

Heuristic Search

CSPs

1 handout: slides
asst 1 milestone was due

EOLQs

Heuristic Search

CSPs

Heuristic Search

- Comparison
- Heuristics
- Other Algorithms
- Search Algorithms
- Take 2

CSPs

Heuristic Search

Comparison

Heuristic Search

Comparison

- Heuristics
- Other Algorithms
- Search Algorithms
- Take 2

CSPs

Algorithm	Time	Space	Complete	Admissible
Depth-first	b^m	bm	If $m \geq d$	No
Breadth-first	b^d	b^d	Yes	if ops cost 1
Uniform-cost	b^d	b^d	Yes	Yes
IDDFS	b^d	bd	Yes	Yes
Greedy			depends	No
A*	depends	\approx time	if h adm.	if h adm.
branching factor	b			
maximum depth	m			
solution depth	d			

Simplified problem must give lower bound on original!

1. Relaxation: fewer and/or weaker constraints
 - Sometime efficient closed form
2. Abstraction: simplify token identity
 - Smaller search space

Want highest value

- If $h_1(n) \leq h_2(n)$ for all n , h_2 dominates h_1

Other Shortest-path Algorithms

Heuristic Search

■ Comparison

■ Heuristics

■ Other Algorithms

■ Search Algorithms

■ Take 2

CSPs

- IDA*
- SMA*, IE
- RBFS
- RTA*, LRTA*

Course projects!

Search Algorithms

Heuristic Search

- Comparison
- Heuristics
- Other Algorithms
- Search Algorithms
- Take 2

CSPs

- Uninformed methods
 - ◆ Depth-first
 - ◆ Breadth-first
 - ◆ Uniform-cost
- Informed methods
 - ◆ Greedy
 - ◆ A*
- Bounding memory
 - ◆ Iterative deepening
 - ◆ Beam search

Search Algorithms, Take 2

Heuristic Search

- Comparison
- Heuristics
- Other Algorithms
- Search Algorithms
- Take 2

CSPs

- Depth-first
 - ◆ Depth-first
- 'Best-first'
 - ◆ Breadth-first
 - ◆ Uniform-cost
 - ◆ Greedy
 - ◆ A*
- Bounding memory
 - ◆ Iterative deepening
 - ◆ Beam search

Heuristic Search

CSPs

- Other Problems
- Types of Problems
- Backtracking
- Break
- Forward Checking
- 'Heuristics'
- Example Results
- MAC
- Other Algorithms
- EOLQs

Constraint Satisfaction Problems

Some Other Search Problems

Heuristic Search

CSPs

■ Other Problems

- Types of Problems
- Backtracking
- Break
- Forward Checking
- 'Heuristics'
- Example Results
- MAC
- Other Algorithms
- EOLQs

Map coloring: Given a map of n countries and a set of k colors, color every country differently from its neighbors.

n -queens : Given an $n \times n$ chessboard, arrange n queens so that none is attacking another.

What algorithm would you use?

Types of Search Problems

Heuristic Search

CSPs

■ Other Problems

■ Types of Problems

■ Backtracking

■ Break

■ Forward Checking

■ 'Heuristics'

■ Example Results

■ MAC

■ Other Algorithms

■ EOLQs

- Shortest-path (vacuum, tile puzzle, M&C)
 - ◆ given operators and their costs
 - ◆ want least-cost path to a goal
 - ◆ goal depth/cost unknown
- Constraint satisfaction (map coloring, n -queens)
 - ◆ any goal is fine
 - ◆ fixed depth
 - ◆ explicit constraints on partial solutions

Chronological Backtracking

Do not expand any partial solution that violates a constraint.

Heuristic Search

CSPs

- Other Problems
- Types of Problems
- Backtracking
- Break
- Forward Checking
- 'Heuristics'
- Example Results
- MAC
- Other Algorithms
- EOLQs

Break

Heuristic Search

CSPs

- Other Problems
- Types of Problems
- Backtracking
- Break
- Forward Checking
- 'Heuristics'
- Example Results
- MAC
- Other Algorithms
- EOLQs

- office hours
- asst 1
 - ◆ milestone:
 - ◆ you know it all now
 - ◆ don't forget: beauty, write-up
- blog entries due on Wednesdays, 8am, 400 words
- projects (Apr 2), UROP (Mar 1)

Forward Checking

Heuristic Search

CSPs

- Other Problems
- Types of Problems
- Backtracking
- Break
- Forward Checking
- 'Heuristics'
- Example Results
- MAC
- Other Algorithms
- EOLQs

When assigning a variable, remove the conflicting values for all connected variables. Backtrack on domain wipeout.

Arc consistency: for every value in the domain of x , there exists a value in the domain of y that satisfies all the constraints.

Heuristics for CSPs

Heuristic Search

CSPs

- Other Problems
- Types of Problems
- Backtracking
- Break
- Forward Checking
- 'Heuristics'
- Example Results
- MAC
- Other Algorithms
- EOLQs

Variable choice: choose most constrained variable (smallest domain)

- want to keep tree small, failing quickly

Value choice: try least constraining value first (fewest removals)

- might as well succeed sooner if possible

Example Results

Heuristic Search

CSPs

- Other Problems
- Types of Problems
- Backtracking
- Break
- Forward Checking
- 'Heuristics'
- Example Results
- MAC
- Other Algorithms
- EOLQs

	BT	FC	FC+MCV
USA	> 1M	2K	60
n-Queens	> 40M	> 40M	820K
Zebra	3.9M	35K	500
Random 1	420K	26K	2K
Random 2	940K	77K	15K

Maintaining Arc Consistency

Heuristic Search

CSPs

- Other Problems
- Types of Problems
- Backtracking
- Break
- Forward Checking
- 'Heuristics'
- Example Results
- **MAC**
- Other Algorithms
- EOLQs

Ensure every value for x has a legal value in all neighbors y . If one doesn't, remove it and ensure consistency of all y .

Maintaining Arc Consistency

Heuristic Search

CSPs

- Other Problems
- Types of Problems
- Backtracking
- Break
- Forward Checking
- 'Heuristics'
- Example Results
- **MAC**
- Other Algorithms
- EOLQs

Ensure every value for x has a legal value in all neighbors y . If one doesn't, remove it and ensure consistency of all y .

while Q is not empty

$(x, y) \leftarrow \text{pop } Q$

if **revised**(x, y) then

if x 's domain is now empty, return failure

for every other neighbor z of x

push (z, x) on Q

revised(x, y)

revised \leftarrow false

foreach v in x 's domain

if no value in domain of y is compatible with v

remove v from x 's domain

revised \leftarrow true

return *revised*

Other Algorithms for CSPs

Heuristic Search

CSPs

- Other Problems
- Types of Problems
- Backtracking
- Break
- Forward Checking
- 'Heuristics'
- Example Results
- MAC
- Other Algorithms
- EOLQs

- (Conflict-directed) Backjumping
- Dynamic backtracking
- Randomized restarting

Course projects!

Heuristic Search

CSPs

- Other Problems
- Types of Problems
- Backtracking
- Break
- Forward Checking
- 'Heuristics'
- Example Results
- MAC
- Other Algorithms

■ EOLQs

Please write down the most pressing question you have about the course material covered so far and put it in the box on your way out.

Thanks!