

AI Experimental Course

Reporter: Zhaoshuai Liu

School of Data and Computer Science
Sun Yat-sen University

24747380@qq.com

August 28, 2019



1 Search

- Uninformed Search
- Informed (Heuristic) Search
- Adversarial Search
- Constraint Satisfaction Problems (CSPs)
- Typical Problems

2 Prolog

- Typical Problems

3 FF Planning System

- Typical Problems

4 Machine Learning

- Typical Algorithms



1 Search

- Uninformed Search
- Informed (Heuristic) Search
- Adversarial Search
- Constraint Satisfaction Problems (CSPs)
- Typical Problems

2 Prolog

- Typical Problems

3 FF Planning System

- Typical Problems

4 Machine Learning

- Typical Algorithms



Uninformed Search

- DFS
- BFS
- Uniform-cost search
- Depth-limited search
- Iterative-Deepening search

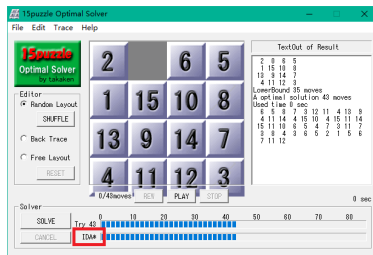
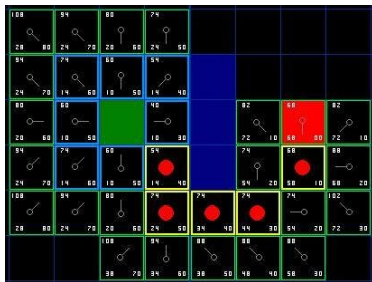
function ITERATIVE-DEEPENING-SEARCH(*problem*) **returns** a solution, or failure
 for *depth* = 0 **to** ∞ **do**
 result \leftarrow DEPTH-LIMITED-SEARCH(*problem*, *depth*)
 if *result* \neq cutoff **then return** *result*

- Bidirectional search



Informed (Heuristic) Search

- Greedy best-first search
- A* (<https://www.redblobgames.com/pathfinding/a-star/introduction.html>)
 - $f(n) = g(n) + h(n)$
 - $f(n)$ is the estimated cost of the cheapest solution through n
 - $g(n)$ is the path cost from the start node to node n
 - $h(n)$ is the estimated cost of the cheapest path from n to the goal
- Iterative-deepening A* (IDA*)



Adversarial Search

- The minimax algorithm
- $\alpha - \beta$ pruning

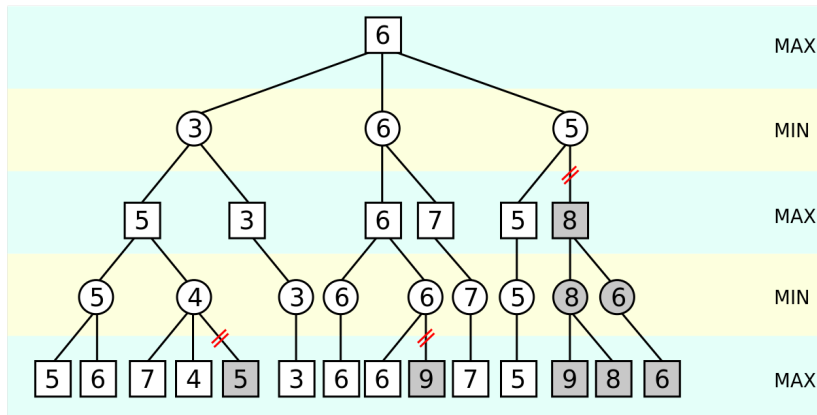


Figure 1: $\alpha - \beta$ pruning

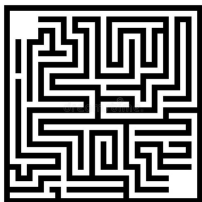


Constraint Satisfaction Problems (CSPs)

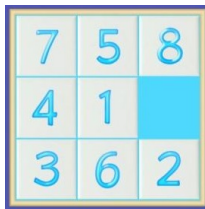
- Backtracking Search for CSPs
- Forward Checking (FC)
- Generalized Arc Consistency (GAC)



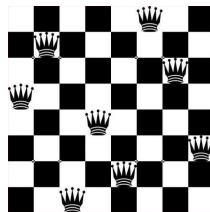
Typical Problems



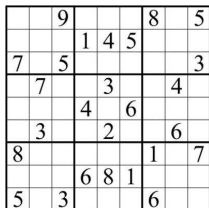
(a) Maze



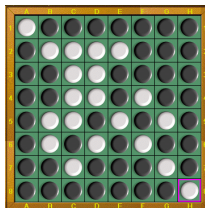
(b) 8 Puzzle



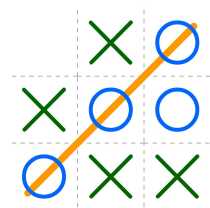
(c) 8 queens puzzle



(d) Sudoku



(e) Othello



(f) Tic Tac Toe

Figure 2: Typical Problems

1 Search

- Uninformed Search
- Informed (Heuristic) Search
- Adversarial Search
- Constraint Satisfaction Problems (CSPs)
- Typical Problems

2 Prolog

- Typical Problems

3 FF Planning System

- Typical Problems

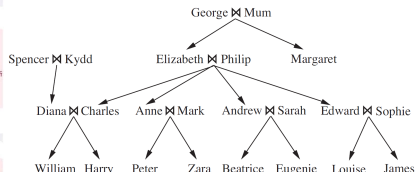
4 Machine Learning

- Typical Algorithms



Typical Problems

- Search Problems
- KR (e.g. Family Problem)
- Queries on KB (Similar to SQL)
- CSPs
 - Sudoku Problem
 - Eight Queens Problem
 - Other Games



Examples

Example 1 (Family Problem)

grandChild(A,B):-child(A,C),child(C,B).

aunt(A,B):-child(B,C),sister(A,C).

...

male('George').

child('Elizabeth','George').

...

Example 2 (Tower of Hanoi)

hanoi(N):-move(N,a,b,c).

move(1,A,_,C):-inform(A,C).

move(N,A,B,C):-N1 is

N-1,move(N1,A,C,B),inform(A,C),move(N1,B,A,C).

inform(Loc1,Loc2):-nl,write('from '),write(Loc1),write(' to '),write(Loc2).



1 Search

- Uninformed Search
- Informed (Heuristic) Search
- Adversarial Search
- Constraint Satisfaction Problems (CSPs)
- Typical Problems

2 Prolog

- Typical Problems

3 FF Planning System

- Typical Problems

4 Machine Learning

- Typical Algorithms



Typical Problems

- Blocks Problem
- Logistic Problem
- 8-puzzle Problem
- Freecell Game
- Boxman Game

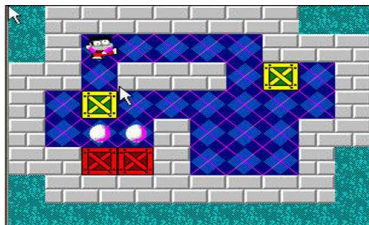


Figure 3: Freecell Game and Boxman Game



Examples

Spare Tire (domain)

```
(:action Remove
:parameters (?x - physob ?y - location)
:precondition (At ?x ?y)
:effect (and (not (At ?x ?y)) (At ?x Ground)))

(:action PutOn
:parameters (?x - physob)
:precondition (and (Tire ?x) (At ?x Ground)
(not (At Flat Axle)))
:effect (and (not (At ?x Ground)) (At ?x Axle)))
```

Spare Tire (data)

```
(define (problem prob)
(:init (Tire Flat)(Tire Spare)(At Flat Axle)(At Spare Trunk))
(:goal (At Spare Axle))
)
```

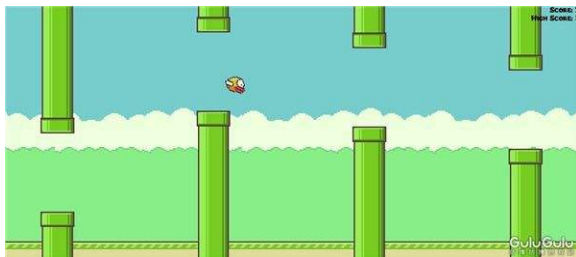


- 1 Search
 - Uninformed Search
 - Informed (Heuristic) Search
 - Adversarial Search
 - Constraint Satisfaction Problems (CSPs)
 - Typical Problems
- 2 Prolog
 - Typical Problems
- 3 FF Planning System
 - Typical Problems
- 4 Machine Learning
 - Typical Algorithms



Typical Algorithms

- Probabilistic Reasoning (Bayesian Network)
- Decision Tree (ID3 and C4.5)
- Naive Bayes
- EM Clustering
- BP-Neural Network
- Deep Learning
- Reinforcement Learning (Q learning and Sarsa)
- Deep Q-Learning (DQN)



The End

