472 Recitation

Week 6

Problem set 2 due on this Friday at 5:00 PM

Game search

Weakness of alpha-beta search:

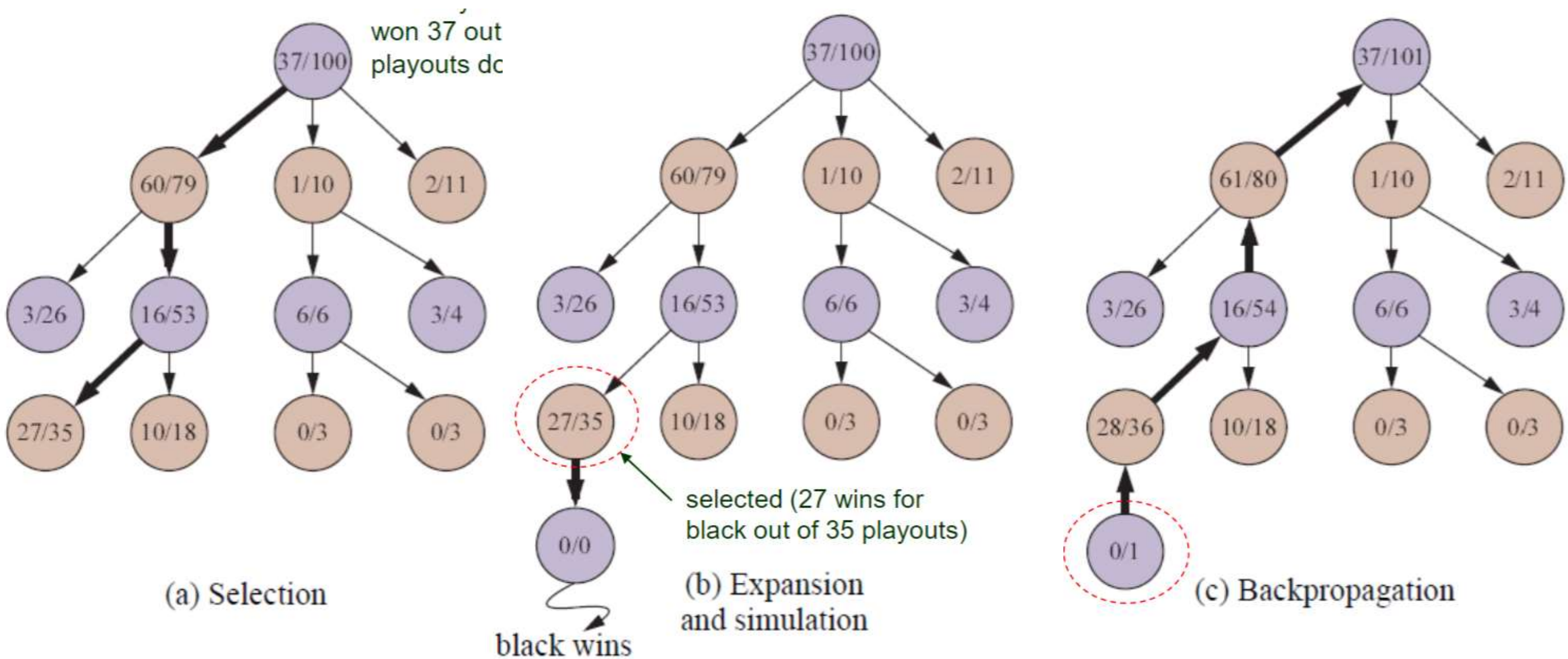
- High branching factor
- Rely on good Evaluation function

↓

Monte Carlo Method
Idea: rely on repeated random sampling to obtain numerical results

For game: no heuristic evaluation function and the value of a state estimated as the average utility over a number of simulations of complete games

Mento Carlo Method



Mento Carlo Method

Upper confidence bound formula:

$$\text{UCB}(n) = \frac{U(n)}{N(n)} + C \times \sqrt{\frac{\log N(PARENT(n))}{N(n)}}$$

- MCTS has advantage over alpha-beta when $b$is high.
- MCTS is less vulnerable to a single error.
- MCTS can be applied to brand-new games via training by self-play.
- MCTS is less desired than alpha-beta on a game like chess with low b and good evaluation function.

Constraint Satisfaction Problem

- A set of variables $\mathcal{X} = \{X_1, \dots, X_n\}$.
- A set of domains $\mathcal{D} = \{D_1, \dots, D_n\}$.
- A set of constraints $\mathcal{C} = \{C_1, \dots, C_m\}$ that specifies allowable combination of values.
  - C_j: $\langle (v_i, v_j)$, relation$\rangle$

Assignment: $\{X_i = v_i, X_j = v_j, \dots\}$

- consistent if no constraint is violated.
- complete if every variable is assigned a value.
- partial if some variables are unassigned.

A solution to a CSP is a consistent, complete assignment.

A partial solution is a partial assignment that is consistent.

Formulate CSP

- A natural representation for a wide variety of problem.
- Fast and efficient CSP solvers.
- Ability of a CSP solver to reduce the search space significantly.

Upon violation by a partial assignment

- discard its further refinements,
- see which variables cause the violation,
- focus on those variables that matter.

Formulation of CSP is getting the variables, domains, and constraints form the problem description.

Constraints

A *unary constraint* restricts the value of a single variable.

$$\langle (SA), SA \neq green \rangle$$

A *binary constraint* relates two variables.

$$SA \neq WA$$

A higher order constraint relate more variables.

A *global constraint* involves an arbitrary number of variables (but not necessarily all the variables).

Constraint Propagation

Idea: Use constraints to reduce the number of legal values for a variable, which in turn reduce those for another variable, and so on.

A variable X_i is *arc-consistent* with respect to (w.r.t.) another variable X_j if for every value in D_i there exists a value in D_j that satisfies the binary constraint on the arc (i.e., edge (X_i, X_j)).

The variable is *arc-consistent* if it is so w.r.t every other variable that it shares a binary constraint with.

The constraint graph is *arc-consistent* if every variable is arc consistent with every other variable.

Any CSP can be transformed into a binary CSP.

Arc-Consistency Algorithm AC-3

```
function AC-3( csp) returns false if an inconsistency is found and true otherwise
  queue ← a queue of arcs, initially all the arcs in csp

  while queue is not empty do
    (X_i, X_j) ← POP(queue)
    if REVISE(csp, X_i, X_j) then
      if size of D_i = 0 then return false
      for each X_k in X_i.NEIGHBORS - {X_j} do   // propagate to other variables sharing a
        add (X_k, X_i) to queue                   // constraint with X_i since its domain has
  return true                                     // reduced.

function REVISE( csp, X_i, X_j) returns true iff we revise the domain of X_i
  revised ← false
  for each x in D_i do
    if no value y in D_j allows (x,y) to satisfy the constraint between X_i and X_j then
      delete x from D_i
      revised ← true
  return revised
```

Domain size $\leq d$, c binary constraints.

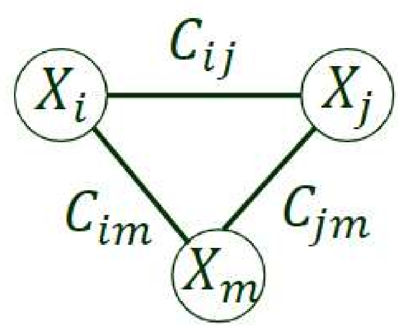
- Each arc inserted into the queue d times.
- Each checking takes $O(d^2)$ time.

$O(cd^3)$

Constraint Propagation

- Arc-consistency
- Path Consistency
- Bounds Propagation

$\{X_i, X_j\}$ is *path-consistent* w.r.t. X_m if for every assignment to X_i, X_j consistent with their constraint C_{ij} (if exists), there exists an assignment to X_m that satisfies the constraints C_{im} and C_{jm} between them and X_m.



Backtracking Search

- Constraint propagation often ends with partial solutions.
- Backtracking search can be employed to extend them to full solutions.