Foundations of Artificial Intelligence

CS472/3 — Fall 1999
Lecture #11

Bart Selman

Slide CS472–1

Today's Lecture

Deep Blue.

Hardness of Search Problems

Synopsis of search.

Slide CS472–2

Case Study: Deep Blue

See separate sheets.

Slide CS472–3

Hardness of Search Problems

See separate sheets.

Slide CS472–4

Concludes "Search"

- **Problem Solving as Search**
- **Uninformed search:** DFS / BFS / Uniform cost search
  time / space complexity
  size search space: up to approx. 10^{11} nodes
  special case: **Constraint Satisfaction / CSPs**
    generic framework: variables & constraints
    backtrack search (DFS); propagation (forward-checking / arc-consistency, variable / value ordering
    (but **incomplete**; N-queens example)

Slide CS472–5

- **Informed search:** use heuristic function guide to goal
  Greedy search
  A search / provably optimal*
    Search space up to approx. 10^{25}
  Local search (incomplete)
    Greedy / hillclimbing / GSAT
    Simulated annealing
    Tabu search
    Genetic Algorithms / Genetic Programming
    search space 10^{100} to 10^{1000}

Slide CS472–6

- **Aversary search / game playing**

  Minimax

  Up to around 10^{10} nodes, 6 — 7 ply in chess.

  alpha-beta pruning

  Up to around 10^{21} nodes, 14 ply in chess.

  provably optimal
- **Hardness or Search Problems**

  Hardest problems: **critically constrained**

  at *phase transition from solvable to unsolvable.*

Slide CS472–7

Search and AI

Why such a central role?

A. Basically, because lots of task in AI are **intractable**.
Search is "only" way to handle them.

Many applications of search, in e.g.,:
Learning / Reasoning / Planning / NLU / Vision.

Good thing: much recent progress (10^{30} quite feasible;
sometimes up to 10^{1000}. **Qualitative difference from only a few years ago!**

Slide CS472–8