Topic 19 - Genetic Algorithms (continued)

Monday, May 9

numpy

Announcements:

- → HW 5 due today by midnight
- → Final assigned today, due midnight Mon, May 16
- → Office Hours:

Tues	5/10	1pm-2pm	} in my office,
Wed	5/11	2pm-3pm	} on Mon can
Mon	5/16	10am-11am	} have Teams
			open if desired

Pseudocode:

```
pop = [n random solutions]
while True:
    best = best solution we've ever seen
    next_gen = []
    while len(next_gen) < len(pop):
        select two parents P1 and P2 in pop   (how?)
        perform crossover on P1 and P2        (how?)
            to get some children
        allow each child to mutate with       (how?)
            some probability
        add the children to next_gen
```

pop = next_gen

Crossover:

Ex: Knapsack

You can think about a solution as a vector of booleans (T/F) that tell you whether each item is in or out.

0 1 2 3 4 5 6 7

S = [T T F T F F F T]

⇒ items 0, 1, 3, 7 are in.

Suppose we have two solutions:

S_1 = [T T F T F F F T]

S_2 = [T F T T F F F F]

How can we crossover to create one or more children?

* One-point crossover: Pick a random place in the vector and swap the blocks after.

C_1 = [T T F T F | F F F]

C_2 = [T F T T F | F F T]

(you're going to want a penalized scoring

function, because these may violate constraints.)

* Two-point crossover: Pick two points, swap in the middle

$C_1 = [T\ T \mid T\ T\ F \mid F\ F\ T]$
$C_2 = [T\ F \mid F\ T\ F \mid F\ F\ F]$

* Form one child by flipping a coin for each item.

$S_1 = [T\ T\ F\ T\ F\ F\ F\ T]$
$S_2 = [T\ F\ T\ T\ F\ F\ F\ F]$

$C = [T\ F\ F\ T\ F\ F\ F\ F]$

Can be repeated many times with same parents.

Which is best? Depends on the problem.
Knapsack → the order of the items doesn't mean anything
one and two point crossover have a

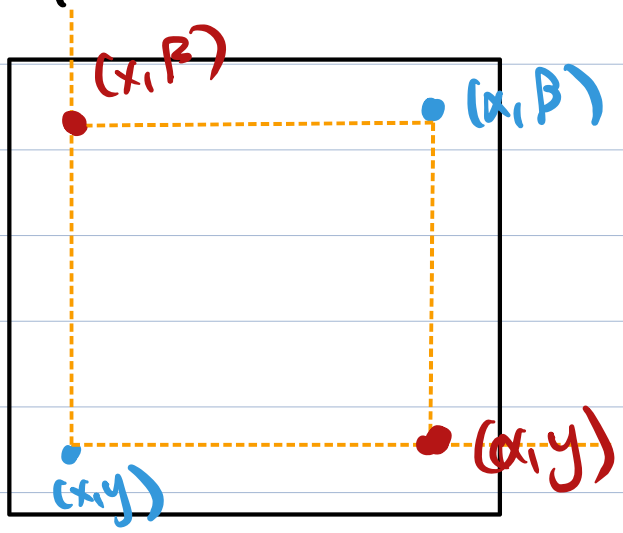
bias that the third option does not

How about for continuous functions? In 2D, a solution is a pair of real #s (x, y).

$S_1 = (x, y)$ $\quad$ $S_2 = (\alpha, \beta)$

$C_1 = (x, \beta)$ $\quad$ $C_2 = (\alpha, y)$

Does it make sense that the children could "inherit" good qualities from their parents? No.



0.9

Blue Lights problem:

$S_1 = [l_1, l_2, \ldots, l_{10}]$

$S_2 = [m_1, m_2, \ldots, m_{10}]$

Flip a coin for each one:

$$C_1 = [l_1, l_2, m_3, l_4, l_5, m_6, m_7, m_8, l_9, l_{10}]$$

Mutation:

After getting children from crossover, you want each child to have a chance at mutating into something better or worse.

Option 1: With some fixed probability (~20%) mutate each child by doing a tweak.

Option 2: Think about the components that make up a solution ("chromosomes") Give each individual chromosome the chance to mutate with a small prob: $\frac{1}{\#\text{ of chromosomes}}$

Ex: Knapsack:

$S_1 = [T\ T\ F\ T\ F\ F\ F\ T]$

$S_2 = [T\ F\ T\ T\ F\ F\ F\ F]$

$C = [T\ F\ F\ T\ F\ F\ F\ F]$ with prob $\frac{1}{8}$, change each

$C' = [T\ T\ F\ F\ F\ F\ F\ F]$

Selection Methods:

How do we choose pairs of parents to crossover?

(1) Randomly pick 2 different parents, but leave them in the pool for future pairings.

(2) Fitness Proportionate Selection / Roulette Selection

Select at random, but not with equal prob. Set the probs based on fitness score.

(3) Stochastic Universal Sampling

(4) Tournament Selection

(2) and (3) have the problem that they're very dependent on precise numerical values for the scores.

Ex: Three parents with scores

99.98, 99.99, 100.0

T.S. picks based on relative ranking