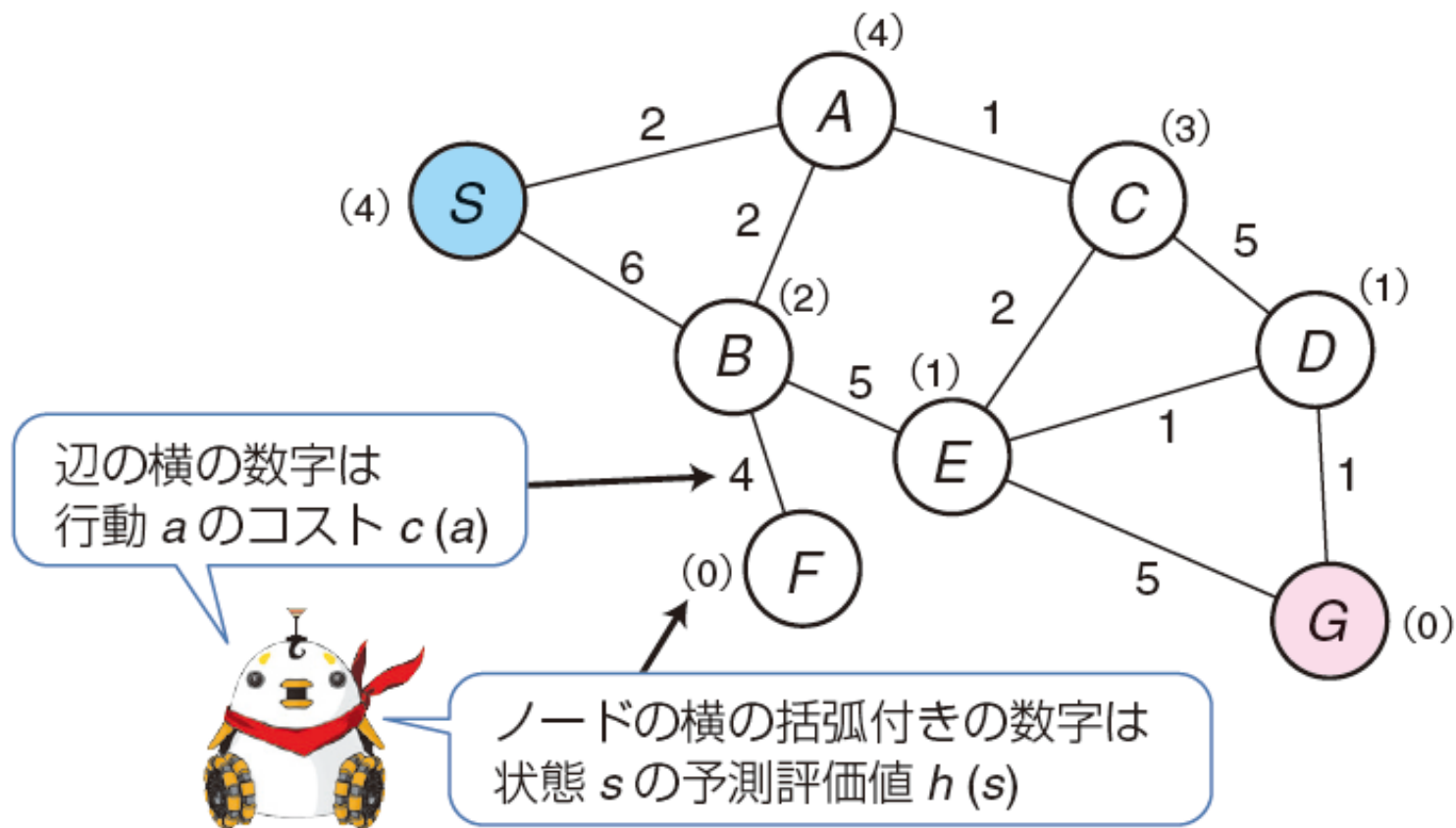


図 3.2

コストと予測評価値の記されたグラフ

実行してみましょう！

3.4.2 A*アルゴリズムの実行結果

Open List | Close List

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)

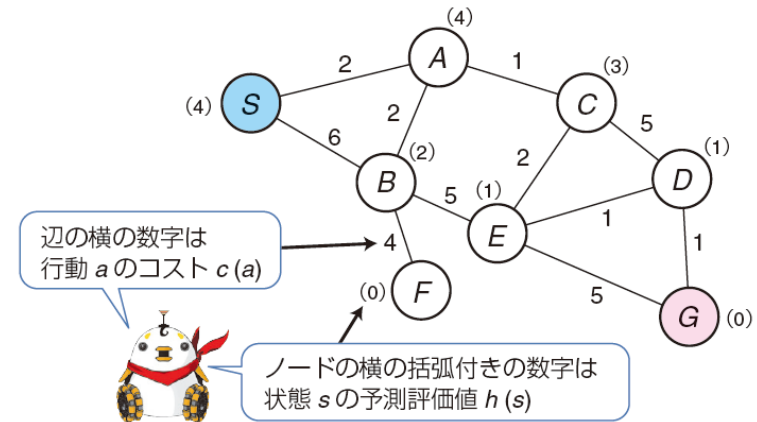
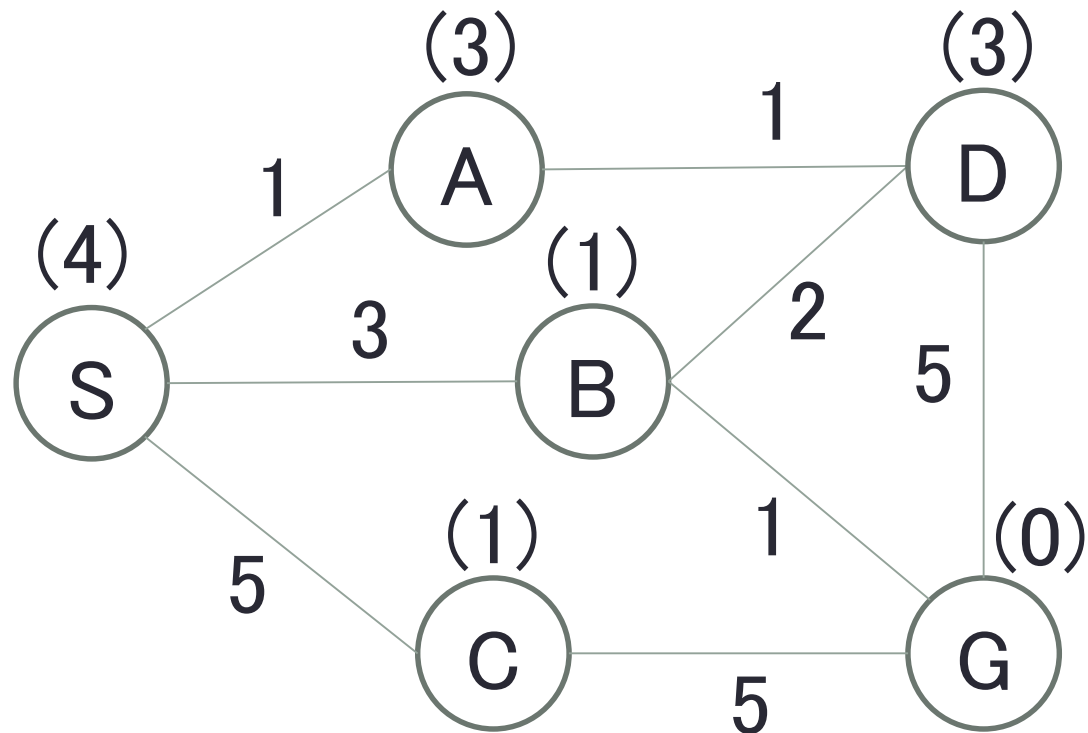


図 3.2 コストと予測評価値の記されたグラフ

演習3-3 A*アルゴリズム

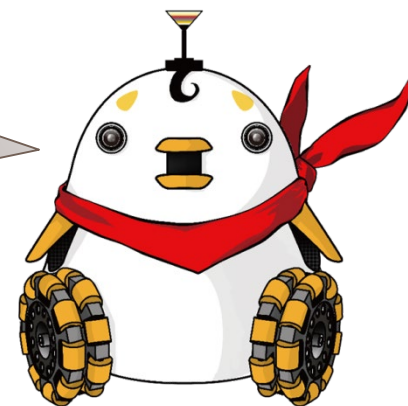
- 下のグラフにおいてSからスタートしてA*アルゴリズムで探索せよ. 探索中の様子をオープンリストで示し, 最終的に得られる経路を示せ.



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!!!!



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

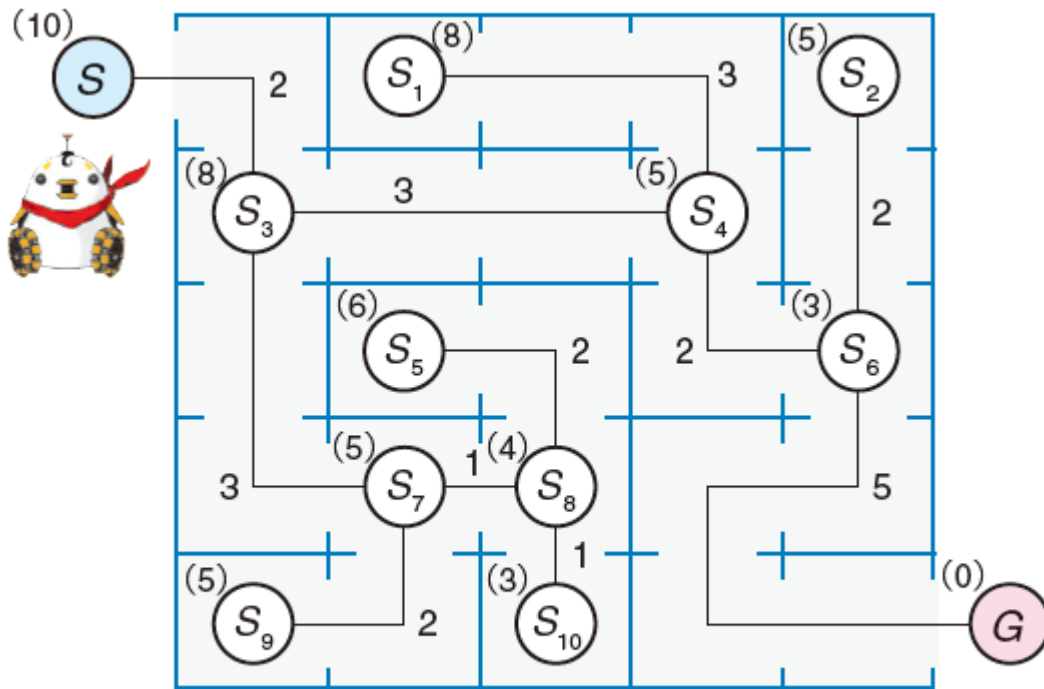


図 3.3 コスト付きのグラフ

- 辺にコスト $c(a)$ を置く
- 予測評価値 $h^{\wedge}(s)$ 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**

解いてみよう

3.5 迷路の最適経路

Open List | Close List

1. S(10) |
2. S₃(10) | S(10)
3. S₄(10), S₇(10) | S(10), S₃(10)
4. S₆(10), S₇(10), S₁(16) | S(10), S₃(10), S₄(10)
5. S₇(10), G(12), S₂(14), S₁(16) | S(10), S₃(10), S₄(10), S₆(10)
6. S₈(10), G(12), S₉(12), S₂(14), S₁(16) | S(10), S₃(10), S₄(10), S₆(10), S₇(10)
7. S₁₀(10), G(12), S₉(12), S₂(14), S₅(14), S₁(16) | S(10), S₃(10), S₄(10), S₆(10), S₇(10), S₈(10)
8. G(12), S₉(12), S₂(14), S₅(14), S₁(16) | S(10), S₃(10), S₄(10), S₆(10), S₇(10), S₈(10), S₁₀(10)
9. S₉(12), S₂(14), S₅(14), S₁(16) | S(10), S₃(10), S₄(10), S₆(10), S₇(10), S₈(10), S₁₀(10), G(12)

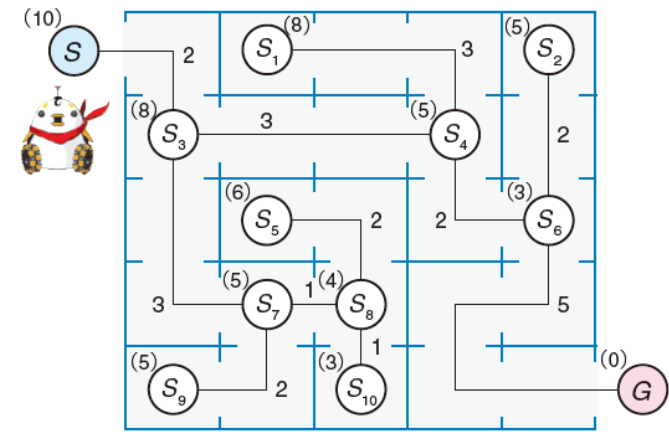


図 3.3 コスト付きのグラフ

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	○	○	△

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.