Optimization

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

- measure of strength of resulting agent
- constants in heuristic

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

TSP
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

NP-complete
find ordering of teams to minimize upsets during previous season

	A	B	C	H	W
A		W	W	L	W
B			W	L	W
C				W	W
H					W

B A H C W

W A C B H

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

-2	1	3 for 1's open
-4	2	6 for 2's
0	12	12 for 3's
		18 for 6's open
		+6/open upper if on pace for bonus
-5	15	20 for open 3 of a kind
-10	0	10 for 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

Find parameters to maximize avg. score

chess: x_1 9 pts green queen
5 pts rook
x_5 1 pt pawn

Fitness for 2-player games:
(strength)

Elo:

numeric measure of strength, based on strength of opponents

2000 beats 1000
↓
small change

1000 beats 2000
↓
larger change

difference in rating of 200 means stronger

ratings of players

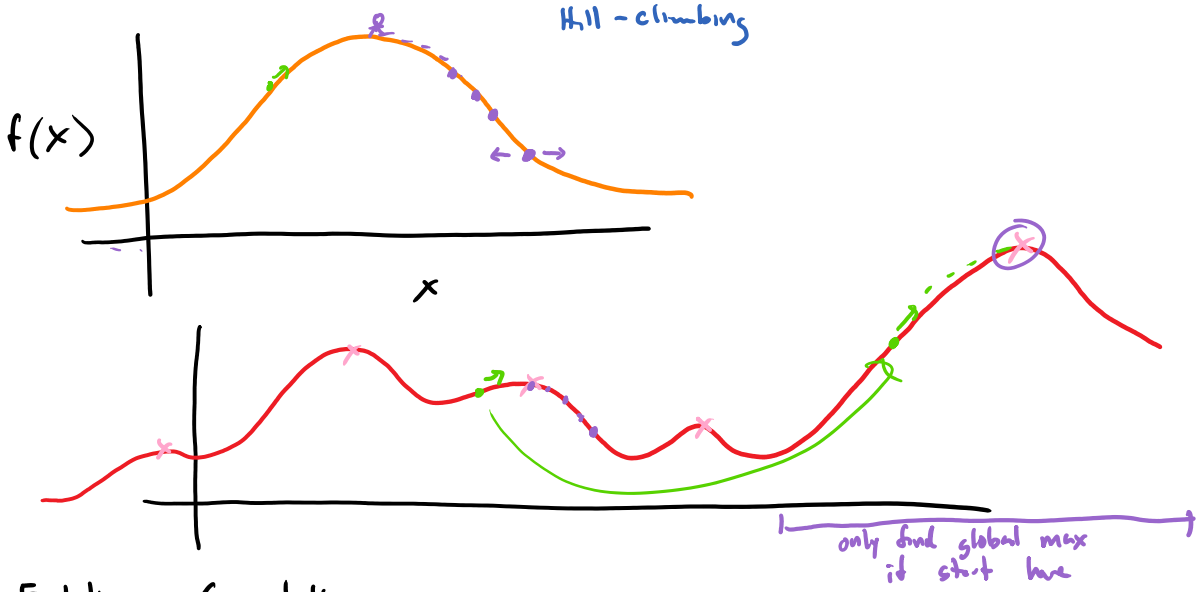
$$\text{expected wins for } A = \frac{1}{1 + 10^{\frac{R_B - R_A}{400}}}$$

$$\text{update for } R_A = K \cdot (\text{observed wins for } A - \text{expected wins for } A)$$

$K = 32$ (constant)

$B = 1600$
$A = 1400$
A beats B → 1576
1424

Genetic Algorithms



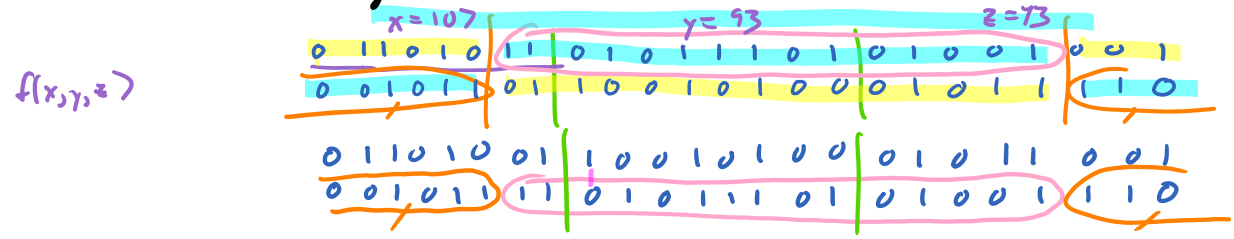
Evolutionary Computation
mimic natural selection
individuals have genes (numeric inputs)
genes determine phenotype
phenotype contributes to fitness
fitness contributes to propagation

Genetic Algorithms

start with random population — collection of candidate solutions randomly initialized
while not done (out of time, no improvement, good enough)
evaluate each individual — genes → inputs to f → evaluate f → fitness (fitness fxn)
select for crossover — select pairs of individuals, bias towards higher fitness — exploitation
crossover — combine genes of selected pairs to produce offspring
select for survival — replacement or fitness-based — offspring replace parents
mutate — randomly change genes in survivors ← exploration

pop of 100 → 50 pairs → 2 offspring per pair → next generation of 100

Representation: what is genetic code?



Crossover: bitstrings

3 | 4 6 | 1 11
4 | 2 10 | 6 3

Permutation C E B | A F | D

permutation

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

C F B A B D

D C F A E E

A	B	C	D	E	F
40	70	100	20	30	80
40	30	70	90	10	50

DFBCAE ← 40 70 70 90 30 80

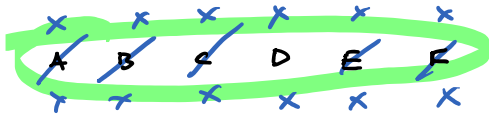
3	4	6	1	11	
4	2	10	6	3	

3 2 10 1 11

4 4 6 6 3

2	5	2	1	2	B F C A E D
4	3	4	1	1	D C F A B E

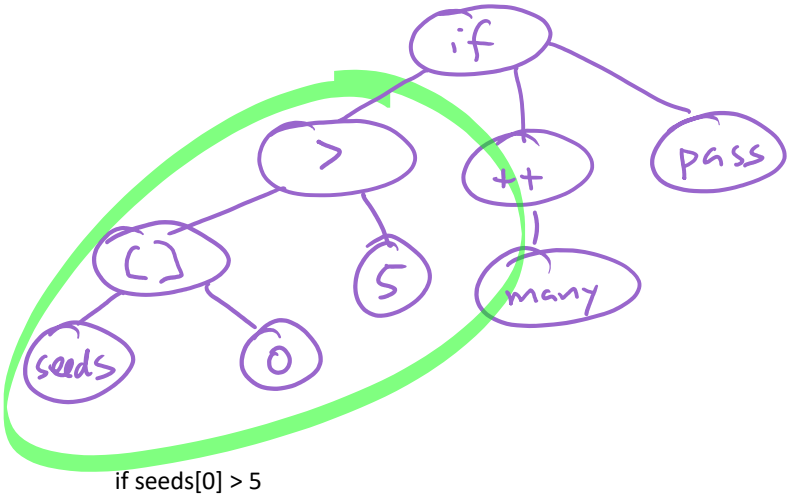
2 3 4 1 2 B D F A E C



Genetic Programming: GA on programs

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

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




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