

人工知能入門

-探索による人工知能-

Lecture 10

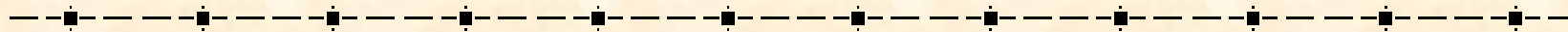
ヒューリスティック探索(2)

<http://www2.teu.ac.jp/gamelab/LECTURES/ArtificialIntelligence/index.html>

第2回目 のミニテスト

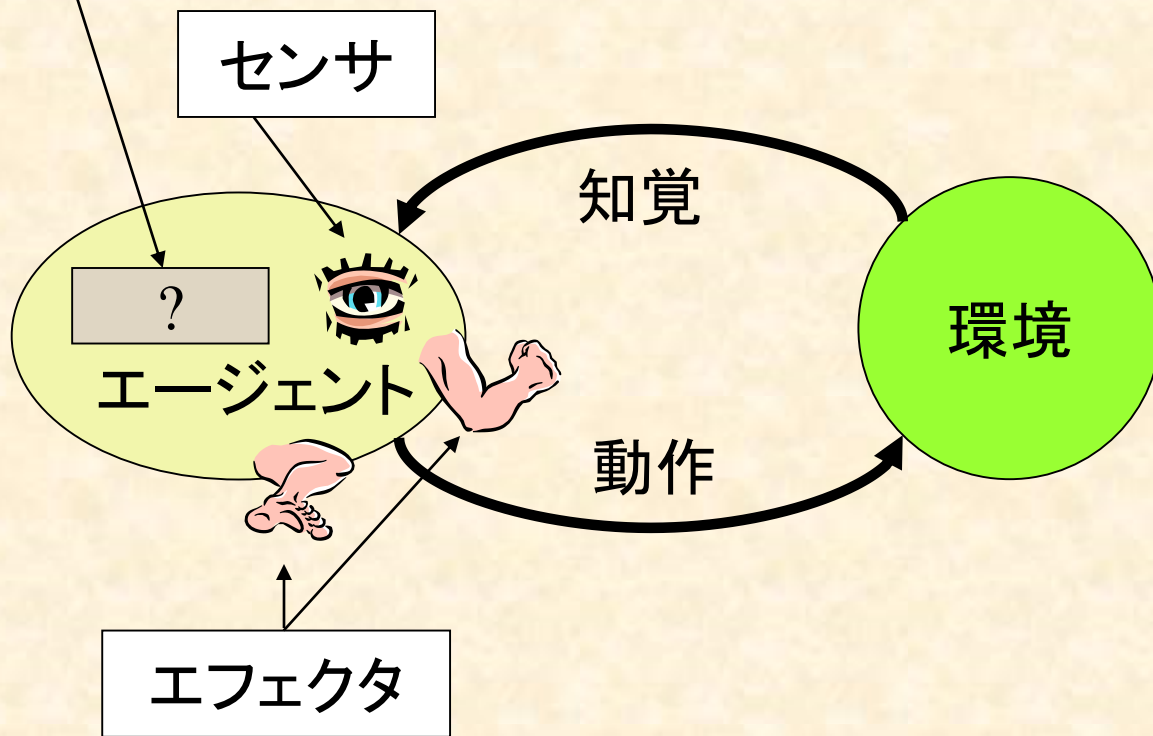
持ち込み可

ミニテストの解説



合理的エージェント

この設計方法は？



探索戦略

探索戦略の種類

✦ 情報のない探索 (盲目探索)

- ◆ 現在の状態からのゴールに至るステップの数や経路コストに関する情報を持たない
- ◆ ゴール状態と非ゴール状態との区別することだけ
- ◆ 幅優先探索、均一コスト探索、深さ優先探索

✦ 情報のある探索 (ヒューリスティック探索)

- ◆ ゴールに近そうな状態を先に展開する等の情報を利用する探索戦略
- ◆ 欲張り探索、 A^*

ヒューリスティック探索

最良優先探索

✦ 一般的探索アルゴリズム

- ◆ 知識を適用できる箇所は待ち行列関数のところだけ
- ◆ この知識は評価関数として与えられる
- ◆ 節点を展開することの好ましさに比例した数字を返す関数

✦ 最良優先探索

- ◆ 評価値が最大の節点が最初に展開されるような順番で待ち行列を並べられる

function Best-First-Search(*problem*, Eval-Fn)

returns a solution sequence

inputs: *problem*, a problem

Eval-Fn, an evaluation function

Queuing-Fn ← a function that orders nodes by Eval-Fn

return General-Search(*problem*, *Queuing-Fn*)

ヒューリスティック探索

欲張り探索

✦ ヒューリスティック関数

- ◆ $h(n)$ = 節点 n の状態からゴール状態までの最短の道の見積もりコスト
- ◆ n がゴールのときに $h(n) = 0$ が成り立ってさえいれば、 h はいかなる関数でもよい

✦ 欲張り探索

- ◆ 展開すべき次の節点の選択に h を用いる最良優先探索

function Greedy-Search(*problem*) **returns** a solution or failure

return Best-First-Search(*problem*, h)

最良優先探索

A* 探索

★ 欲張り探索

- ◆ ゴールまでのコストの見積もりである $h(n)$ を最小化する
- ◆ 探索コストをかなり減らすことができるが、最適でも完全でもない

★ 均一コスト探索

- ◆ 経路のコスト $g(n)$ を最小化する
- ◆ 最適で完全であるが、非効率的である

最良優先探索

A* 探索

★ A* 探索

- ◆ 二つの評価関数を単に足し合わせる:

$$f(n) = g(n) + h(n)$$

- ◆ $f(n) = n$ 経路の最短解の見積りコスト

★ 完全で最適であるための条件: h は許容的

- ◆ h 関数が目標までの実際コストを決して超えない
- ◆ 例: ルーマニア問題の直線距離

function A*-Search(*problem*) **returns** a solution or failure
return Best-First-Search(*problem*, $g+h$)

最良優先探索

A* 探索の振舞い

✦ 単調性

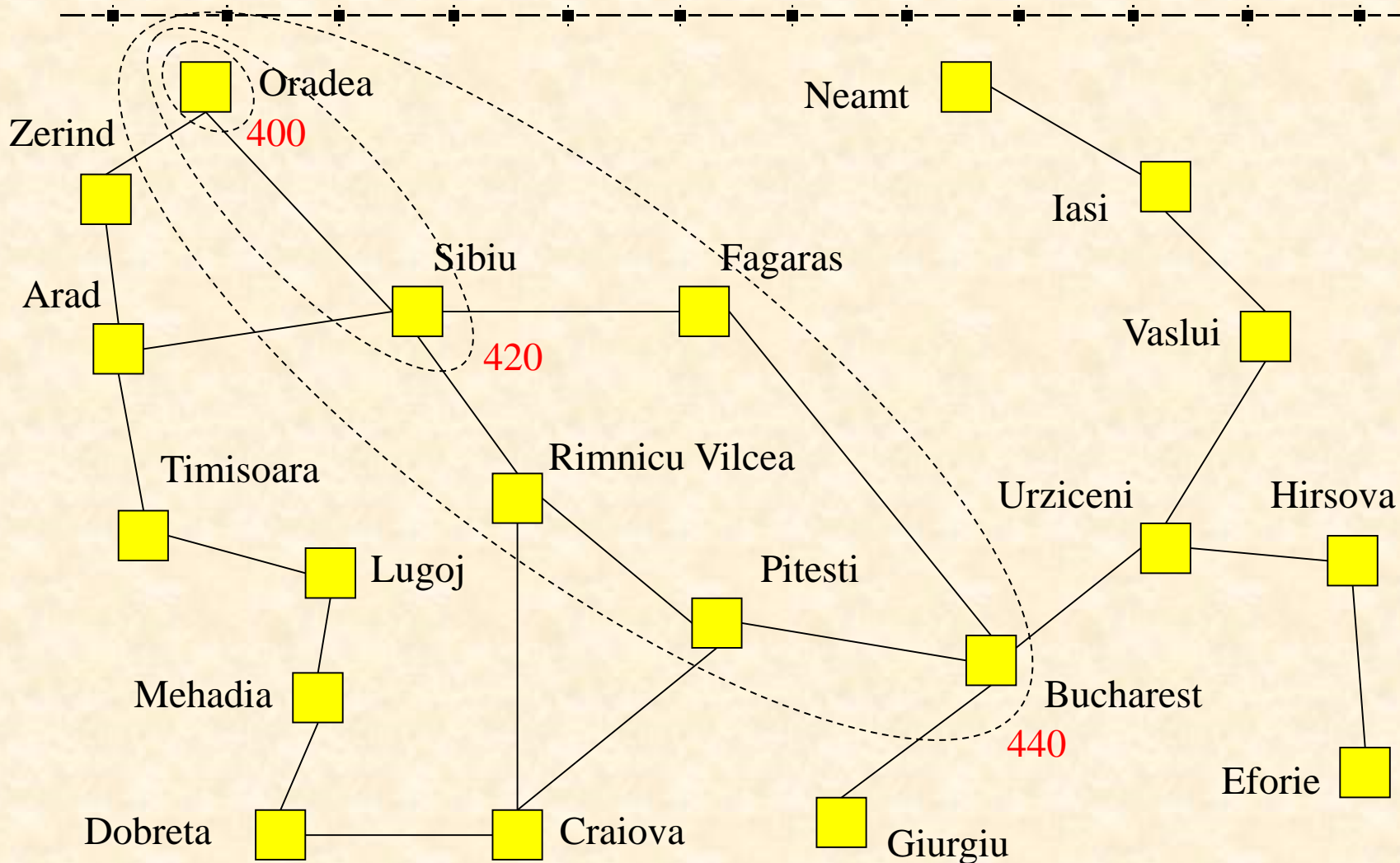
- ◆ f のコストは決して減少していない
- ◆ ほとんどすべての許容的ヒューリスティックで成り立つ

✦ 等高線

- ◆ A* 探索は出発節点から f コストが増加する環状の帯の節点を付け加えながら進んでいく
- ◆ 均一コスト探索 ($h = 0$) では出発状態を中心とする同心円
- ◆ より正確なヒューリスティック: 帯はゴールに向かって延びていき、最適経路の周辺に焦点を集中させる

A* 探索

等高線



最良優先探索

A* 探索の完全性と計算量

✦ 局所有限グラフについてA* は完全

- ◆ 無限の分岐度を有する節点が存在しない
- ◆ 経路コストは有限だが無限の節点を経由する経路が存在しない

✦ A* の計算量

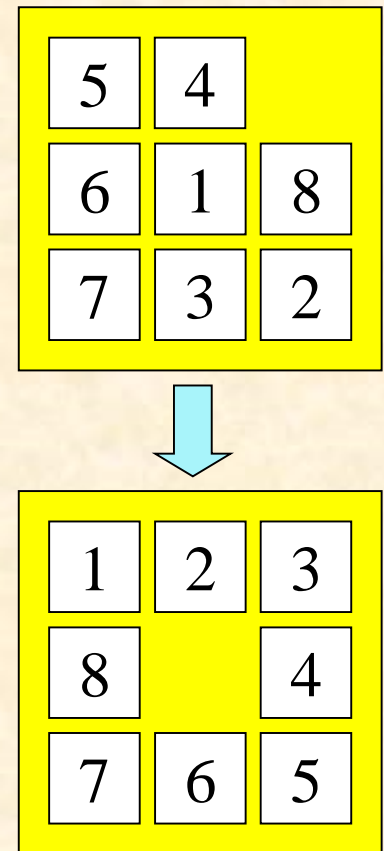
- ◆ ほとんどの問題に対して、ゴール等高線の探索空間内の節点の数が多すぎ
- ◆ ほとんどのヒューリスティックに起こる問題
- ◆ すべての節点をメモリにおいておくために、時間切れの前に普通はメモリがパンクする

ヒューリスティック関数

8パズル

★ 8パズルの特徴

- ◆ 典型的な解の長さ: 20ステップ
- ◆ 分枝度: ほぼ3
- ◆ しらみつぶし探索はおよそ $3^{20} = 3.5 \times 10^9$ 状態を見る
- ◆ 状態の繰り返しに注意する:
異なる状態は $9! = 362,880$ だけ



ヒューリスティック関数

8パズル

★ ヒューリスティック関数の候補

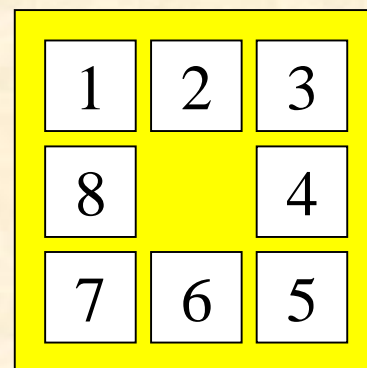
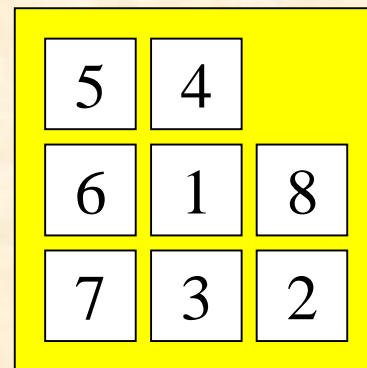
- ◆ h_1 = ゴールの位置にないタイルの数
- ◆ h_2 = ゴール状態からのタイルの距離の和
(街ブロック距離かマンハッタン距離)
- ◆ h_1 と h_2 はともに許容的
- ◆ 右上の出発状態の場合

$$h_1 = 7, h_2 = 2 + 3 + 3 + 2 + 4 + 2 + 0 + 2 = 18$$

- ★ h_2 は h_1 より優位なので h_2 は常に h_1 より優れている

- ◆ いかなる節点 n においても $h_2(n) \geq h_1(n)$ が成り立つ

実際の値を上回らない限り、大きい値のヒューリスティック関数を用いた方が常に良い結果が得られる



まとめ

★ 最良優先探索

- ◆ 評価値が最大の節点が最初に展開されるような順番で待ち行列を並べられる

★ A* 探索

- ◆ 展開すべき次の節点の選択に経路コストとヒューリスティック関数を用いる最良優先探索
- ◆ 完全性があるが、計算量が問題
- ◆ ヒューリスティック関数は計算量に大きい影響を与える
- ◆ 実際の値を上回らない限り、大きい値のヒューリスティック関数を用いた方が常に良い結果が得られる