Center for LifeLong Learning & Design
University of Colorado at Boulder

Wisdom is not the product of schooling
but the lifelong attempt to acquire it.
- Albert Einstein

Informed Search Methods
—
Chapter 4 in Russell / Norvig Book

Gerhard Fischer

AI Course, Fall 1996, Lecture September 18

Overview

- **objective:** how information about the state space can prevent algorithms from blundering in the dark

- **evaluation function:** determine the desirability or lack thereof of expanding a node (i.e., searching in a specific direction)

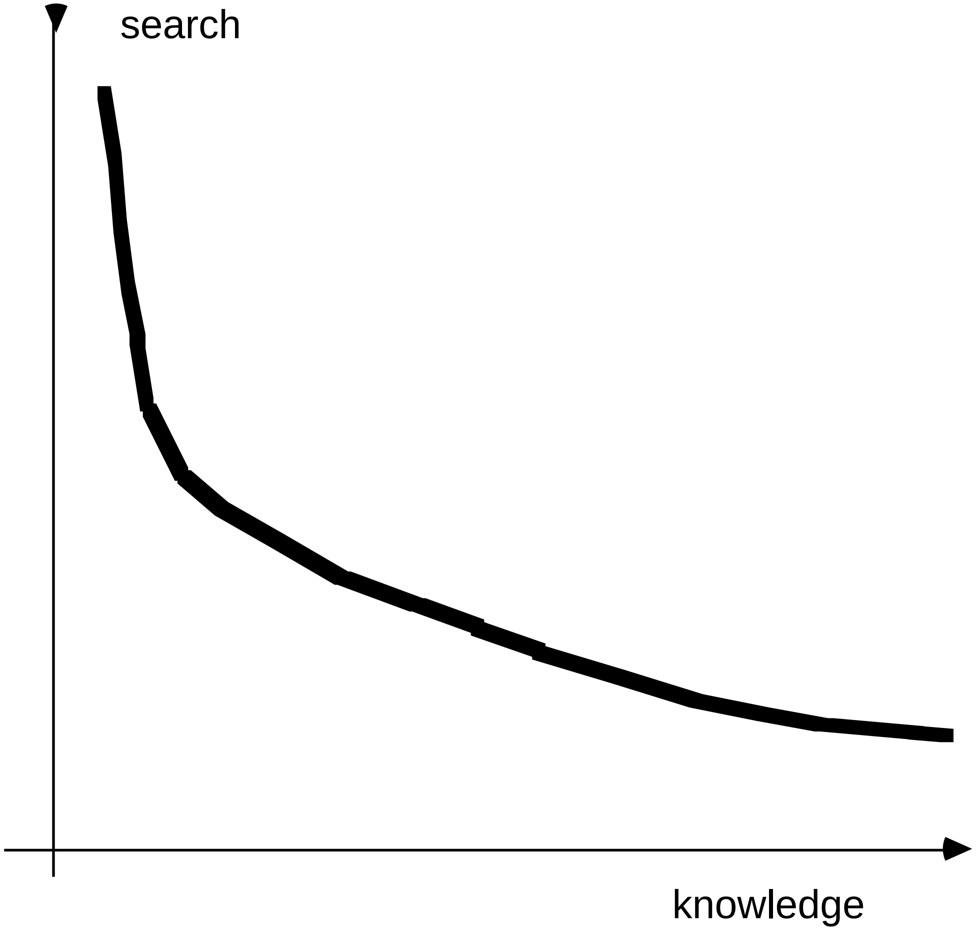
- **best first searches:** expand the most promising node first

- **heuristic functions:**
    - heuristics = study of methods for discovering and inventing problem solving techniques (Polya)
    - heuristics = “rules of thumb” used by domain experts to generate good solutions without exhausitive search (expert systems)

Examples

- mutilated checkerboard
- cryptarithmetic problems
- the defect safe
- route-finding
- 8 puzzle

Search and Knowledge



The Defect Safe Problem

- the task: open a safe whose lock has 10 dials, each with 100 possible settings
- 100^{10} possible settings ---> on average we need to examine half of these = 50 billion billion settings ($50*10^{19}$)
- assume: safe is defect — a dial is heard when any one dial is turned to the correct setting ----> each dial can be adjusted independently
- average trials: 10 * 50 ----> 500
- partial results, intermediate stepping stones reduce the search space tremendously (stable subassemblies are of critical importance in problem solving)

Problem Solving and Search — Cryptarithmetic Problems

```
   DONALD
 + GERALD          D=5
   -------------
   ROBERT
```

(1) D = 5	---->	T = 0
(2) R = 2L + 1	---->	R is odd
(3) R = 5 + G	---->	R > 5 (R = 7 or 9)
(4) O + E = O	---->	E = 0 or 9
(5) T = 0	---->	E = 9 and R = 7
(6) A + A = 9	---->	A = 4 and a carry
(7) 2L = 17	---->	L = 8
(8) 5 + G = 7	---->	carry is there ----> G = 1

trial and error: N = 6, B = 3, O = 2

Best-First Search

- using GENERAL-SEARCH: the only place where knowledge can be applied is in the queuing function which determines the node to expand next

- best-first search ----> “seemingly-best-first search”

Greedy Search: minimize the estimated cost to reach the goal

- a function that calculates such cost estimates: heuristic function h(n) — h(n) can be any function with the requirement h(n) = 0 if n is a goal

- example for the route finding problem: straight-line distance

  H_{SLD} (n) = straight line distance between n and the goal location

```
function GREEDY SEARCH (problem) returns a solution or failure
    return BEST-FIRST-SEARCH (problem,h)
```

- properties of Greedy Search (illustrated by the route-finding problem):
  - susceptible to false starts (Iasi ----> Fagaras)
  - minimal search cost (no node is expanded that is not on the solution path)
  - it is not optimal

A* Search: Minimizing the total Path Cost

- g(n) = uniform-cost search: minimizes the cost of the path so far

- f(n) = g (n) + h (n) -----> estimated cost of the cheapest solution through n

```
function A*-Search (problem) returns a solution or failure
    return BEST-FIRST-SEARCH (problem,g+h)
```

- f(n) is complete and optimal if h is a function which never overestimates the cost to reach the goal ----> such an h is called an **admissible heuristic**

- **example**: H_{SLD} (n) is an admissible heuristic because the shortest path between any two points is a straight line

Heuristic Functions — Example: 8-Puzzle

- **some characteristics:**
  - a typical solution is about 20 steps
  - branching factor is about 3 ----> number of states: $3^{20} = 3.5 \times 10^{9}$ states
  - keeping track of repeated states: 9! = 362,880 different arrangements of 9 squares

- **possible heuristic functions:**
  - h1 = number of tiles in wrong position
  - h2 = sum of the distances of the tiles from their goal position

- **the big question:** how can we invent heuristic functions?

- **relaxed problems:**
  - a problem with less restrictions on the operators
  - good strategy: the cost of an exact solution to a relaxed problem is a good heuristic for the original problem