



Kecerdasan Buatan JTI POLINEMA

Aplikasi Strategi Pencarian

Tim Pengajar Mata Kuliah Kecerdasan Buatan Tahun 2020





Tujuan

- Mahasiswa memahami teknik-teknik yang digunakan pada Strategi Pencarian dalam Kecerdasan Buatan.
 - Mahasiswa memahami Strategi Pencarian Informed Search yakni:
 - **Best-First Search (Greedy dan A*)**
-





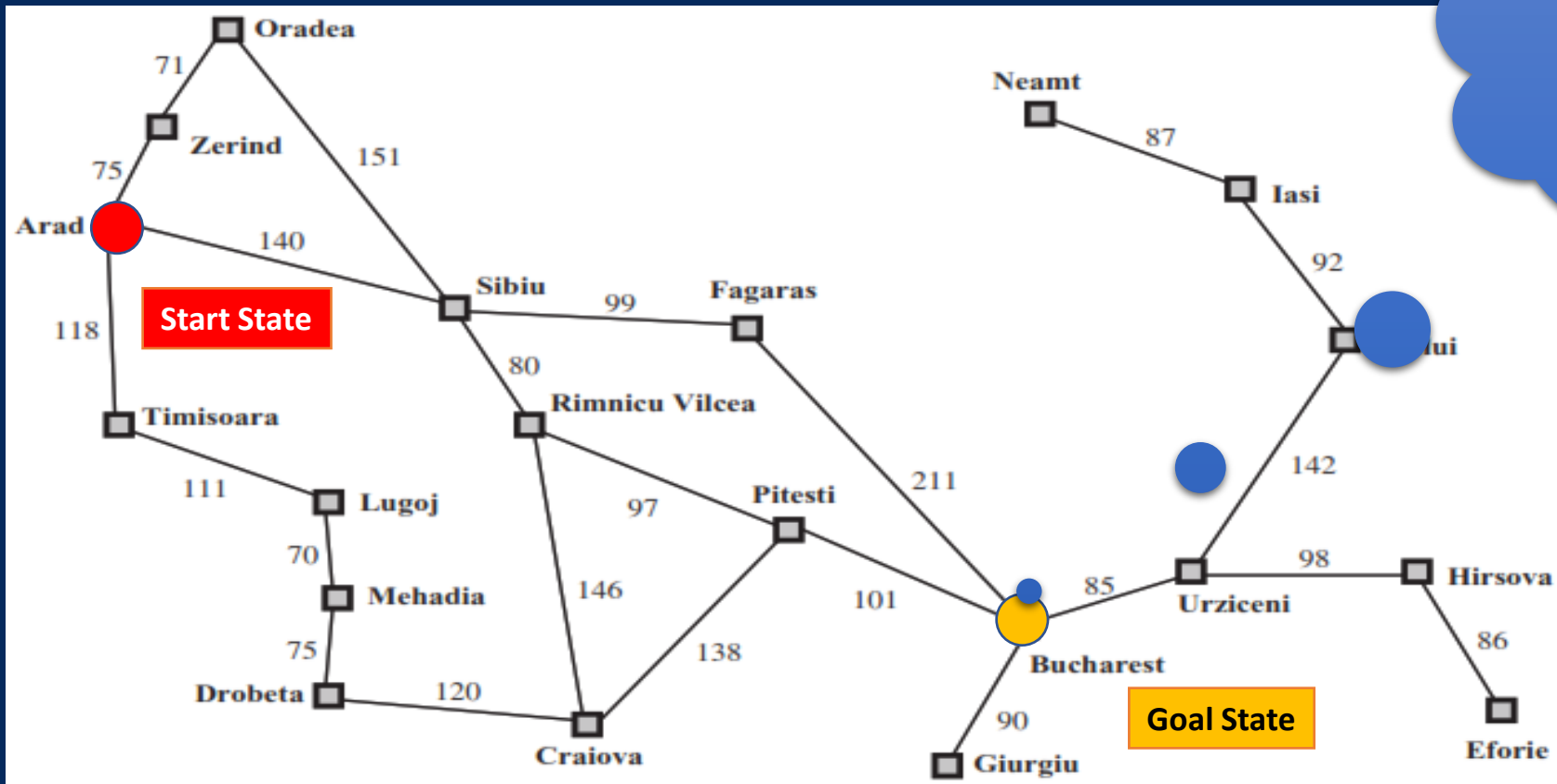
Mengingat Kembali

- Uninformed Search
 - Teknik penyelesaian masalah tanpa adanya informasi mengenai permasalahannya.
 - Dapat menyelesaikan permasalahan namun tidak efisien.
 - 4 parameter pengukuran:
 - **Completeness, Time Complexity, Space Complexity, dan Optimality**
 - Contoh: Breadth-First Search dan Depth-First Search.
-





Kembali ke Romania



Problem – jarak terpendek yang akan ditempuh oleh Agen dari Arad ke Bucharedt





Mengingat Kembali – Definisi

- Problem dapat didefinisikan ke dalam 5 komponen yakni:
 - Initial State atau Start State (S)
 - Action
 - Transition Model
 - Goal Test (G)
 - Path Cost
-





Initial atau Start State

- Titik awal agen memulai pencarian solusi.
- Contoh: Dalam kasus Arad-Bucharest, maka *In(Arad)*.





Action

- **Action** = langkah yang dijalankan oleh Agen.
 - Diberikan sebuah keadaan tertentu (state), s , maka **ACTION(s)** memberikan satu himpunan aksi yang dapat dieksekusi dalam s .
 - Contoh: dari state *In(Arad)*, aksi yang dapat dilakukan adalah *{Go(Sibiu), Go(Timisoara, Go(Zerind))}*
-





Transition Model

- Satu deskripsi dari yang dilakukan oleh setiap aksi melalui fungsi **RESULT(s,a)**, yakni state hasil dari aksi a dalam state **s**.
- Contoh: **RESULT(In(Arad), Go(Zerind)) = In(Zerind)**





Goal Test

- Menetapkan bahwa state yang dicapai adalah Goal state (G) atau bukan.
 - G tidak selalu dinyatakan secara eksplisit dan test akan meyakinkan bahwa satu state merupakan G.
 - Contoh: G untuk Agen di Romania adalah ***{In(Bucharest)}***.
-





Path Cost

- Fungsi biaya jalan (path) yang memberikan biaya numerik ke setiap jalan. Agen problem-solving, memilih fungsi biaya yang mereflesikan ukuran kinerjanya.
- Biaya langkah (step cost) aksi a dari state s menuju state s' dinyatakan dengan (s, a, s') .
- Contoh: $(In(Arad), Go\{Zerind), In(Zerind))$





Pseudo-code Agen Problem-Solving

function SIMPLE-PROBLEM-SOLVING-AGENT(*percept*) **returns** an action

persistent: *seq*, an action sequence, initially empty

state, some description of the current world state

goal, a goal, initially null

problem, a problem formulation

state \leftarrow UPDATE-STATE(*state*, *percept*)

if *seq* is empty **then**

goal \leftarrow FORMULATE-GOAL(*state*)

problem \leftarrow FORMULATE-PROBLEM(*state*, *goal*)

seq \leftarrow SEARCH(*problem*)

if *seq* = *failure* **then return** a null action

action \leftarrow FIRST(*seq*)

seq \leftarrow REST(*seq*)

return *action*

Figure 3.1 A simple problem-solving agent. It first formulates a goal and a problem, searches for a sequence of actions that would solve the problem, and then executes the actions one at a time. When this is complete, it formulates another goal and starts over.





Solusi dan Solusi Optimal

- **Solusi** terhadap satu permasalahan adalah serangkaian aksi yang bergerak dari initial state (S) menuju goal state (G).
 - **Kualitas solusi** diukur dengan **fungsi biaya jalan (path cost)**, dan **solusi optimal** memiliki fungsi biaya jalan **terendah** dari semua solusi yang tersedia.
-





Informed Search

- **Informed** = terinformasikan atau telah diberi informasi lebih awal.
 - Teknik penyelesaian masalah dengan diberikan satu panduan untuk mencari solusi-solusi.
 - Menggunakan pengetahuan spesifik-problem.
 - Memperoleh solusi lebih efisien daripada strategi uninformed.
 - Contoh: **Best-First Search (Greedy dan A*)**
-





Pendekatan Umum Informed Search

- **Best-First Search**

- Algoritma TREE-SEARCH atau GRAPH-SEARCH yang mana simpul (node) yang dipilih untuk dibuka (expand) didasarkan pada **evaluation function, $f(n)$** .
 - **$f(n)$** adalah estimasi biaya, maka simpul dengan biaya termurah (cheapest cost) akan dibuka lebih dulu.
 - Pemilihan **f** akan menentukan strategi pencarian.
-





Fungsi Heuristik

- Sebagian besar algoritma best-first mengikut sertakan sebuah komponen dari **f** yakni **fungsi heuristik, $h(n)$**

$h(n)$ = biaya perkiraan dari jalan termurah (cheapest path) dari state pada simpul **n** ke Goal state (G)

- Contoh: Kasus Romania, **$h(n)$ = garis lurus dari Arad ke Bucharest (hSLD). SLD = straight-line distance**





Greedy Best-First Search

- Membuka simpul yang terdekat dengan Goal **hanya** menggunakan fungsi heuristik, **$f(n) = h(n)$**
- **Kekurangan**: dapat terjebak pada **infinite loop** sehingga tidak pernah berhasil mencapai Goal state



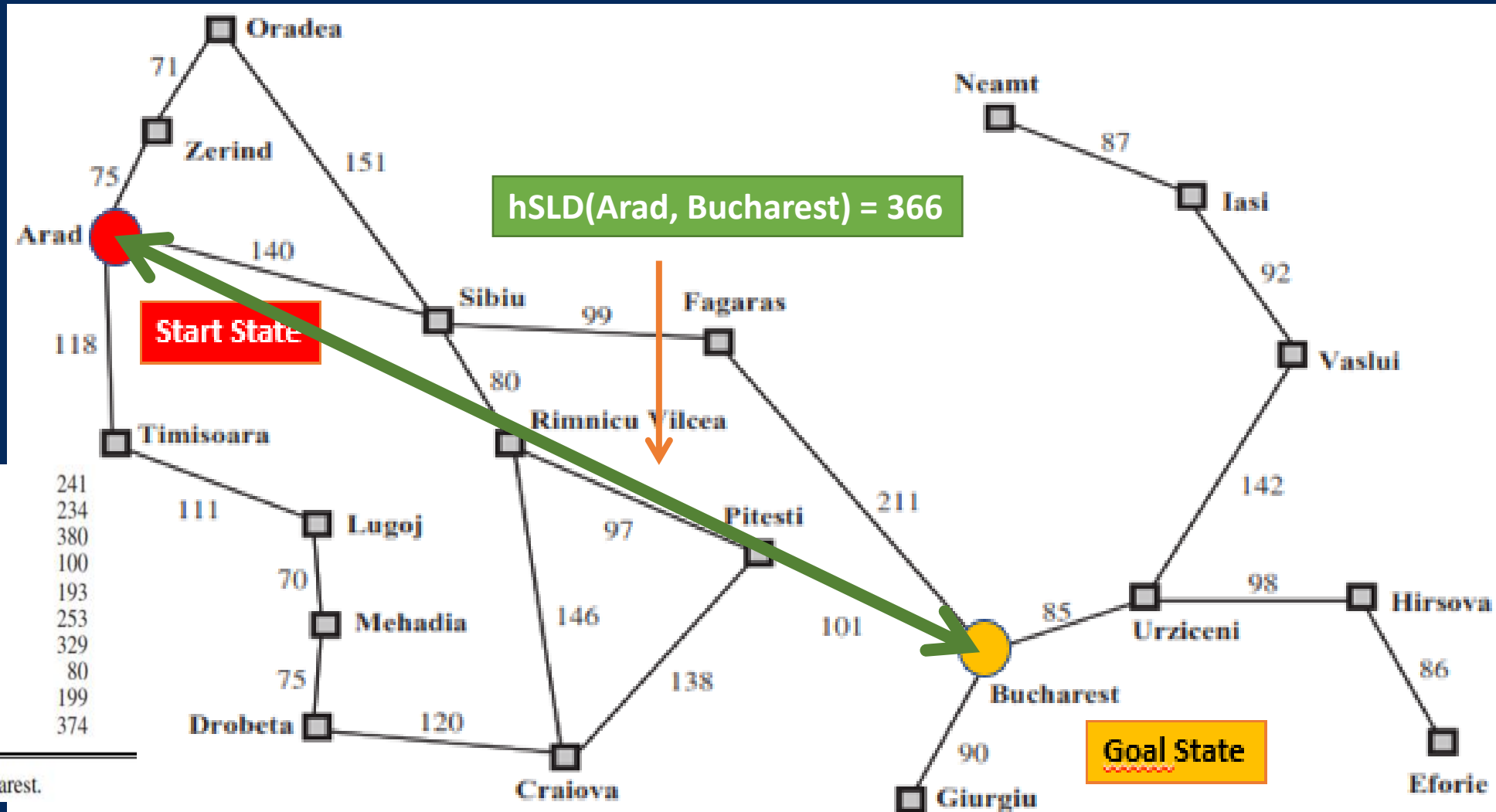


$$f(n) = h(n)$$

Heuristik,
dimana
 $h(n) = h_{SLD}$

Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Drobeta	242	Pitesti	100
Eforie	161	Rimnicu Vilcea	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374

Values of h_{SLD} —straight-line distances to Bucharest.



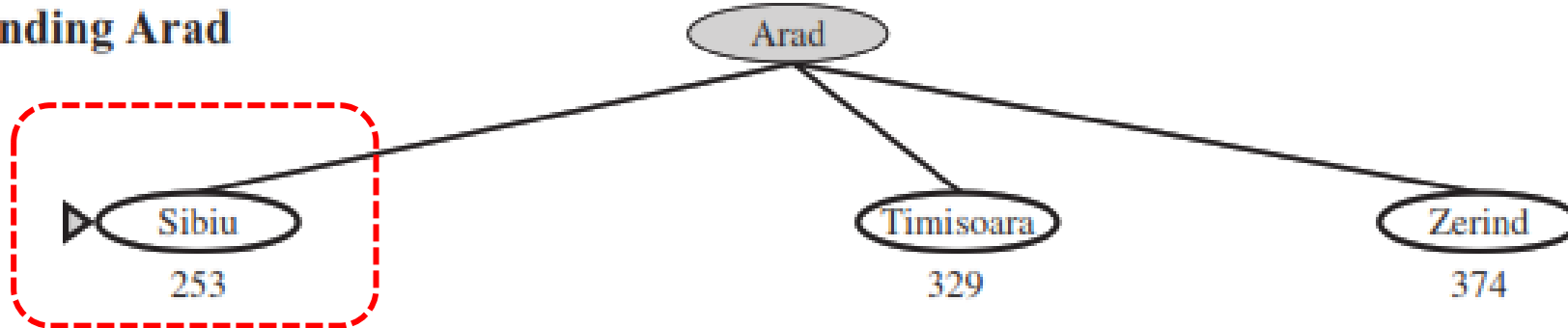


Greedy in Action

(a) The initial state



(b) After expanding Arad



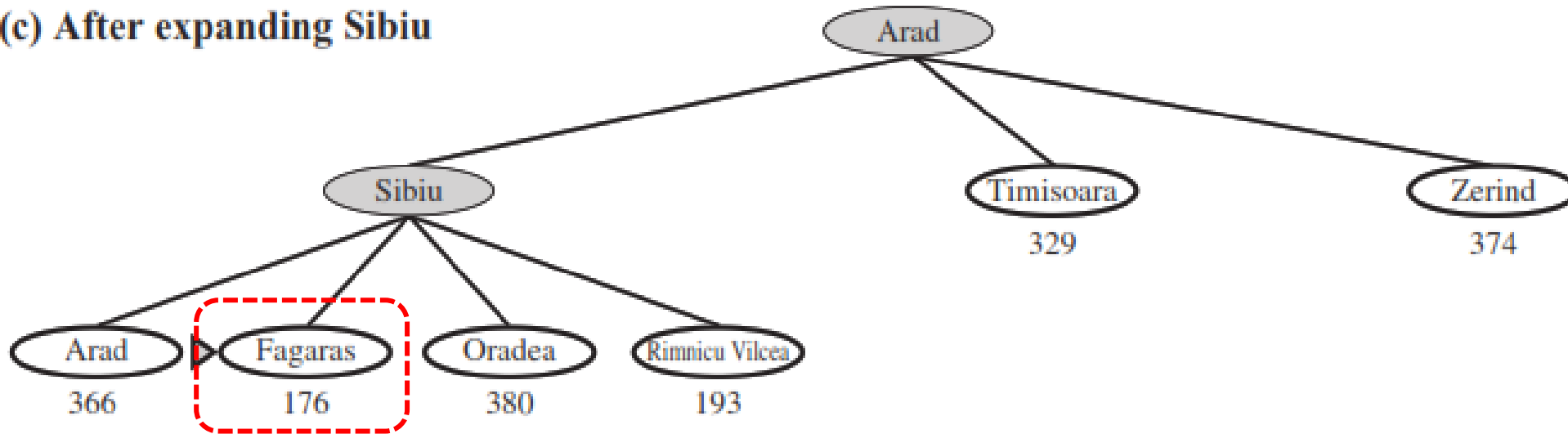
Perhatikan nilai $h(n)$ sebagai dasar Action selanjutnya
 $h(\text{Sibiu}) < h(\text{Timisoara}) < h(\text{Zerind})$





Greedy in Action

(c) After expanding Sibiu

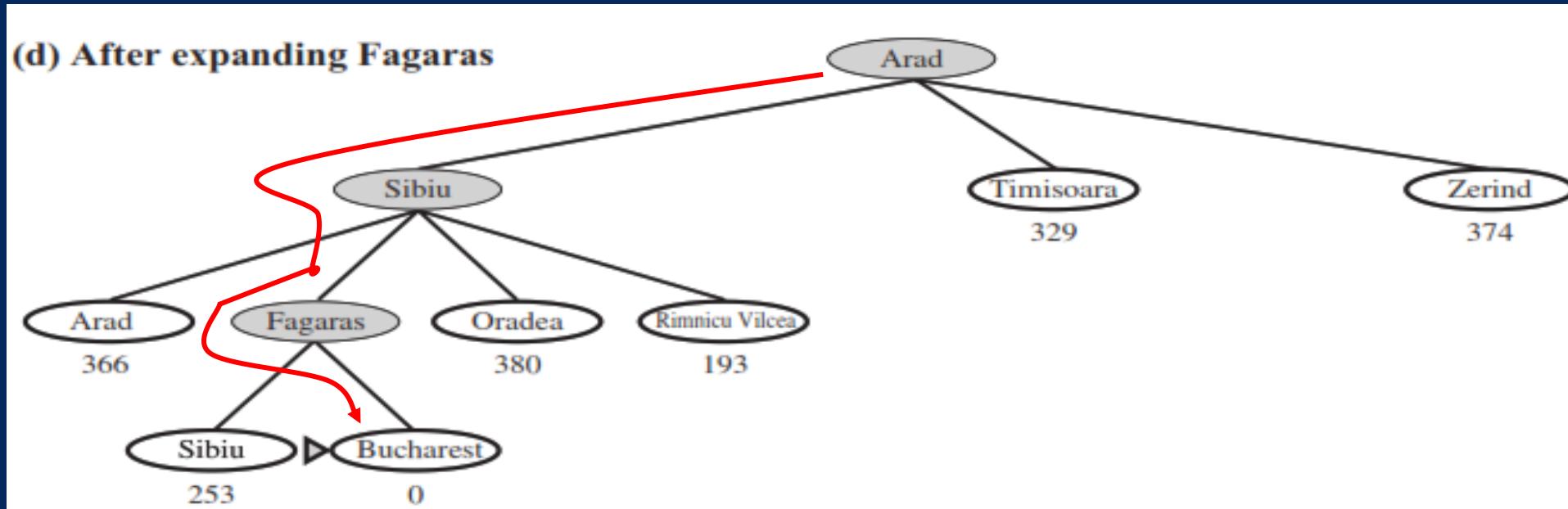


Perhatikan nilai $h(n)$ sebagai dasar Action selanjutnya
 $h(\text{Fagaras}) < h(\text{Rimnicu Vilcea}) < h(\text{Arad}) < h(\text{Oradea})$





Greedy in Action



- Goal state berhasil dicapai
- **Path = {Arad, Sibiu, Fagaras, Bucharest}**
- **Total Cost = 140 + 99 + 211 = 450**





A* Search

- Mengevaluasi simpul-simpul dengan mengombinasikan **$g(n)$** – biaya untuk mencapai simpul, dan **$h(n)$** – biaya untuk mencapai Goal dari simpul, **$f(n) = g(n) + h(n)$**
 - **$g(n)$** = path cost dari start node ke simpul (node) **n**
 - **$h(n)$** = estimated cost dari cheapest path dari simpul **n** ke Goal state

$f(n)$ = estimated cost dari solusi termurah **melalui node n**

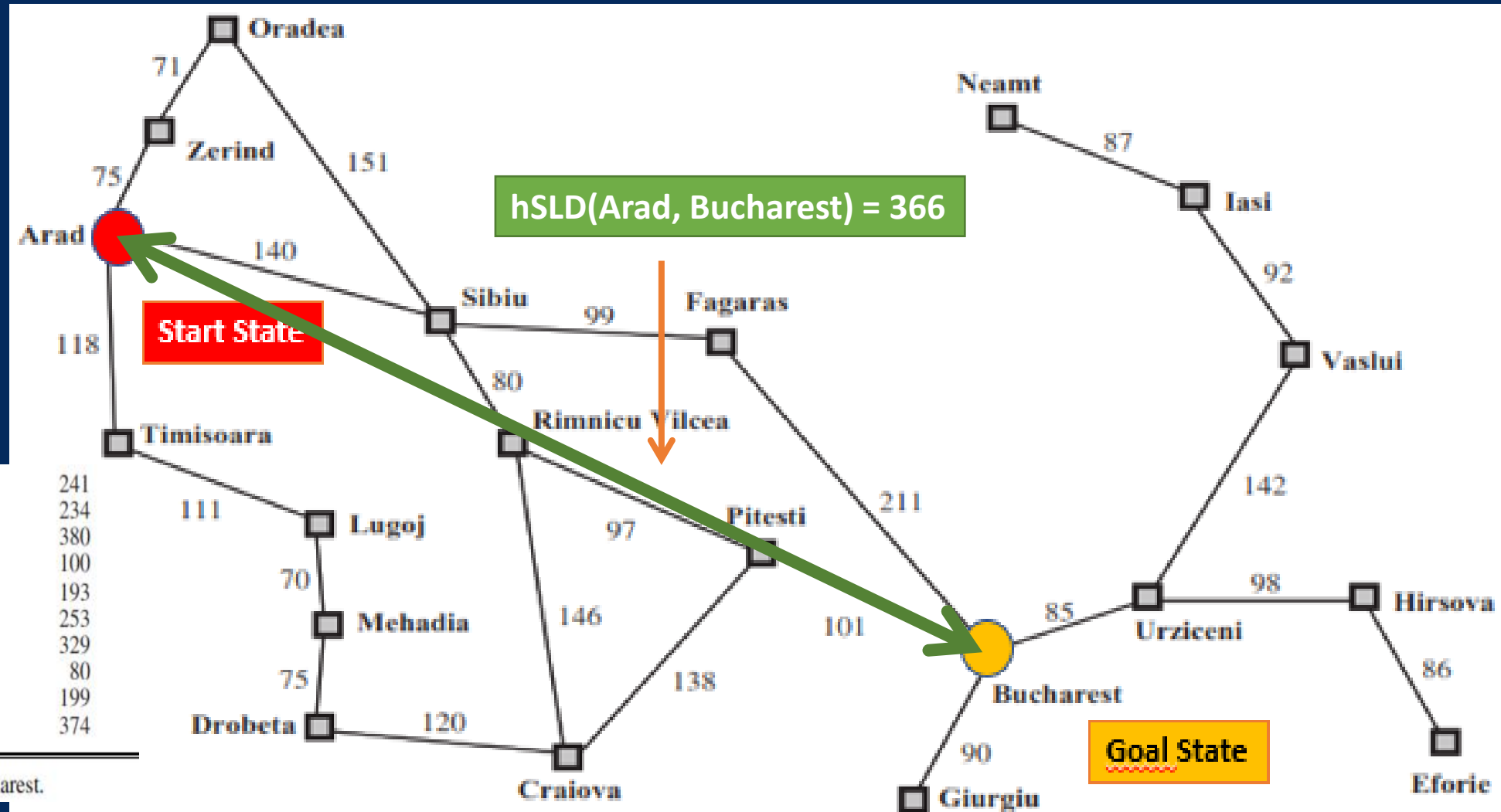




$$f(n) = g(n) + h(n)$$

Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Drobeta	242	Pitesti	100
Eforie	161	Rimnicu Vilcea	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374

Values of h_{SLD} —straight-line distances to Bucharest.



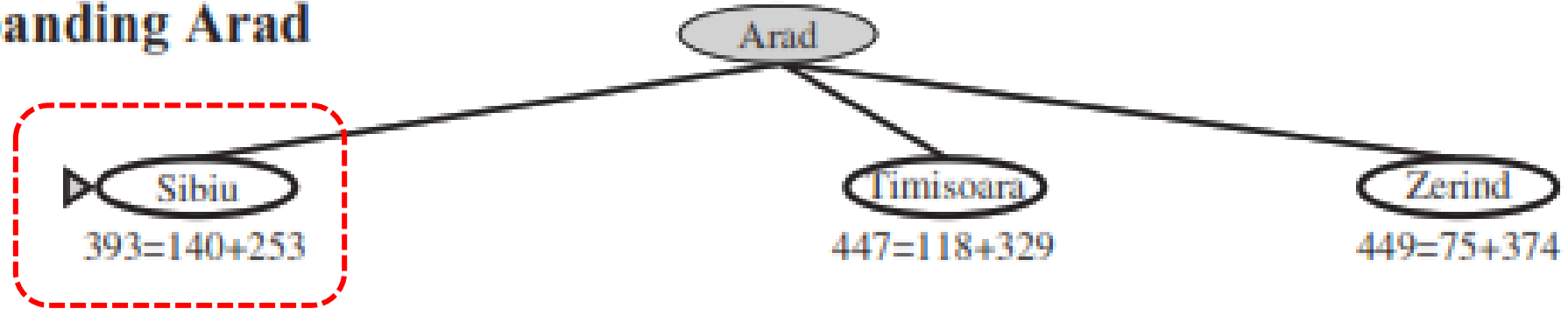


A* Search in Action

(a) The initial state



(b) After expanding Arad



Perhatikan nilai $f(n) = g(n) + h(n)$ sebagai dasar Action selanjutnya

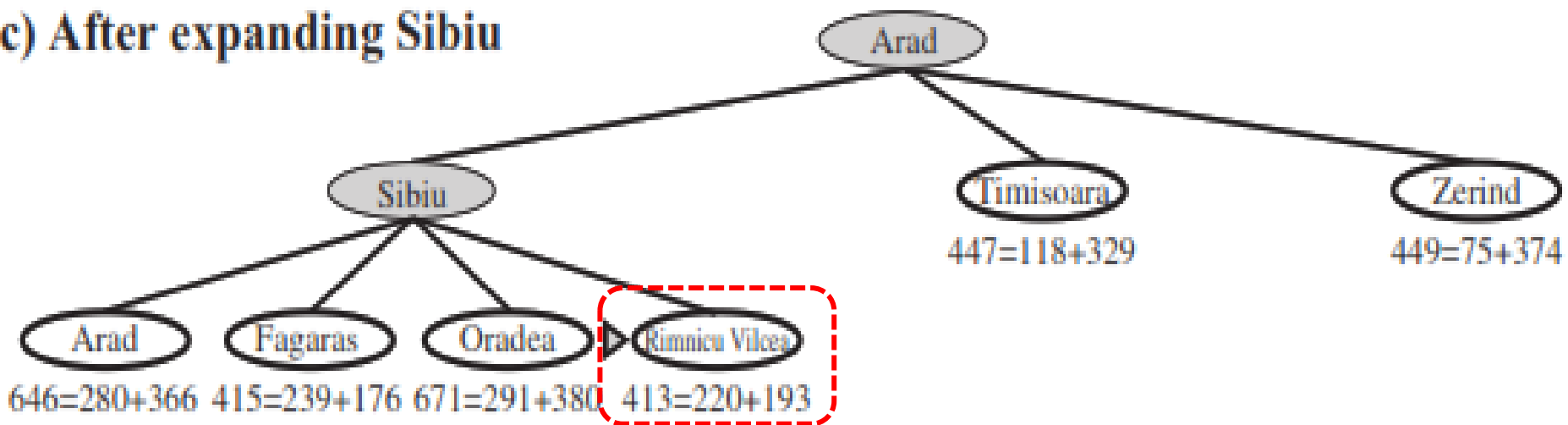
$g(\text{Sibiu}) = 140$, $h(\text{Sibiu}) = 253$, maka $f(\text{Sibiu}) = 393$

$f(\text{Sibiu}) < f(\text{Timisoara}) < f(\text{Zerind})$



A* Search in Action

(c) After expanding Sibiu



Perhatikan nilai $f(n) = g(n) + h(n)$ sebagai dasar Action selanjutnya.

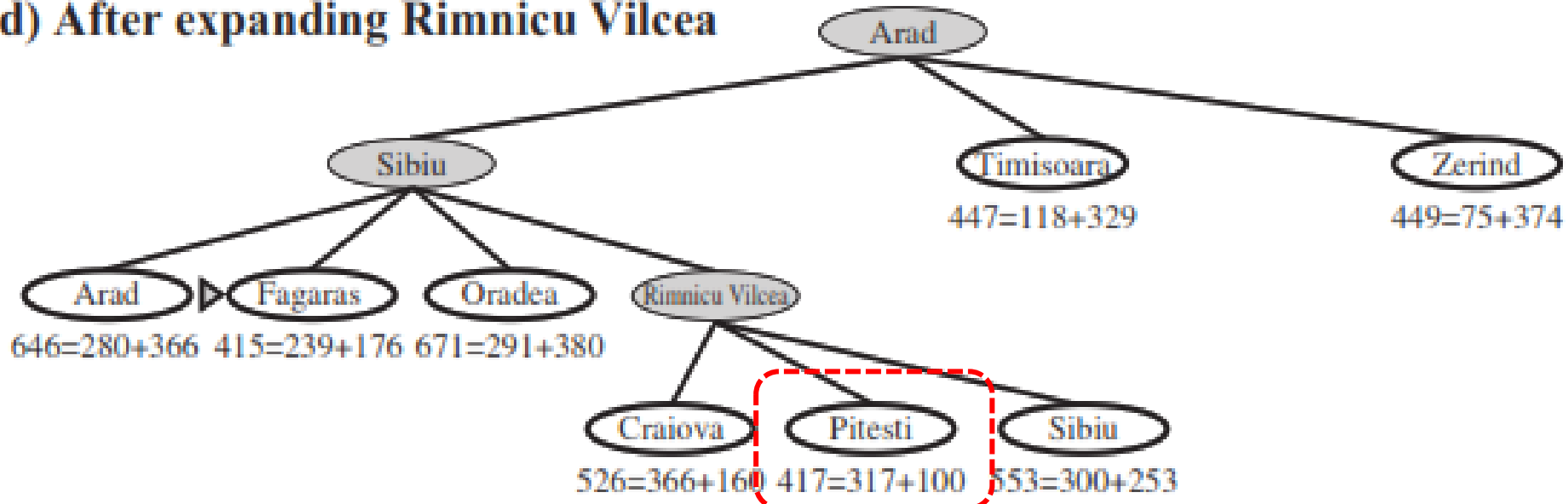
$f(\text{Rimnicu Vilcea}) < f(\text{Fagaras}) < f(\text{Arad}) < f(\text{Oradea})$





A* Search in Action

(d) After expanding Rimnicu Vilcea



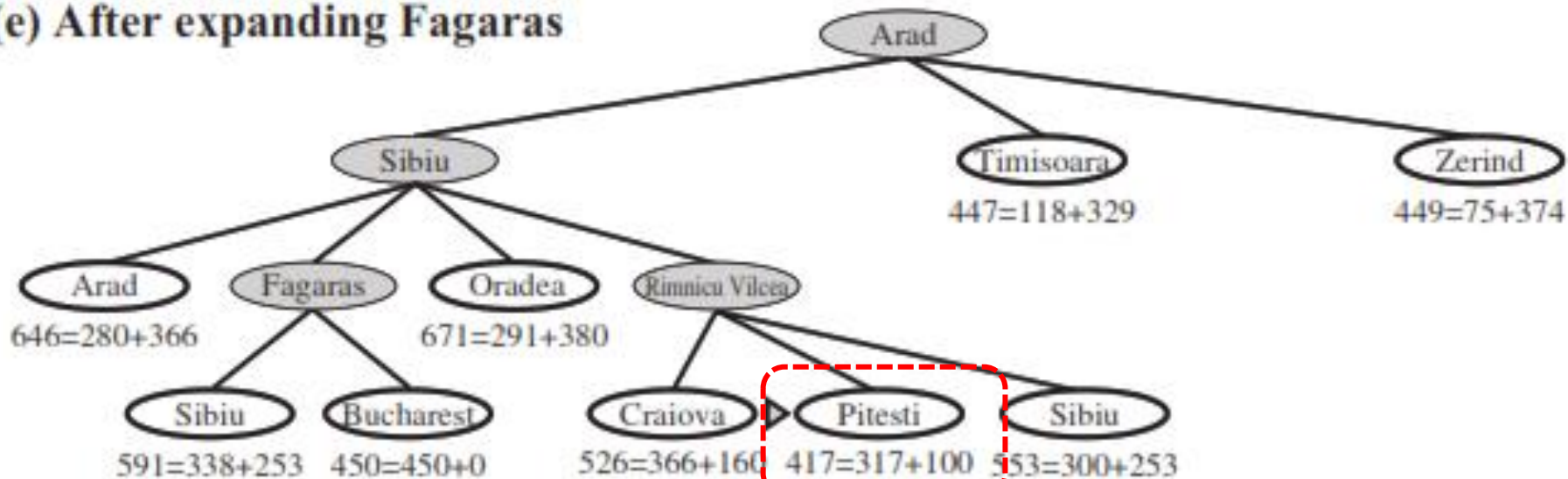
Perhatikan nilai $f(n) = g(n) + h(n)$ sebagai dasar Action selanjutnya.
 $f(\text{Pitesti}) < f(\text{Craiova}) < f(\text{Sibiu})$





A* Search in Action

(e) After expanding Fagaras



Perhatikan nilai $f(n) = g(n) + h(n)$ sebagai dasar Action selanjutnya.

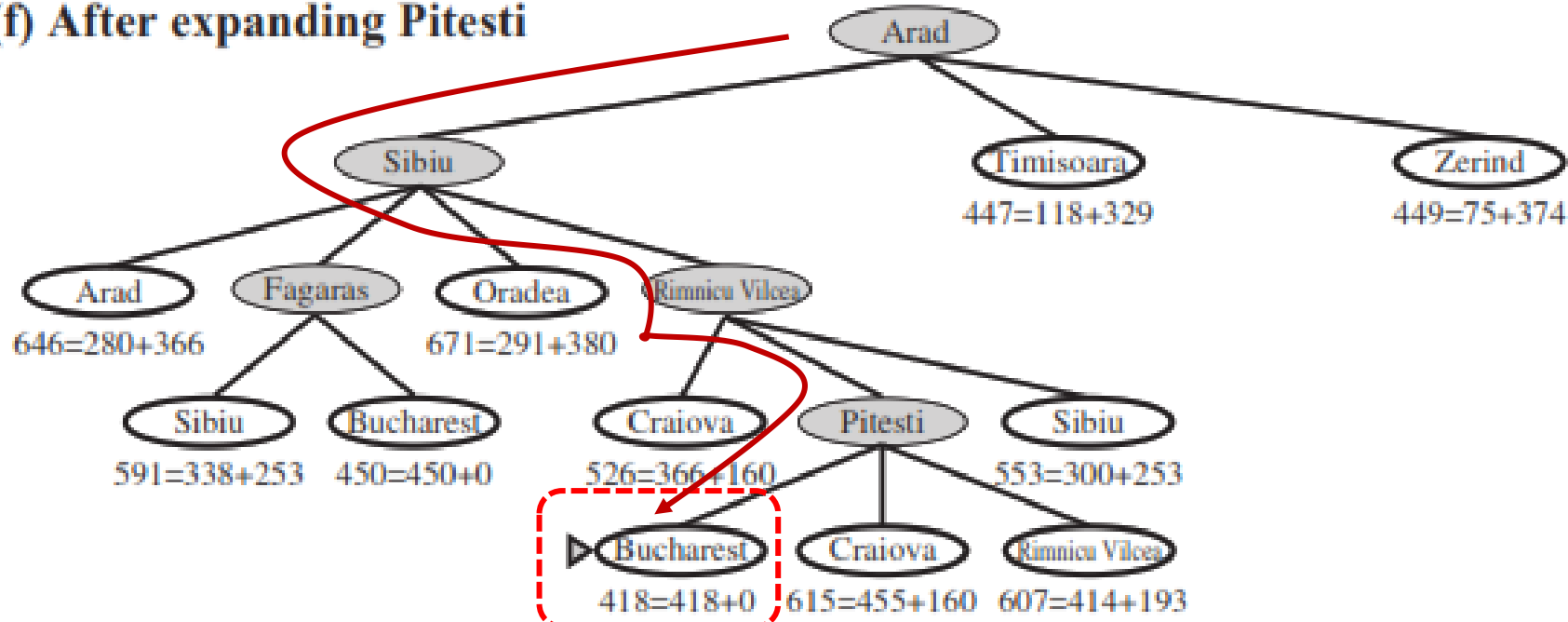
$$f(\text{Pitesti}) < f(\text{Bucharest}) < f(\text{Craiova}) < f(\text{Sibiu})$$





A* Search in Action

(f) After expanding Pitesti



- Goal state berhasil dicapai
- **Path = {Arad, Sibiu, Rimnicu Vilcea, Pitesti, Bucharest}**
- **Path Cost = $140 + 80 + 97 + 101 = 418$**





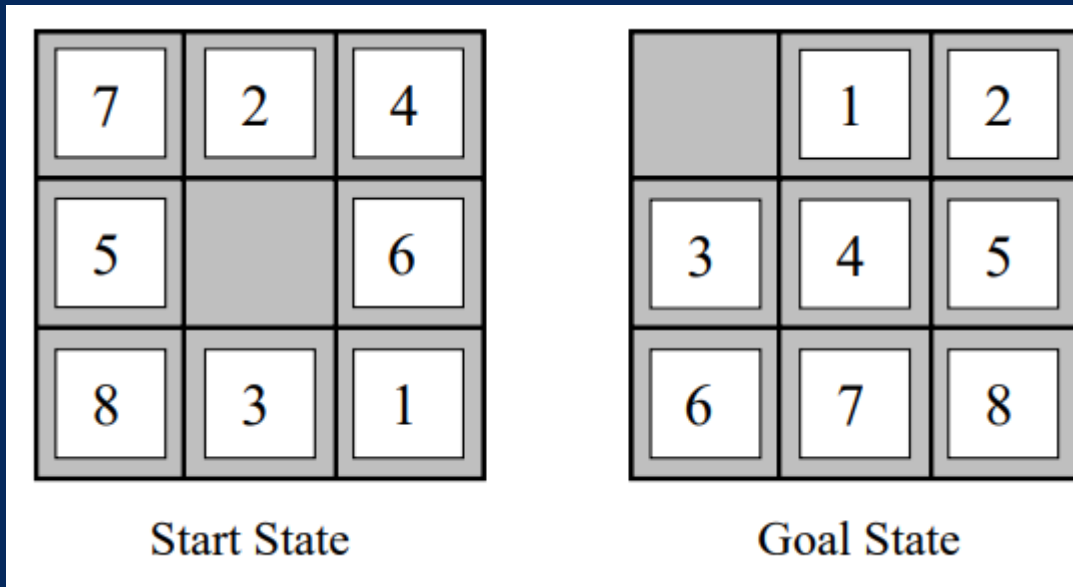
Benchmark Greedy vs. A* Search

- **Greedy**
 - Start -> Goal = 4 simpul
 - Total Cost = 450
 - Efficient cost pada pembukaan simpul berikutnya
 - Bertumpu murni pada **$h(n)$**
- **A* Search**
 - Start -> Goal = 5 simpul
 - Total Cost = 418
 - Cheapest cost pada pembukaan simpul berikutnya
 - Memperhitungkan **$g(n)$**





Fungsi Heuristik



Bagaimana cara memperoleh $h(n)$?





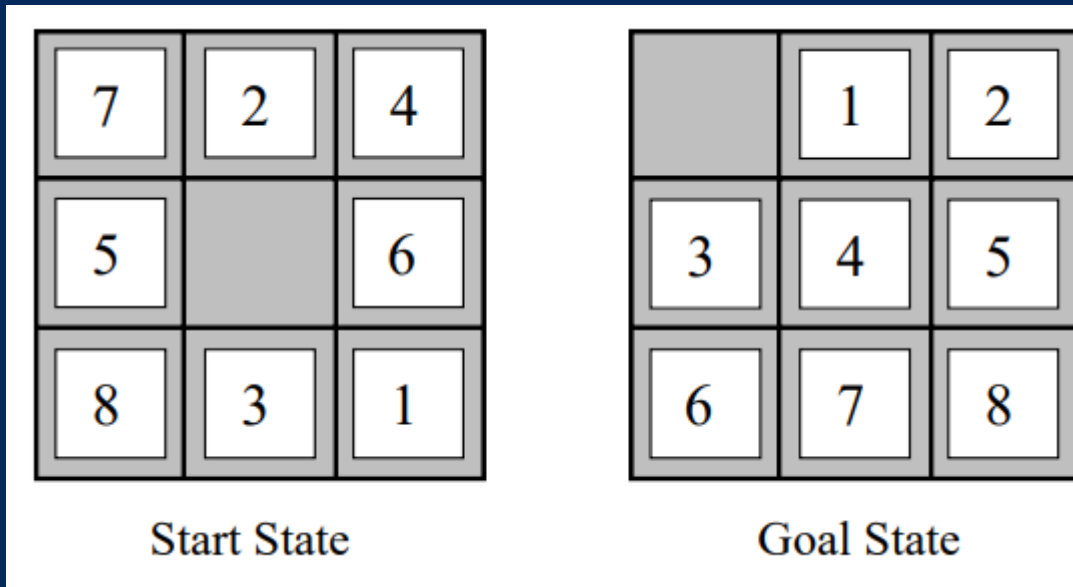
Fungsi Heuristik

- **$h1$ = jumlah ubin yang tidak pada tempatnya** sebagai start awal.
 - Setiap ubin yang tidak pada tempatnya akan digerakkan minimal sekali
- **$h2$ = jumlah jarak total dari semua ubin dari posisi Goal.**
 - Karena ubin tidak dapat digerakkan secara diagonal, maka jarak akan dihitung dari jumlah jarak vertikal dan horisontal ke goal.
 - Teknik ini disebut dengan **city block distance** or **Manhattan Distance**.
 - Semua gerakan adalah menggerakkan ubin satu step mendekat ke goal.





Fungsi Heuristik



- **$h1 = 8$** (setiap ubin minimal digerakkan satu kali)
- **$h2 = 3 + 1 + 2 + 2 + 2 + 3 + 3 + 2 = 18$**





Catatan-catatan

- Metode-metode Informed Search memiliki akses ke fungsi heuristik, $h(n)$ yang mengestimasi biaya dari sebuah solusi dari simpul n
 - Best-First Search algoritma memilih satu simpul yang akan dibuka berdasarkan **sebuah fungsi evaluasi**
-





Catatan-catatan

- Algoritma **Greedy** membuka simpul-simpul yang memiliki **$h(n)$ minimal**. Tidak optimal, namun efisien.
 - Algoritma **A* Search** membuka simpul yang memiliki **cost minimal, $f(n) = g(n) + h(n)$** .
-





Catatan-catatan

- Kinerja algoritma pencarian berdasarkan heuristik bergantung kepada **kualitas fungsi heuristik, $h(n)$** .





Reading Pleasure

- S. J. Russell and P. Norvig, 2016, **Artificial Intelligence: A Modern Approach Third Edition**, Pearson Education Limited
 - Especially **Chapter 3 – Solving Problems by Searching**





Terima Kasih

Kecerdasan Buatan JTI POLINEMA

