



IKI30320  
Kuliah 5  
12 Sep 2007

Ruli Manurung

Best-first  
search

Greedy  
best-first  
search

A\* search

Merancang  
heuristic

Search di  
environment  
yang 'sulit'

Ringkasan

## IKI 30320: Sistem Cerdas Kuliah 5: Informed Search

Ruli Manurung

Fakultas Ilmu Komputer  
Universitas Indonesia

12 September 2007



IKI30320  
Kuliah 5  
12 Sep 2007

Ruli Manurung

Best-first  
search

Greedy  
best-first  
search

A\* search

Merancang  
heuristic

Search di  
environment  
yang 'sulit'

Ringkasan

## Outline

- 1 Best-first search
- 2 Greedy best-first search
- 3 A\* search
- 4 Merancang heuristic
- 5 Search di environment yang 'sulit'
- 6 Ringkasan



## Best-first search

IKI30320  
Kuliah 5  
12 Sep 2007

Ruli Manurung

Best-first  
search

Greedy  
best-first  
search

A\* search

Merancang  
heuristic

Search di  
environment  
yang 'sulit'

Ringkasan

### Prinsip best-first search

Lakukan node expansion terhadap node di *fringe* yang nilai  $f(n)$ -nya paling kecil.

- Ide dasar:  $f(n)$  adalah sebuah *evaluation function* → fungsi yang menyatakan *perkiraan* seberapa “bagus” sebuah node.
- Kenapa *perkiraan*? Kalau tidak, bukan *search* namanya!
- Implementasi: *fringe* adalah sebuah *priority queue* di mana node disortir berdasarkan  $f(n)$ .
- Contoh:
  - Uniform-cost search
  - Greedy (best-first) search
  - A\* search



## Heuristic function

IKI30320  
Kuliah 5  
12 Sep 2007

Ruli Manurung

Best-first  
search

Greedy  
best-first  
search

A\* search

Merancang  
heuristic

Search di  
environment  
yang 'sulit'

Ringkasan

- Kunci keberhasilan *best-first search* terletak di *heuristic function*.
- Heuristic adalah:
  - *rule of thumb*
  - “kiat-kiat sukses”, “tips-tips keberhasilan”
  - informasi tambahan bagi si agent (agar lebih sukses) → *informed search*
- *Heuristic function*  $h(n)$  adalah fungsi yang menyatakan estimasi cost dari  $n$  ke *goal state*.
- Ada banyak kemungkinan *heuristic function* untuk sebuah masalah.



## Contoh heuristic function

IKI30320  
Kuliah 5  
12 Sep 2007

Ruli Manurung

Best-first search

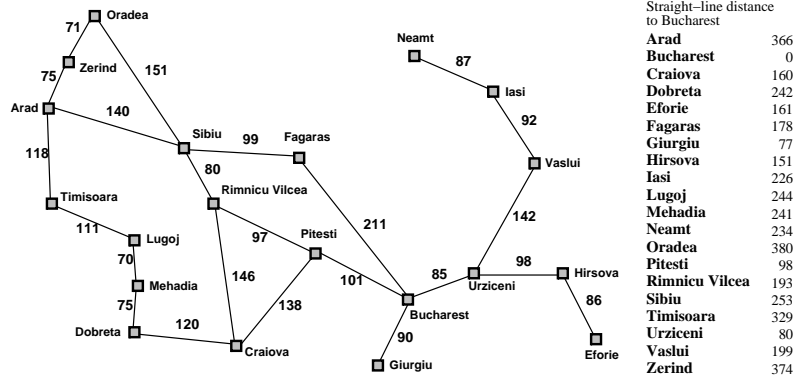
Greedy best-first search

A\* search

Merancang heuristic

Search di environment yang 'sulit'

Ringkasan



Sebuah heuristic function untuk agent turis Rumania

$h_{SLD}(n)$  = jarak *straight-line distance* dari  $n$  ke Bucharest.



## Greedy best-first search

IKI30320  
Kuliah 5  
12 Sep 2007

Ruli Manurung

Best-first search

Greedy best-first search

A\* search

Merancang heuristic

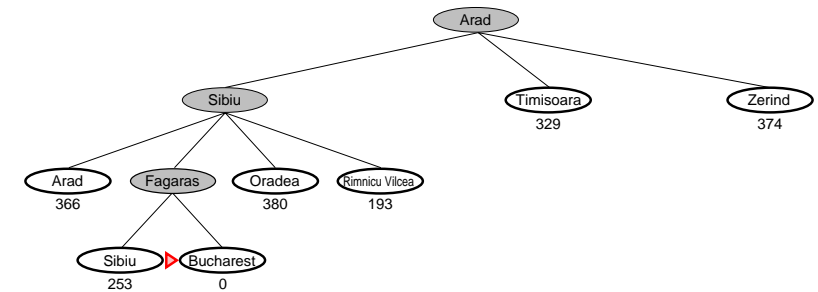
Search di environment yang 'sulit'

Ringkasan

### Prinsip greedy best-first search

Lakukan node expansion terhadap node di *fringe* yang nilai  $h(n)$ -nya paling kecil.

Greedy best-first search selalu memilih node yang **kelihatannya** paling dekat ke *goal*.



## Sifat greedy best-first search

IKI30320  
Kuliah 5  
12 Sep 2007

Ruli Manurung

Best-first search

Greedy best-first search

A\* search

Merancang heuristic

Search di environment yang 'sulit'

Ringkasan

- **Complete?** Ya, jika *state space* terbatas dan pengulangan state ditangani. (Lihat Neamt → Oradea)
- **Time complexity?** Secara teoritis,  $O(b^m)$ , tetapi heuristic function yang baik akan mempercepat drastis
- **Space complexity?**  $O(b^m)$  → semua node disimpan di memory
- **Optimal?** Tidak.



## A\* search

IKI30320  
Kuliah 5  
12 Sep 2007

Ruli Manurung

Best-first search

Greedy best-first search

A\* search

Merancang heuristic

Search di environment yang 'sulit'

Ringkasan

### Prinsip A\* search

Hindari node yang berada di *path* yang "mahal"

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

- $g(n)$  = Path cost ke  $n$
- $h(n)$  = Estimasi path cost dari  $n$  ke *goal*
- $f(n)$  = Estimasi **total** cost melalui  $n$



## Contoh penelusuran A\* search

IKI30320  
Kuliah 5  
12 Sep 2007

Ruli Manurung

Best-first search

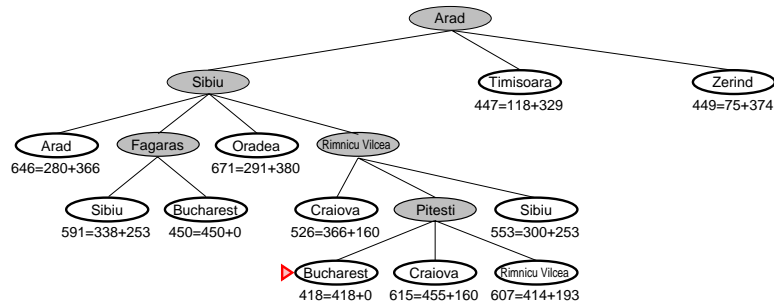
Greedy best-first search

A\* search

Merancang heuristic

Search di environment yang 'sulit'

Ringkasan



## Admissible heuristic

IKI30320  
Kuliah 5  
12 Sep 2007

Ruli Manurung

Best-first search

Greedy best-first search

A\* search

Merancang heuristic

Search di environment yang 'sulit'

Ringkasan

A\* search menggunakan heuristic yang **admissible**

$0 \leq h(n) \leq h^*(n)$ , di mana  $h^*(n)$  adalah cost dari  $n$  yang **sebenarnya**.

**Bahasa gampangnya:** nilai sebuah heuristic function tidak pernah **melebihi** cost ke goal yang sebenarnya.

Contoh:  $h_{SLD}(n)$

### Theorem

A\* search adalah **optimal**.



## Bukti optimalitas A\* (1)

IKI30320  
Kuliah 5  
12 Sep 2007

Ruli Manurung

Best-first search

Greedy best-first search

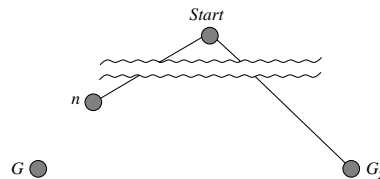
A\* search

Merancang heuristic

Search di environment yang 'sulit'

Ringkasan

Andaikan  $G_2$  adalah goal *suboptimal* di dalam *fringe*. Ambil  $n$  sebuah *fringe* node pada path menuju  $G_1$ , goal *optimal*, sbb:



- $f(G_2) = g(G_2)$ , karena  $h(G_2) = 0$
- $g(G_2) > g(G_1)$ , karena  $G_2$  tidak optimal
- $g(G_1) \geq f(n)$ , karena  $h$  *admissible*

Karena  $f(G_2) > f(n)$ , algoritma A\* search tidak pernah akan memilih  $G_2$  untuk di-expand. **Teorema terbukti!**



## Consistency sebuah heuristic

IKI30320  
Kuliah 5  
12 Sep 2007

Ruli Manurung

Best-first search

Greedy best-first search

A\* search

Merancang heuristic

Search di environment yang 'sulit'

Ringkasan

- Sebuah *heuristic* dikatakan **consistent** jika:  
 $h(n) \leq c(n, a, n') + h(n')$

- Jika  $h$  konsisten, maka:

$$\begin{aligned} f(n') &= g(n') + h(n') \\ &= g(n) + c(n, a, n') + h(n') \\ &\geq g(n) + h(n) \\ &\geq f(n) \end{aligned}$$

- Pada sembarang path, nilai  $f(n)$  tidak pernah turun (**nondecreasing**), atau **monotonic**.



## Bukti optimalitas A\* (2)

IKI30320  
Kuliah 5  
12 Sep 2007

Ruli Manurung

Best-first search

Greedy best-first search

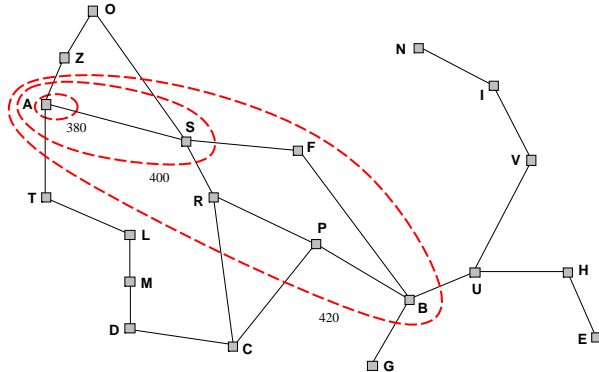
A\* search

Merancang heuristic

Search di environment yang 'sulit'

Ringkasan

- **Node expansion** A\* berdasarkan urutan nilai  $f$ .
- Bayangkan penelusuran *state space* yang dilakukan A\* menambahkan  **$f$ -contour**.



- Bandingkan dengan “lapisan” yang ditelusuri *breadth-first* dan *uniform-cost*.

Di dalam contour ke- $i$  terdapat semua node



## Contoh admissible heuristic

IKI30320  
Kuliah 5  
12 Sep 2007

Ruli Manurung

Best-first search

Greedy best-first search

A\* search

Merancang heuristic

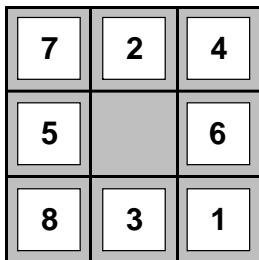
Search di environment yang 'sulit'

Ringkasan

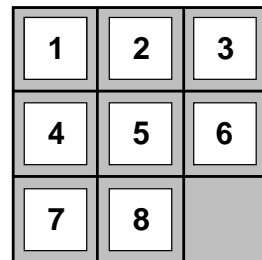
$h(n)$  untuk 8-puzzle

$h_1(n)$ : jumlah angka yang salah posisi

$h_2(n)$ : jumlah jarak semua angka dari posisi yang benar



Start State



Goal State

$$h_1(s) = 6$$

$$h_2(s) = 4+0+3+3+1+0+2+1=14$$



## Sifat A\* search

IKI30320  
Kuliah 5  
12 Sep 2007

Ruli Manurung

Best-first search

Greedy best-first search

A\* search

Merancang heuristic

Search di environment yang 'sulit'

Ringkasan

- **Complete?** Ya, kecuali jumlah node di mana  $f \leq f(G)$  tak terbatas
- **Time complexity?** Eksponensial dalam (error  $h \times$  jumlah step solusi)
- **Space complexity?**  $O(b^m) \rightarrow$  semua node disimpan di memory
- **Optimal?** Ya.  
A\* meng-*expand* semua node di mana  $f(n) < C^*$   
A\* (mungkin) meng-*expand* beberapa node di mana  $f(n) = C^*$   
A\* tidak pernah meng-*expand* node di mana  $f(n) > C^*$



## Membandingkan dua heuristic

IKI30320  
Kuliah 5  
12 Sep 2007

Ruli Manurung

Best-first search

Greedy best-first search

A\* search

Merancang heuristic

Search di environment yang 'sulit'

Ringkasan

- $h_1$  dan  $h_2$  sama-sama *admissible*. Mana yang **lebih baik**? Bandingkan jumlah node yang di-expand:

$d$	IDS	A*( $h_1$ )	A*( $h_2$ )
12	3,473,941	539	113
24	54,000,000,000	39,135	1,641

- Jika  $h_2(n) \geq h_1(n)$  untuk semua  $n$  (dan keduanya *admissible*), dikatakan bahwa  $h_2$  men-*dominate*  $h_1$  dan lebih baik untuk *search*.
- Semakin besar nilai  $h(n)$ , semakin dekat ke  $h^*(n)$ , semakin banyak node yang **tidak di-expand** (di-*prune*), semakin efisien *search*-nya!



## Merancang admissible heuristic

IKI30320  
Kuliah 5  
12 Sep 2007

Ruli Manurung

Best-first search

Greedy best-first search

A\* search

Merancang heuristic

Search di environment yang 'sulit'

Ringkasan

- *Admissible heuristic* dapat diperoleh dari *solution cost* yang sebenarnya dari variasi masalah yang **dipermudah** (*relaxed*).
- Contoh:
  - Andaikan masalah 8-puzzle dipermudah sehingga sebuah angka bisa dipindahkan **ke mana saja**. *Cost* dari solusinya =  $h_1$ .
  - Andaikan masalah 8-puzzle dipermudah sehingga sebuah angka bisa dipindahkan **ke tetangga mana saja** (kosong atau tidak). *Cost* dari solusinya =  $h_2$ .
- *Optimal solution cost* dari masalah yang dipermudah **tidak akan melebihi** *optimal solution cost* masalah yang sebenarnya → *admissible*!
- *Admissible heuristic* bisa juga diperoleh dari **sub-masalah**.



## Environment yang tidak *observable*

IKI30320  
Kuliah 5  
12 Sep 2007

Ruli Manurung

Best-first search

Greedy best-first search

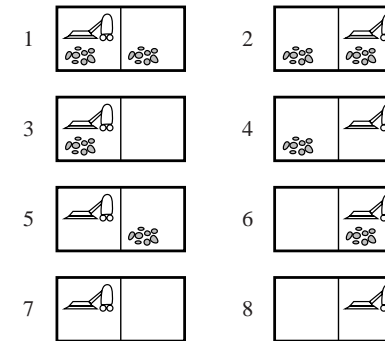
A\* search

Merancang heuristic

Search di environment yang 'sulit'

Ringkasan

- Selama ini, kita berasumsi bahwa *environment* di mana *problem solving agent* kita berada *fully observable*.
- Apa yang terjadi jika si *agent* tidak memiliki **sensor**?



- *Initial state* bisa di mana saja: {1, 2, 3, 4, 5, 6, 7, 8}
- Setelah *DoKeKanan*, bisa di: {2, 4, 6, 8}
- Solusi adalah rangkaian tindakan [*DoKeKanan*, *DoSedot*, *DoKeKiri*, *DoSedot*]



## Sensorless problem

IKI30320  
Kuliah 5  
12 Sep 2007

Ruli Manurung

Best-first search

Greedy best-first search

A\* search

Merancang heuristic

Search di environment yang 'sulit'

Ringkasan

- Si agent harus mencatat **himpunan physical state** ( $S_p$ ) yang  **mungkin** sedang terjadi → **belief state** ( $S_b$ ).
- *Search* dilakukan dalam *space* yang terdiri dari **belief state**, bukan **physical state**.
- **Belief state**  $S'_b$  yang dihasilkan suatu *action* terhadap **belief state**  $S_b$  adalah **union** dari semua **physical state**  $S'_p$  yang dihasilkan *action* tersebut terhadap semua **physical state**  $S_p \in S_b$ .
- Sebuah solusi adalah path yang menuju **belief state** di mana **semua** member **physical state**-nya adalah **goal**.



## Contoh belief state VACUUM CLEANER WORLD

IKI30320  
Kuliah 5  
12 Sep 2007

Ruli Manurung

Best-first search

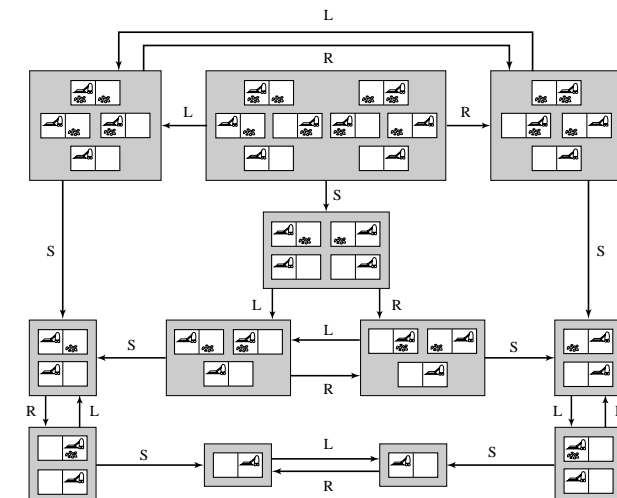
Greedy best-first search

A\* search

Merancang heuristic

Search di environment yang 'sulit'

Ringkasan





## Contingency problem

IKI30320  
Kuliah 5  
12 Sep 2007

Ruli Manurung

Best-first  
search

Greedy  
best-first  
search

A\* search

Merancang  
heuristic

Search di  
environment  
yang 'sulit'

Ringkasan

- Selama ini, kita berasumsi bahwa *environment* di mana *problem solving agent* kita berada *deterministic*.
- Bayangkan robot pembersih kita cacat: jika *DoSedot* dilakukan di ruangan bersih, kadang-kadang ia malah membuatnya kotor! Bagaimana *belief state space*-nya?
- Sekarang bayangkan robot ini punya *sensor* yang melihat apakah ruangan kotor.
- Solusi sekarang bukanlah rangkaian tindakan (*action sequence*), tetapi *action tree*, mis:  
[*DoSedot*, *DoKeKanan*, **if** [*B*, *Kotor*] **then** *DoSedot*].

### Contingency problem

Masalah di mana agent menerima input baru dari sensor **setelah** bertindak.



## Ringkasan

IKI30320  
Kuliah 5  
12 Sep 2007

Ruli Manurung

Best-first  
search

Greedy  
best-first  
search

A\* search

Merancang  
heuristic

Search di  
environment  
yang 'sulit'

Ringkasan

- Best-first search
  - Uniform-cost search:  $f(n) = g(n)$
  - Greedy best-first search:  $f(n) = h(n)$
  - A\* search:  $f(n) = g(n) + h(n)$
- Dengan *heuristic* yang *admissible* dan *consistent*, A\* pasti *complete* dan *optimal*.
- *Heuristic* demikian dapat diperoleh dari variasi masalah yang *dipermudah*, atau *submasalah*.
- Search di mana environment-nya **tidak observable** atau **non-deterministic** masih bisa diatasi.