MAST : move averaging sampling technique

- looks at kinds of moves you can make
- keeps global stats during playouts for each kind of move
  - # times played that move
  - # times won after playing move
- bias random playout in favor of better kinds of moves

can also hard-code move types – Kalah 0 ... 5 or extra turn capture other

PAST : predicate averaging sampling technique

keep global stats for all predicates
- for states that satisfy that predicate
  - count visits / wins

bias playout to favor moves to states with "good" predicates

## Optimization

measure of strength of player

Given a function $f(x_1, \ldots, x_n)$, find $x_1, \ldots, x_n$ that yield max $f(x_1, \ldots, x_n)$

constants in heuristic

find $x, y$ to maximize $4x + 5y - 2xy - x^2 - y^2$

NP-complete

find ordering of Seattle, Denver, Allentown, Baltimore, Amherst, Washington to minimize distance of corresponding tour

find assignment of classrooms to minimize student conflicts

find ordering of teams to minimize upsets during previous season

NP-complete

| | A | B | C | H | W |
|---|---|---|---|---|---|
| A | | W | W | L | W |
| B | | | W | L | W |
| C | | | | W | W |
| H | | | | | W |

(X: beat ↓)

W A C B H

Solitaire Yahtzee: Estimate start-of-turn position value by counting

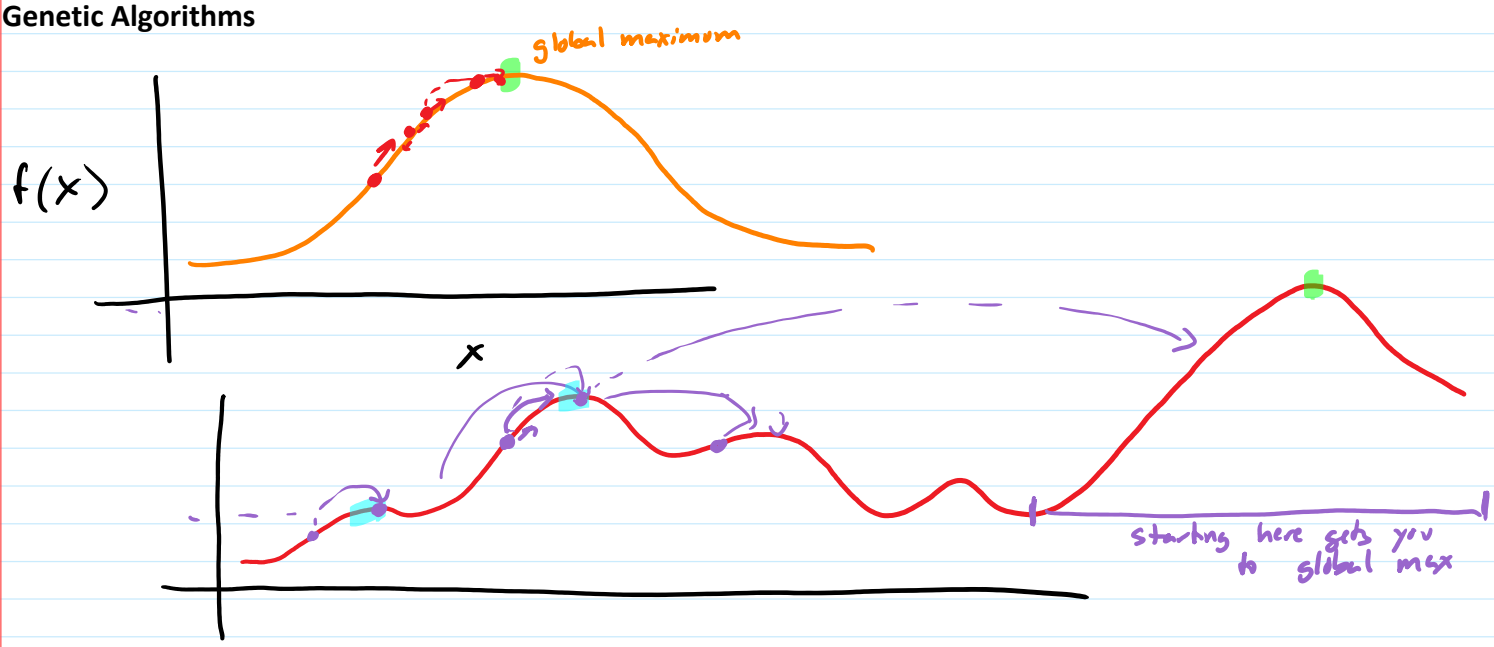
3 for 1's open
6 for 2's
⋮
18 for 6's open
+6/open upper if on pace for bonus

20 for open 3 of a kind
10 4
15 for FH
⋮
10 for Yahtzee

Play each turn to maximize turn score + next position value using heuristic

Find parameters to maximize avg. score

## Genetic Algorithms




Evolutionary Computation
mimic natural selection — nature-inspired
individuals have genes (numbers for numeric functions)
genes determine phenotype — inputs to function
phenotype contributes to fitness
fitness contributes to propagation

Genetic Algorithms

start with random population - collection of individuals with randomly selected genes (random 0's and 1's)

while not done (out of time, no improvement, found good enough solution)

evaluate each individual — genes → inputs to f → evaluate f → fitness

select for crossover — select pairs of individuals, biased towards higher fitness

crossover — randomly select genes from each parent to create offspring's genes

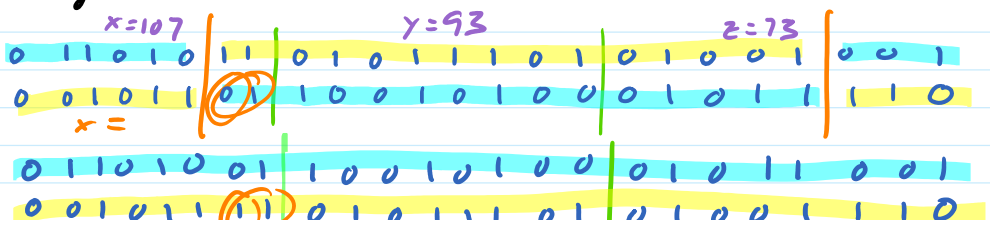
select for survival — replacement or fitness-based

mutate — randomly change genes in some individuals (typically low rate)

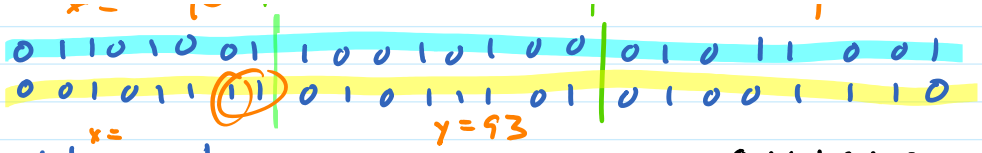
Representation: what is genetic code?

$f(x,y,z)$

$f: \mathbb{Z}\times\mathbb{Z}\times\mathbb{Z} \Rightarrow \mathbb{R}$



f: C × C × C ⇒ K




Crossover : bitstrings : two-point crossover

0 1 1 1 0 1 0
1 1 0 1 1 0 0

permutation

C F B A E D
D C F A B E

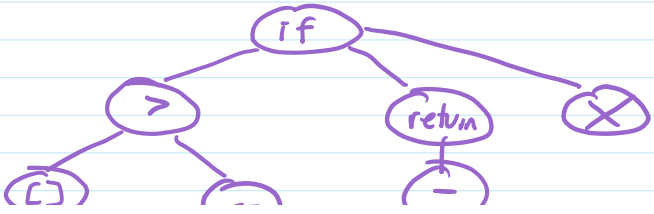
| A | B | C | D | E | F |
|---|---|---|---|---|---|
| 60 | 70 | 100 | 20 | 10 | 80 |
| 32 | 30 | 50 | 75 | 26 | 40 |
| 50 | 70 | 50 | 75 | 21 | 80 |

Genetic Programming :

```
if seeds[0] > 5
{
    many++
}
```

```
if seeds[i] == 0 and seeds[i - 1] == 1
{
    return i-1
}
```

```
if seeds[0] > 5
{
 return i - 1;
}
```



```
 return i - 1;
}
```