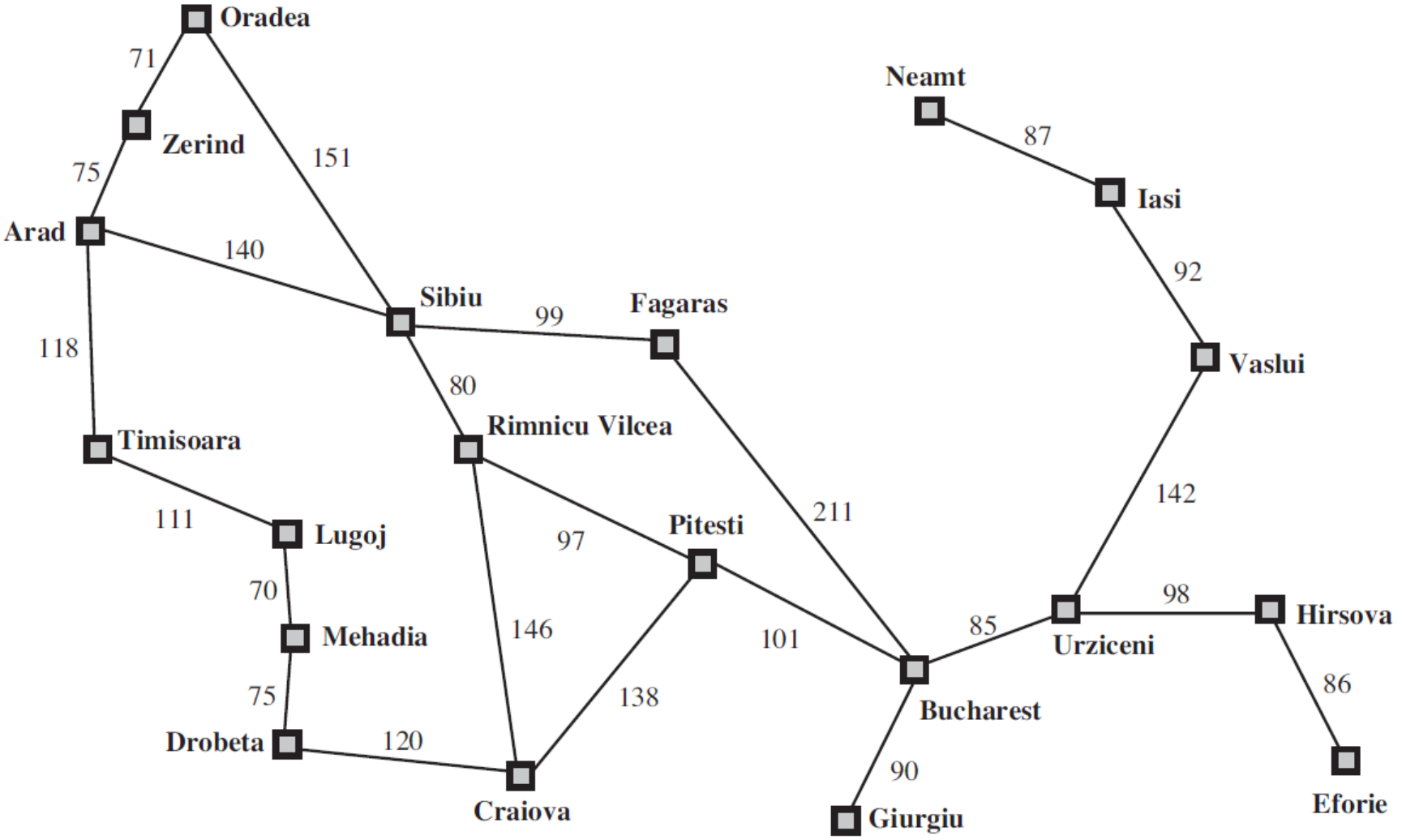
Informed Search

CS161

Prof. Guy Van den Broeck

Best-First Search Motivation

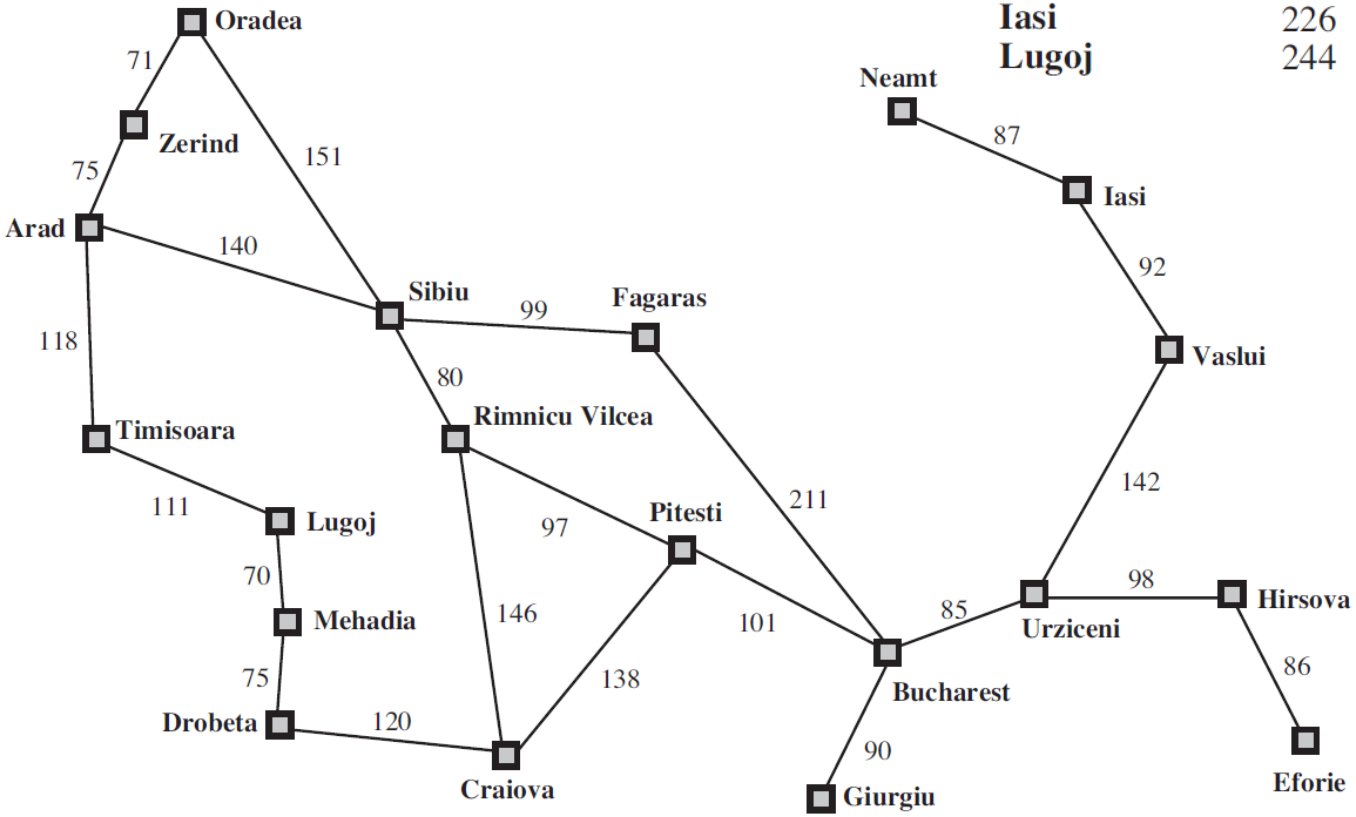
Evaluation function f(n)?



Heuristic Search Motivation

Heuristic function 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



Greedy Best-First Search

- Minimize estimated cost to goal

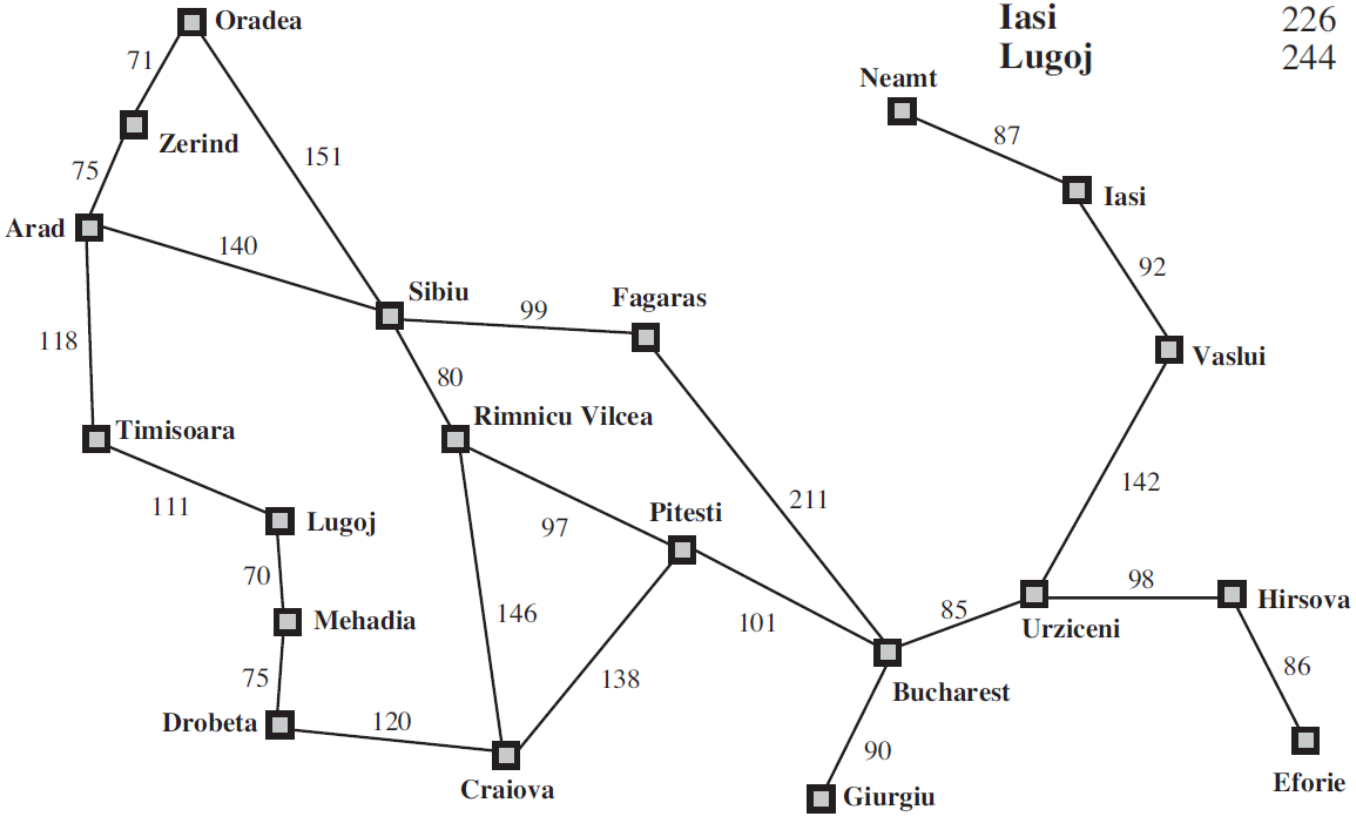
  h(n) estimated cost of cheapest path from n to goal
- Evaluation function to choose node f(n) = h(n)
- Require: h(n) = 0 when n is goal
- Where does h come from?
  - Not easily read from search problem formulation
  - Application-specific

  E.g., straight-line distance from map coordinates

Arad to Bucharest?

Heuristic function 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



Properties

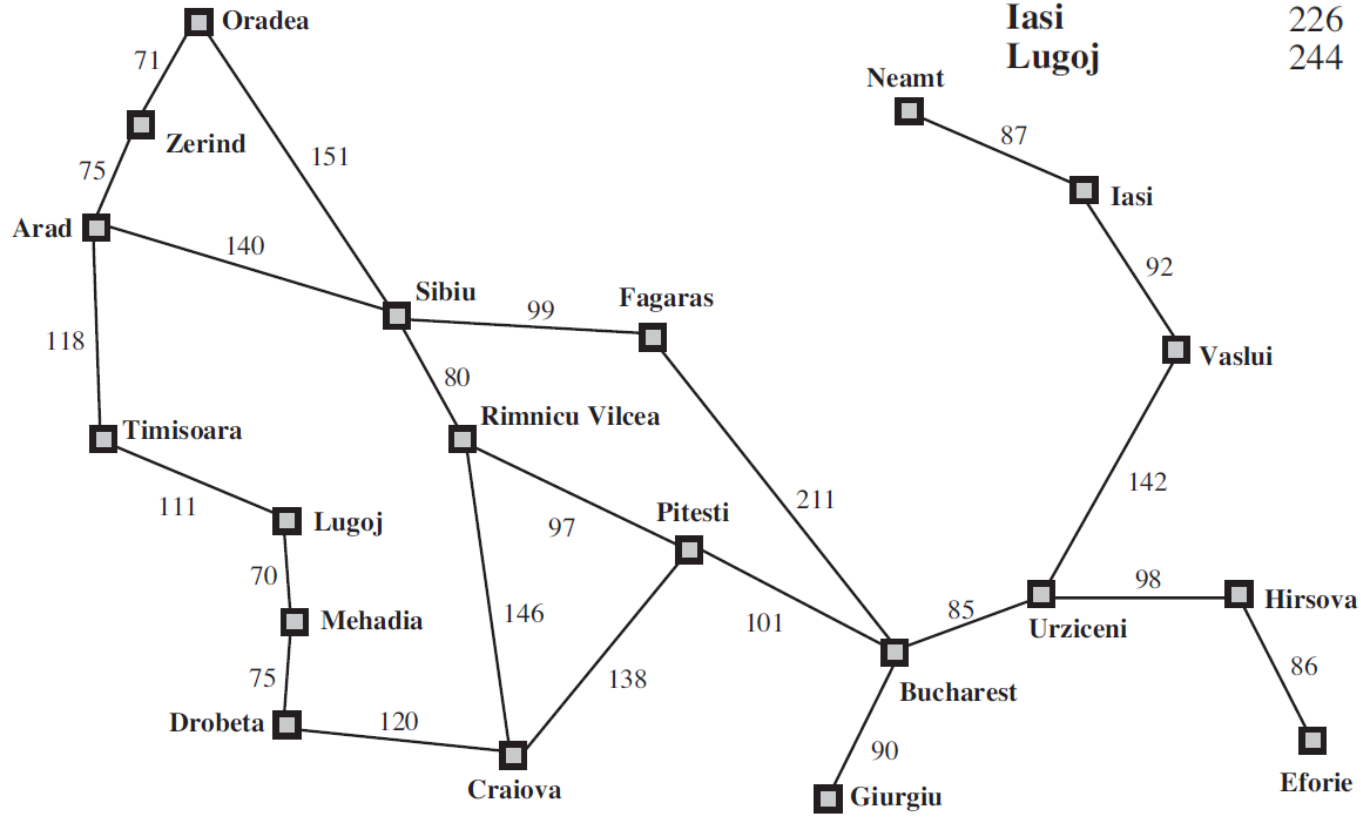
- Arad to Bucharest 32km longer than optimal
- Minimal search cost:

  Solution was found with no unnecessary expansion!
- Optimal? No
- Complete?

Iasi to Fagaras?

Heuristic function 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



Properties

- Arad to Bucharest 32km longer than optimal
- Minimal search cost:

  Solution was found with no unnecessary expansion!
- Optimal? No
- Complete? No
  - Infinite paths
  - Yes if finite state space with graph search
- Time and Space Complexity?

  O(b^m) -- when h(n) = 0 it's blind search!

How to fix this mess?

- Greedy search with heuristic h(n) is too greedy

  no idea where it came from…

- Best-first search with cost g(n) is too conservative

  no idea where the goal is…

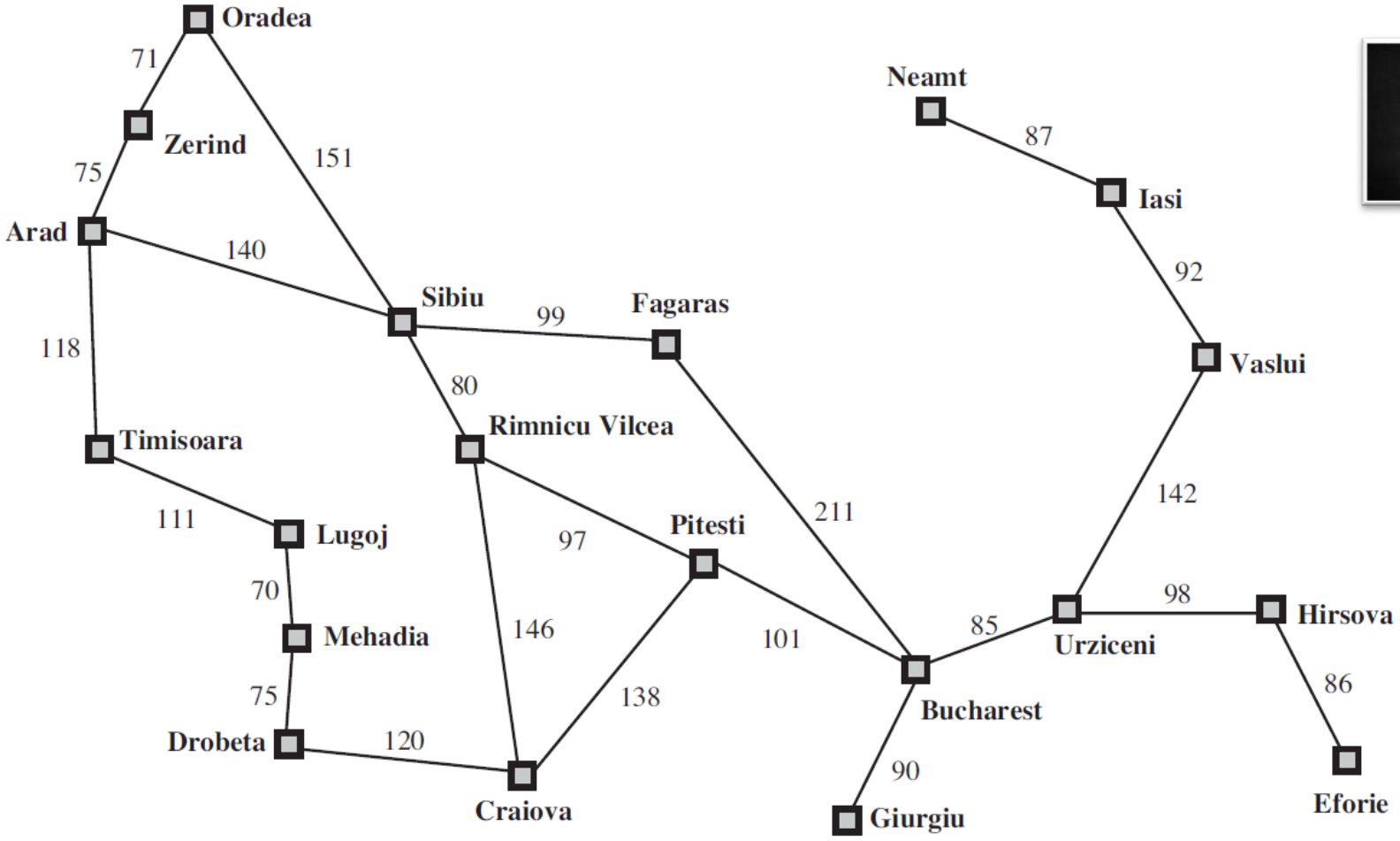
Solution?

A* Search

- f(n) = g(n) + h(n)
  - g(n) = distance from start
  - h(n) = heuristic
- Estimated cost of cheapest solution **through n**

A* Search

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



Heuristic Properties

Optimality of A*

Memory-Bounded Heuristic Search

Constructing Heuristics

Effective Branching Factor

	Search Cost (nodes generated)			Effective Branching Factor		
d	IDS	A*(h_1)	A*(h_2)	IDS	A*(h_1)	A*(h_2)
2	10	6	6	2.45	1.79	1.79
4	112	13	12	2.87	1.48	1.45
6	680	20	18	2.73	1.34	1.30
8	6384	39	25	2.80	1.33	1.24
10	47127	93	39	2.79	1.38	1.22
12	3644035	227	73	2.78	1.42	1.24
14	–	539	113	–	1.44	1.23
16	–	1301	211	–	1.45	1.25
18	–	3056	363	–	1.46	1.26
20	–	7276	676	–	1.47	1.27
22	–	18094	1219	–	1.48	1.28
24	–	39135	1641	–	1.48	1.26

Figure 3.29 Comparison of the search costs and effective branching factors for the ITERATIVE-DEEPENING-SEARCH and A* algorithms with h_1, h_2. Data are averaged over 100 instances of the 8-puzzle for each of various solution lengths d.

Relaxed Problems

Subproblems

Korf's Breakthroughs

- 15-puzzle optimal (1985)
- 24-puzzle optimal (1996)
- Rubik's cube optimal (1997)
- 15-puzzle exhaustive (2005)
- …