Alpha-beta pruning

- reduce the branching factor of nodes
- alpha value
  - associated with MAX nodes
  - represents the worst outcome MAX can achieve
  - can never decrease
- beta value
  - associated with MIN nodes
  - represents the worst outcome MIN can achieve
  - can never increase

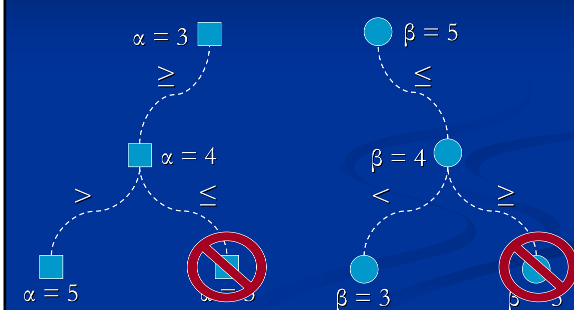
Example

- in a MAX node, $\alpha = 4$
  - we know that MAX can make a move which will result at least the value 4
  - we can omit children whose value is less than or equal to 4
- in a MIN node, $\beta = 4$
  - we know that MIN can make a move which will result at most the value 4
  - we can omit children whose value is greater than or equal to 4

Ancestors and α & β

- alpha value of a node is never less than the alpha value of its ancestors
- beta value of a node is never greater than the beta value of its ancestors

Once again



Rules of pruning

1. Prune below any MIN node having a beta value less than or equal to the alpha value of any of its MAX ancestors.
2. Prune below any MAX node having an alpha value greater than or equal to the beta value of any of its MIN ancestors

Or, simply put: If $\alpha \geq \beta$, then prune below!

Best-case analysis

- omit the principal variation
- at depth $d - 1$ optimum pruning: each node expands one child at depth d
- at depth $d - 2$ no pruning: each node expands all children at depth $d - 1$
- at depth $d - 3$ optimum pruning
- at depth $d - 4$ no pruning, etc.
- total amount of expanded nodes: $\Omega(b^{d/2})$

Principal variation search

- alpha-beta range should be small
  - limit the range artificially → aspiration search
  - if search fails, revert to the original range
- game tree node is either
  - α-node: every move has $e \leq \alpha$
  - β-node: every move has $e \geq \beta$
  - principal variation node: one or more moves has $e > \alpha$ but none has $e \geq \beta$

Principal variation search (cont'd)

- if we find a principal variation move (i.e., between α and β), assume we have found a principal variation node
  - search the rest of nodes the assuming they will not produce a good move
    - assume that the rest of nodes have values < α
    - null window: $[\alpha, \alpha + \varepsilon]$
  - if the assumption fails, re-search the node
  - works well if the principal variation node is likely to get selected first
    - sort the children?

Non-zero sum game: Prisoner's dilemma

- two criminals are arrested and isolated from each other
- police suspects they have committed a crime together but don't have enough proof
- both are offered a deal: rat on the other one and get a lighter sentence
  - if one defects, he gets free whilst the other gets a long sentence
  - if both defect, both get a medium sentence
  - if neither one defects (i.e., they co-operate with each other), both get a short sentence

Prisoner's dilemma (cont'd)

- two players
- possible moves
  - co-operate
  - defect
- the dilemma: player cannot make a good decision without knowing what the other will do

Payoffs for prisoner A

Prisoner B's move / *Prisoner A's move*	**Co-operate: keep silent**	**Defect: rat on the other prisoner**
Co-operate: keep silent	Fairly good: *6 months*	Bad: *10 years*
Defect: rat on the other prisoner	Good: *no penalty*	Mediocre: *5 years*

Payoffs in Chicken

Driver B's move / *Driver A's move*	**Co-operate: swerve**	**Defect: keep going**
Co-operate: swerve	Fairly good: *It's a draw.*	Mediocre: *I'm chicken...*
Defect: keep going	Good: *I win!*	Bad: *Crash, boom, bang!!*

Payoffs in Battle of Sexes

Wife's move *Husband's move*	**Co-operate: boxing**	**Defect: opera**
Co-operate: opera	Wife: Very bad Husband: Very bad	Wife: Good Husband: Mediocre
Defect: boxing	Wife: Mediocre Husband: Good	Wife: Bad Husband: Bad

Iterated prisoner's dilemma

- encounters are repeated
- players have memory of the previous encounters
- R. Axelrod: *The Evolution of Cooperation* (1984)
  - greedy strategies tend to work poorly
  - altruistic strategies work better—even if judged by self-interest only
- Nash equilibrium: always defect!
  - but sometimes rational decisions are not sensible
- Tit for Tat (A. Rapoport)
  - co-operate on the first iteration
  - do what the opponent did on the previous move