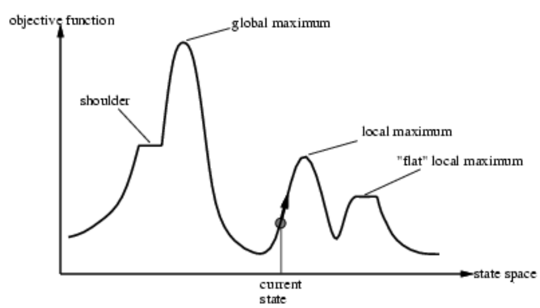
CSE 473

Chaps 4 & 6

Local Search and Games

Last Time: Local Search Algorithms

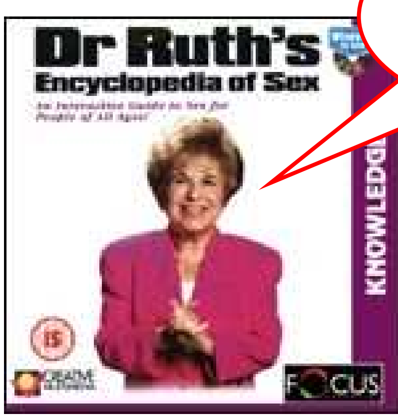


- Hill Climbing
  - Stochastic hill climbing
  - Random restart
- Simulated Annealing
- Local Beam Search

Recall: Local Beam Search

- Keep track of *k* states rather than just one
- Start with *k* randomly generated states
- At each iteration, all the successors of all *k* states are generated
- If any one is a goal state, stop; else select the *k* best successors from the complete list and repeat.

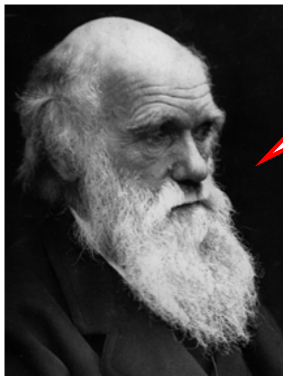
3

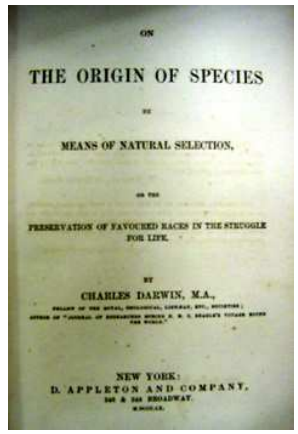


Hey, perhaps sex can improve search?

4

Sure, venerable lady
– check out my
yonder book.





THE ORIGIN OF SPECIES

MEANS OF NATURAL SELECTION,

PRESERVATION OF FAVOURED RACES IN THE STRUGGLE
FOR LIFE.

CHARLES DARWIN, M.A.,

NEW YORK:
D. APPLETON AND COMPANY,

5

Genetic Algorithms

- *A successor state is generated by combining two parent states*
- Start with *k* randomly generated states (population)
- *A state is represented as a string over a finite alphabet (often a string of 0s and 1s)*
- Evaluation function (fitness function). Higher values for better states.
- Produce the next generation of states by selection, crossover, and mutation

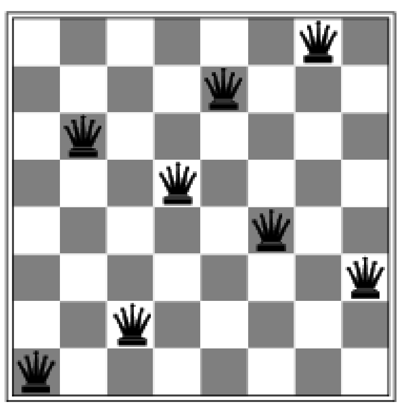
6

Example: Evolving 8 Queens

Sorry, wrong problem

7

Example: 8-queens problem

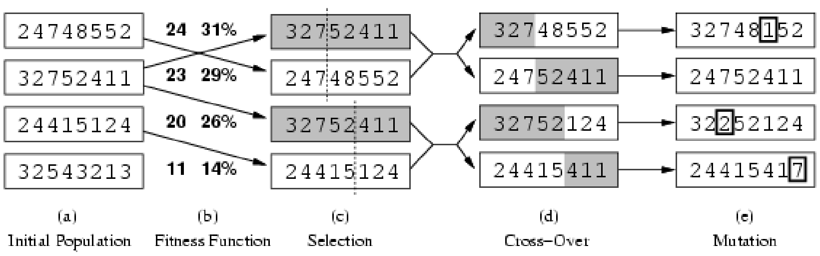


String Representation: 16257483

- *Can we evolve a solution through genetic algorithms?*

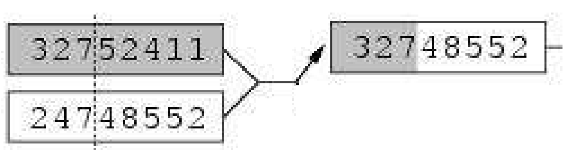
8

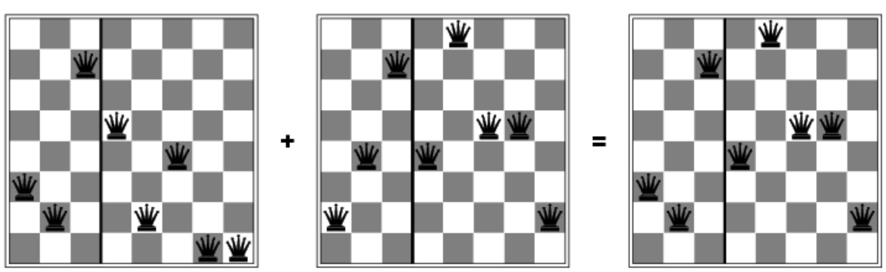
Example: Evolving 8 Queens



- Fitness function: number of non-attacking pairs of queens (min = 0, max = 8 × 7/2 = 28)
- 24/(24+23+20+11) = 31% probability of selection for reproduction
- 23/(24+23+20+11) = 29% etc

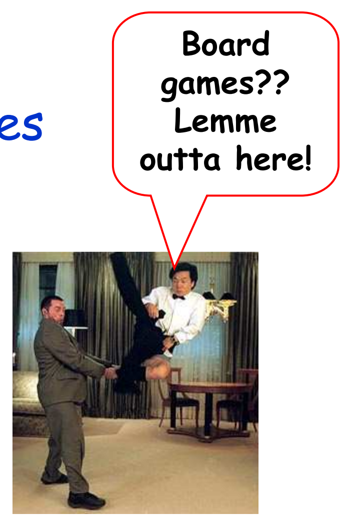
Queens crossing over





Adversarial Search

- Programs that can play competitive board games
- Minimax Search
- Alpha-Beta Pruning



12

Games Overview

	deterministic	chance
Perfect information	chess, checkers, go, othello	backgammon, monopoly
Imperfect information		poker, bridge, scrabble

13

Games & Game Theory

- When there is *more than one agent*, the future is not easily predictable anymore for the agent
- In *competitive* environments (conflicting goals), adversarial search becomes necessary
- In AI, we usually consider special type of games:

  board games, which can be characterized as ***deterministic, turn-taking, two-player, zero-sum*** games with ***perfect information***

14

Games as Search

- Components:
  - States:
  - Initial state:
  - Successor function:
  - Terminal test:
  - Utility function:

15

Games as Search

- Components:
  - States: board configurations
  - Initial state: the board position and which player will move
  - Successor function: returns list of *(move, state)* pairs, each indicating a legal move and the resulting state
  - Terminal test: determines when the game is over
  - Utility function: gives a numeric value in terminal states (e.g., -1, 0, +1 in chess for loss, tie, win)

16

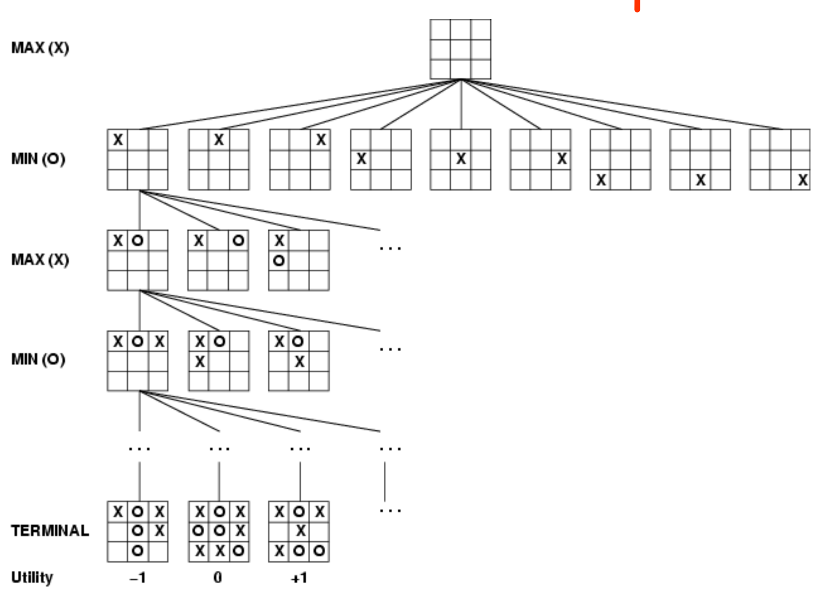
Games as Search

Convention: first player is MAX,
2nd player is MIN

- MAX moves first and they take turns until the game is over
- Winner gets reward, loser gets penalty
- Utility values stated from MAX's perspective
- Initial state and legal moves define the *game tree*
- MAX uses game tree to determine next move

17

Tic-Tac-Toe Example



18

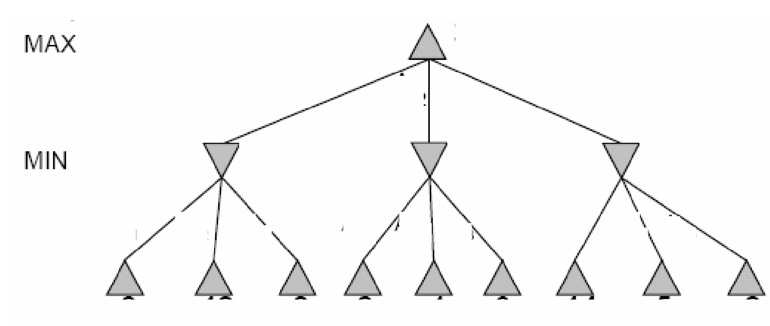
Optimal Strategy: Minimax Search

- Find the contingent *strategy* for MAX assuming an infallible MIN opponent
- Assumption: Both players play optimally!
- Given a game tree, the optimal strategy can be determined by using the *minimax* value of each node:

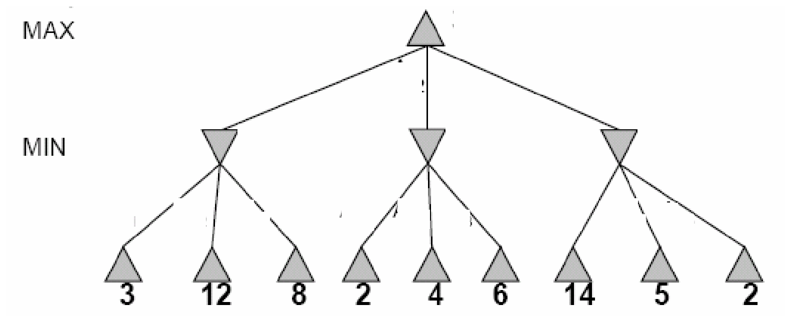
MINIMAX-VALUE(n)=

UTILITY(n)	If n is a terminal
$\max_{s \in succ(n)}$ MINIMAX-VALUE(s)	If n is a MAX node
$\min_{s \in succ(n)}$ MINIMAX-VALUE(s)	If n is a MIN node

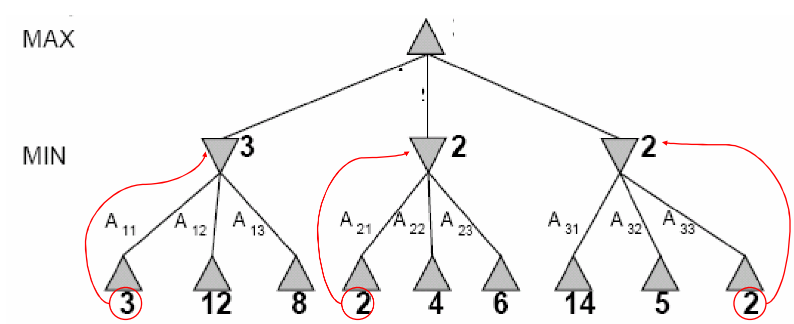
Two-Ply Game Tree



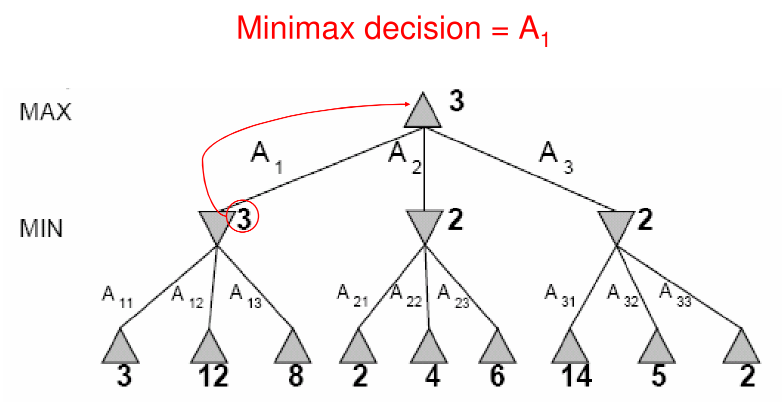
Two-Ply Game Tree



Two-Ply Game Tree



Two-Ply Game Tree



Minimax maximizes the worst-case outcome for max

23

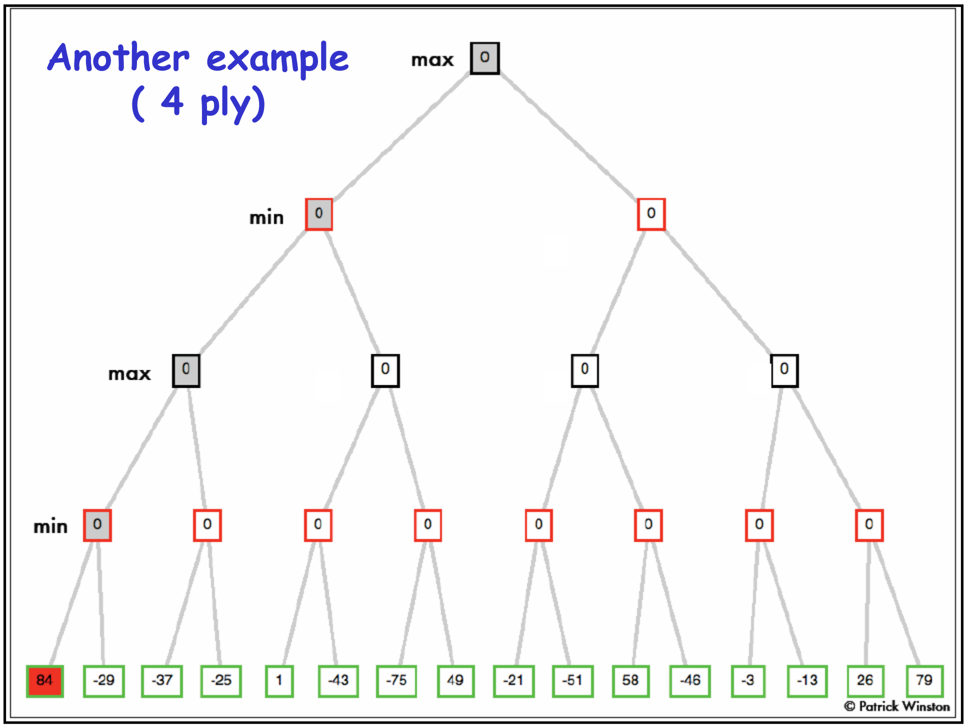
What if MIN does not play optimally?

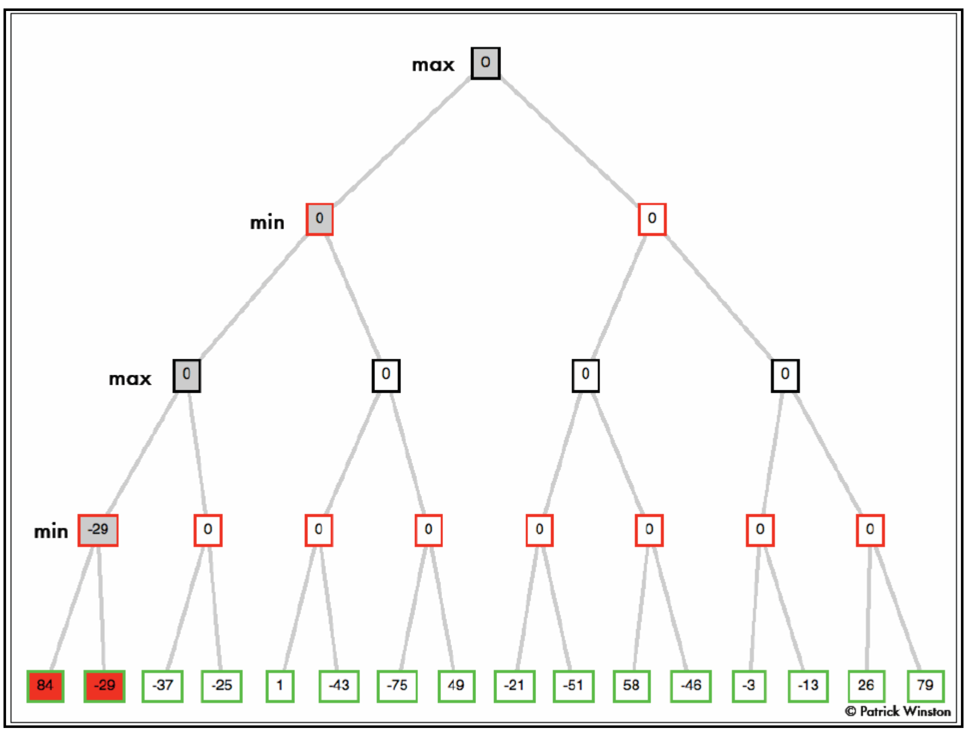
- Definition of optimal play for MAX assumes MIN plays optimally

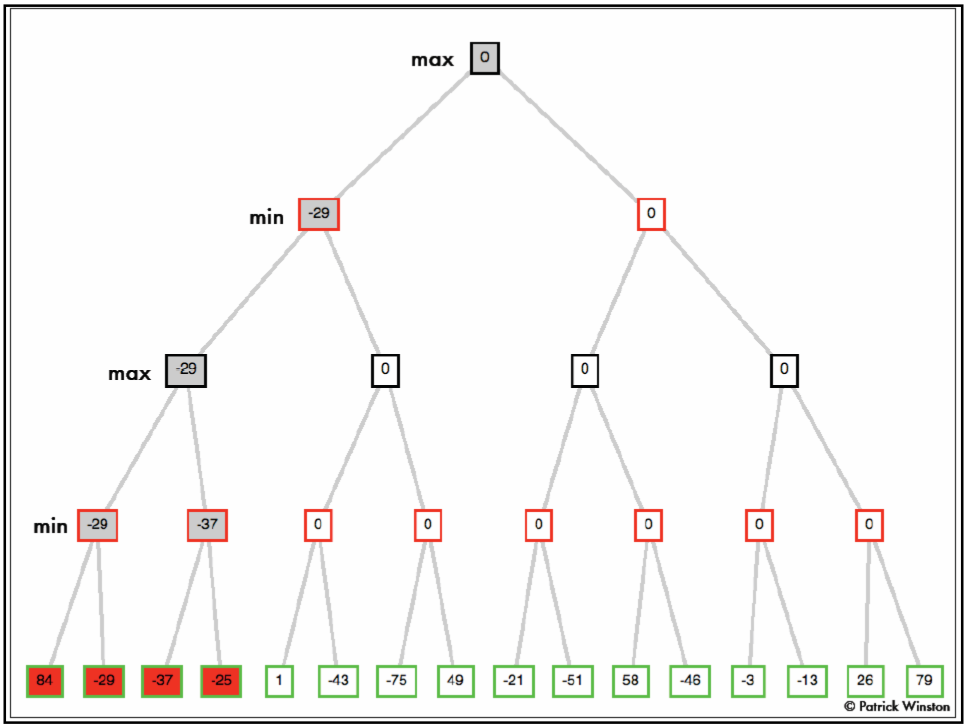
  Maximizes worst-case outcome for MAX

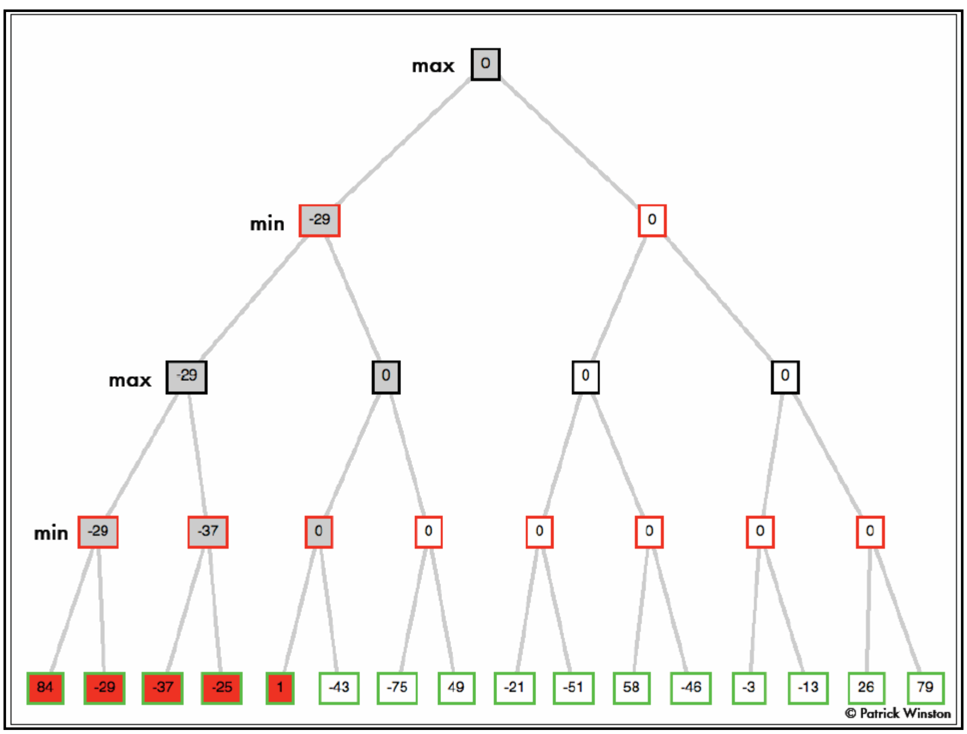
- If MIN does not play optimally, MAX will do even better. [Exercise 6.2]

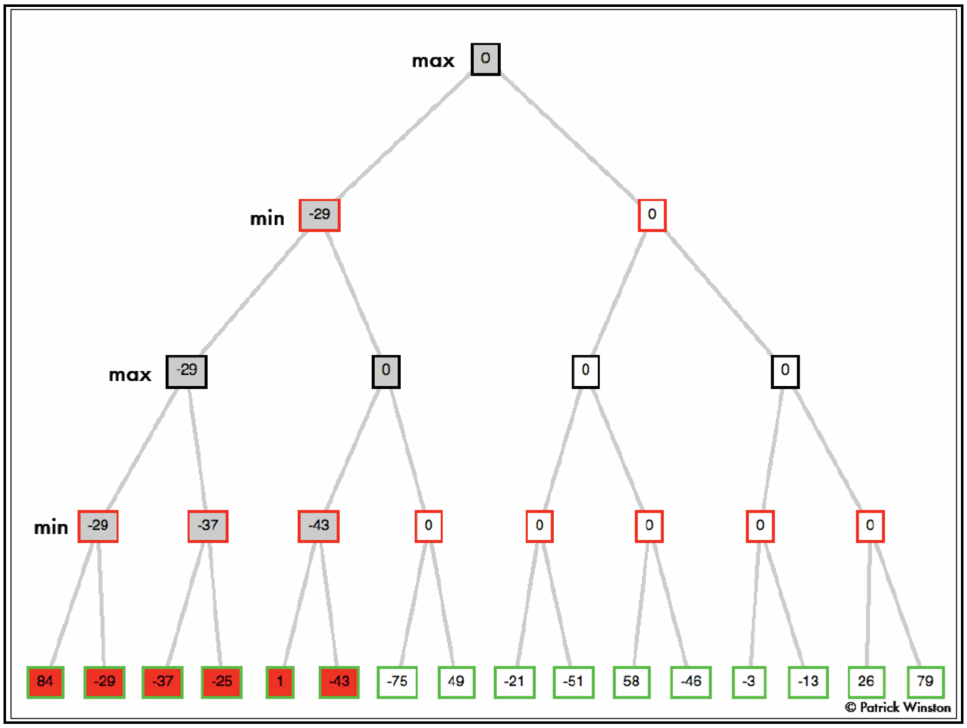
24

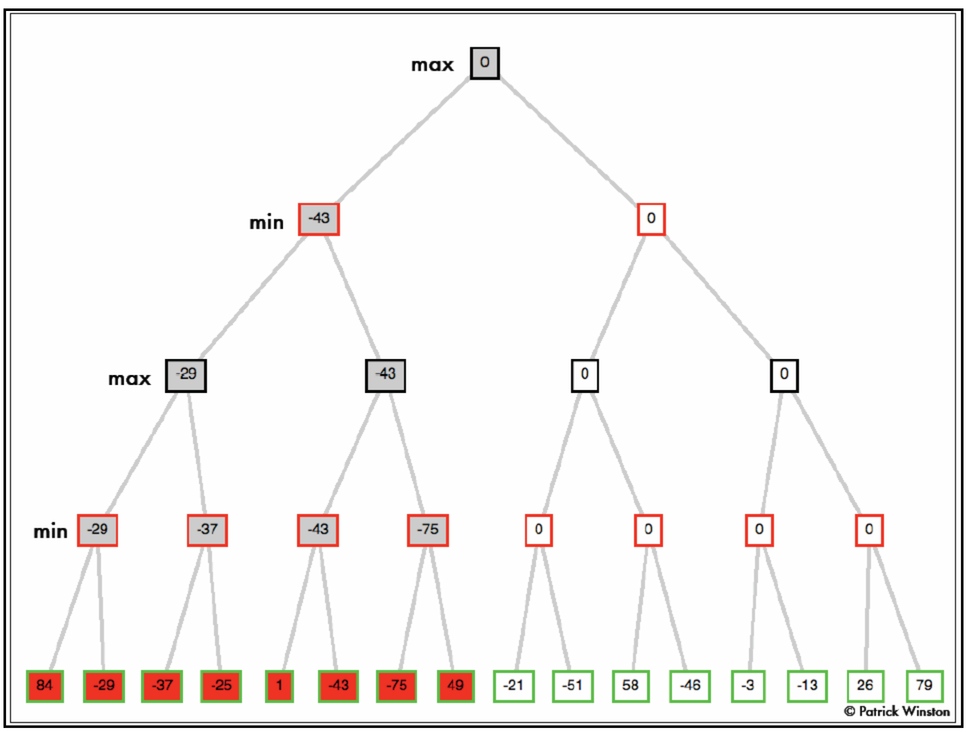

Another example
(4 ply)
max
0
min
0
0
max
0
0
0
0
min
0
0
0
0
0
0
0
0
84
-29
-37
-25
1
-43
-75
49
-21
-51
58
-46
-3
-13
26
79
© Patrick Winston

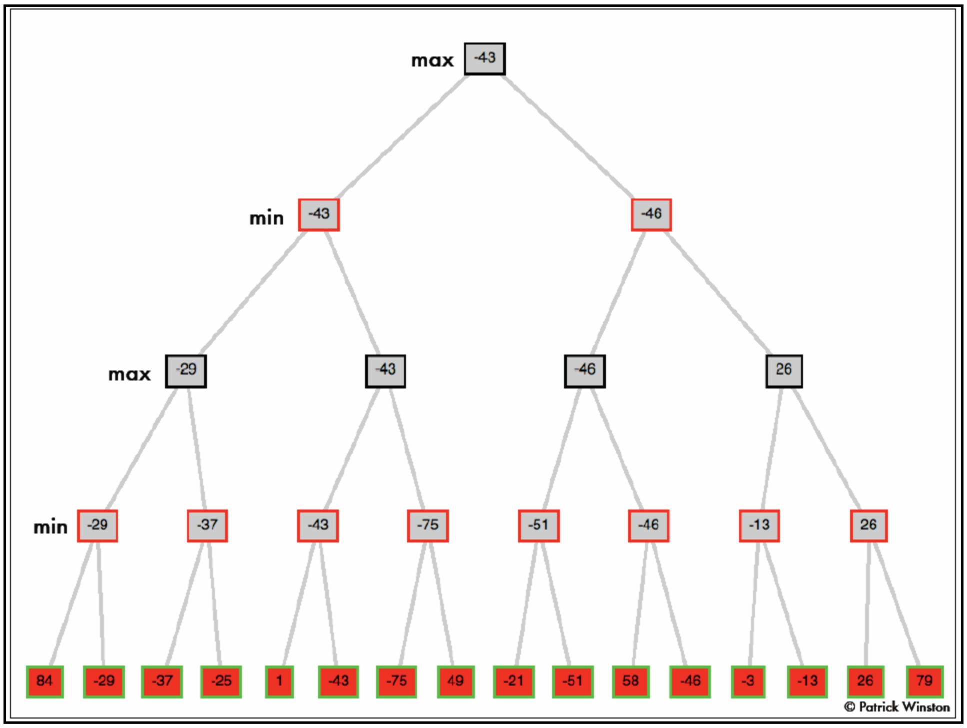

max
0
min
0
0
max
0
0
0
0
min
-29
0
0
0
0
0
0
0
84
-29
-37
-25
1
-43
-75
49
-21
-51
58
-46
-3
-13
26
79
© Patrick Winston

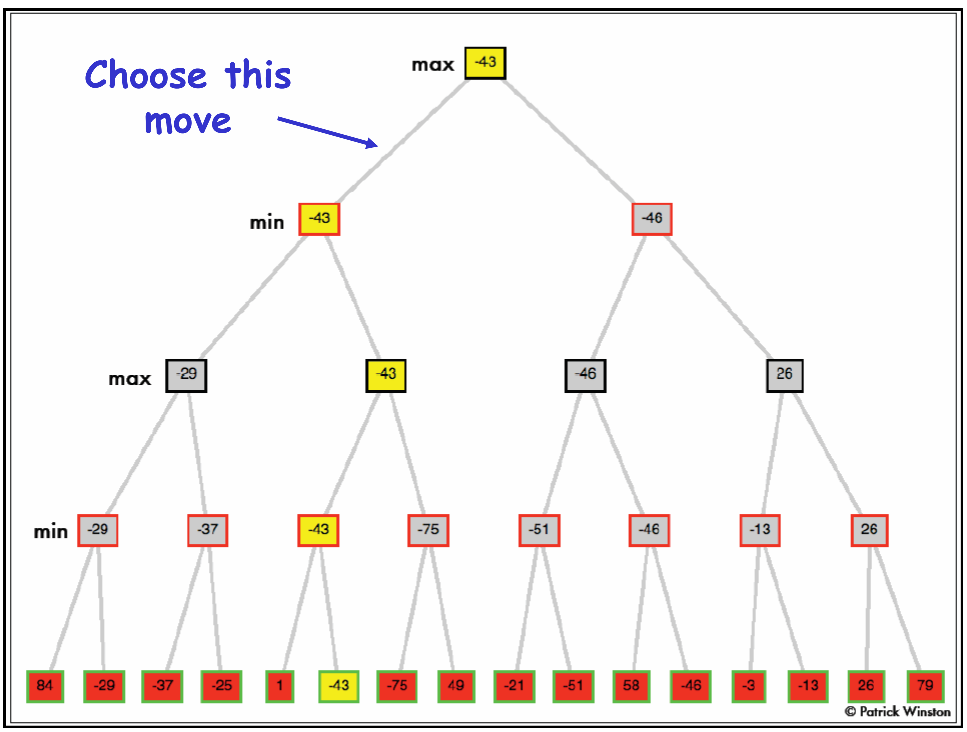

max
min
max
min
0
-29
0
-29
0
0
0
-29
-37
0
0
0
0
0
0
84
-29
-37
-25
1
-43
-75
49
-21
-51
58
-46
-3
-13
26
79
© Patrick Winston


max
min
max
min
0
-29
0
-29
0
0
0
-29
-37
0
0
0
0
0
0
84
-29
-37
-25
1
-43
-75
49
-21
-51
58
-46
-3
-13
26
79
© Patrick Winston


max
0
min
-29
0
max
-29
0
0
0
min
-29
-37
-43
0
0
0
0
0
84
-29
-37
-25
1
-43
-75
49
-21
-51
58
-46
-3
-13
26
79
© Patrick Winston


max
0
min
-43
0
max
-29
-43
0
0
min
-29
-37
-43
-75
0
0
0
0
84
-29
-37
-25
1
-43
-75
49
-21
-51
58
-46
-3
-13
26
79
© Patrick Winston


max
-43
min
-43
-46
max
-29
-43
-46
26
min
-29
-37
-43
-75
-51
-46
-13
26
84
-29
-37
-25
1
-43
-75
49
-21
-51
58
-46
-3
-13
26
79
© Patrick Winston


Choose this move
max
-43
min
-43
-46
max
-29
-43
-46
26
min
-29
-37
-43
-75
-51
-46
-13
26
84
-29
-37
-25
1
-43
-75
49
-21
-51
58
-46
-3
-13
26
79
© Patrick Winston

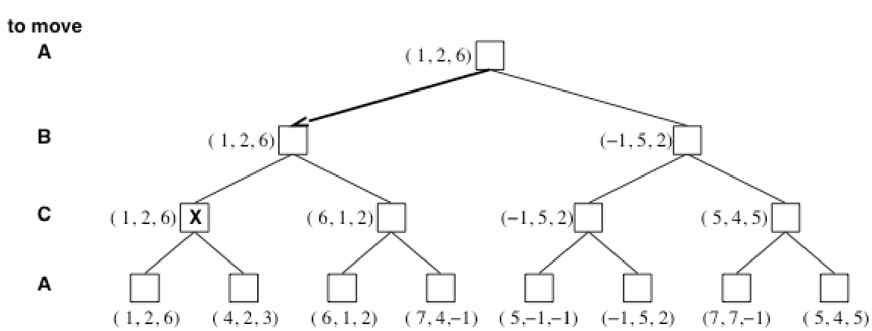
Minimax Algorithm

```
function MINIMAX-DECISION(state) returns an action
   v ← MAX-VALUE(state)
   return the action in SUCCESSORS(state) with value v

function MAX-VALUE(state) returns a utility value
   if TERMINAL-TEST(state) then return UTILITY(state)
   v ← −∞
   for a, s in SUCCESSORS(state) do
      v ← MAX(v, MIN-VALUE(s))
   return v

function MIN-VALUE(state) returns a utility value
   if TERMINAL-TEST(state) then return UTILITY(state)
   v ← ∞
   for a, s in SUCCESSORS(state) do
      v ← MIN(v, MAX-VALUE(s))
   return v
```

33

Multiplayer Games

- More than two players
- Single minimax values become vectors



34

Properties of minimax

- Complete?
- Optimal?
- Time complexity?
- Space complexity?

35

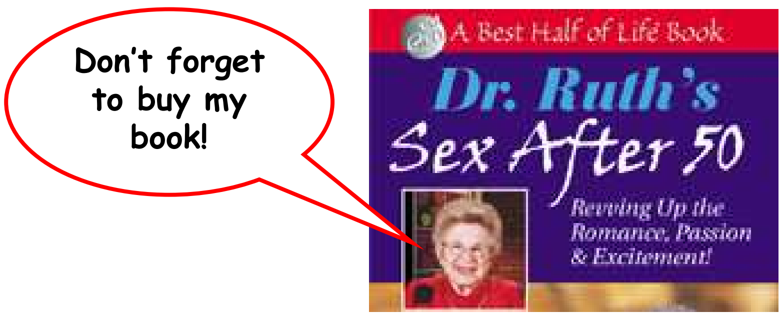
Properties of minimax

- Complete? Yes (if tree is finite)
- Optimal? Yes (against an optimal opponent)
- Time complexity? $O(b^m)$
- Space complexity? $O(bm)$ (depth-first exploration)

36

Next Time

- Alpha-beta pruning
- Real-time search and evaluation functions
- Rolling the dice



37