Friday, April 30 Lecture #40/42

→ Homework Questions?

Topic 20 - Genetic and Evolutionary Algorithms

Pseudocode:

```
pop = [μ random solutions]
while True:
  best = best solution in pop
  next_gen = []
  while len(next_gen) < len(pop):
*     select two parents P1, 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
```

Ex: Knapsack

You can think of a knapsack as

a vector of booleans saying whether each item is in or out.

$$S = [\,T \;\; T \;\; F \;\; T \;\; F \;\; F \;\; F \;\; T\,]$$
$$0 \;\; 1 \;\; 2 \;\; 3 \;\; 4 \;\; 5 \;\; 6 \;\; 7$$

Items 0, 1, 3, 7 are in the K.S.

Suppose we have two solutions:

$S_1 = [\,T \;\; T \;\; F \;\; T \;\; F \;\; F \;\; F \;\; T\,]$ 0, 1, 3, 7

$S_2 = [\,T \;\; F \;\; T \;\; T \;\; F \;\; F \;\; F \;\; F\,]$ 0, 2, 3

How can we create one or more children that resemble each parent in some way?

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

Item 8

$C_1 = [\,T \;\; T \;\; F \;\; T \;\; F \;|\; F \;\; F \;\; F\,]$ 0, 1, 3

$C_2 = [\,T \;\; F \;\; T \;\; T \;\; F \;|\; F \;\; F \;\; T\,]$ 0, 2, 3, 7

To do this, you want a penalized

scoring function.

* Two-point crossover: Pick two places in the vector and swap everything in between.

$C_1 = [T \;\; T \;|\; T \;\; T \;\; F \;|\; F \;\; F \;\; T]$ 0,1,2,3,7

$C_2 = [T \;\; F \;|\; F \;\; T \;\; F \;|\; F \;\; F \;\; F]$ 0,3

* Uniform Crossover: To avoid the location bias, form one child by flipping a coin for each index, if heads take from S_1, if tails take from S_2.

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

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

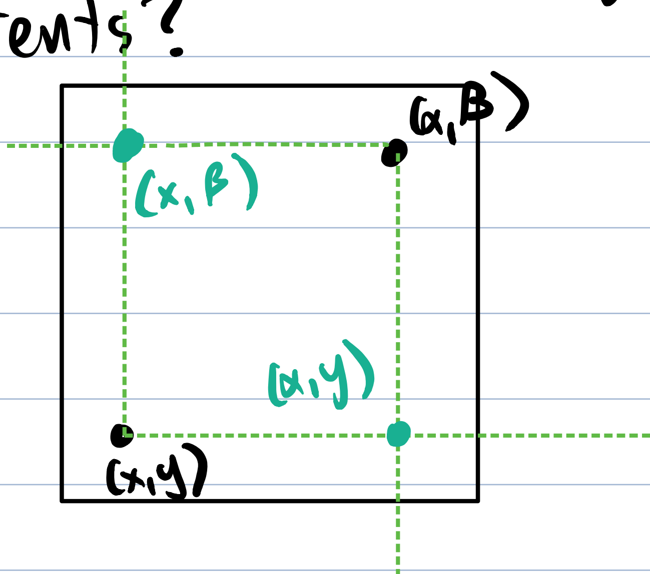
$C = [T \;\; F \;\; F \;\; T \;\; F \;\; F \;\; F \;\; F]$

For knapsack, uniform is best because it eliminates the bias caused by the order of the list.

<u>Ex</u>: Optimizing Continuous Functions

Solutions look like vectors of real #s $[x, y]$.

If we have two good solutions (x, y) and (α, β). Does it make sense that (x, β) and (α, y) could inherit the "goodness" of the parents?

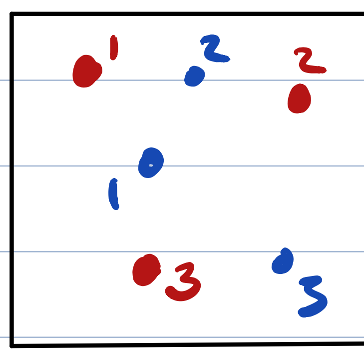


$(x_1, y_1, x_2, y_2, \ldots, x_{10}, y_{10})$
(10 lights)

<u>Ex:</u> Blue Lights Problem

$$S_1 = [l_1, l_2, l_3, \ldots, l_{10}]$$
$$S_2 = [m_1, m_2, m_3, \ldots, m_{10}]$$

* Uniform Crossover



TSP - gets complicated, we'll talk more about this at the end

Mutation:

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

Option 1: With some fixed probability (10% - 20%?), do a tweak to a child.

Option 2: (more common)
Think about the components/features that make up a solution. People call these the "chromosomes."

* Give each individual chromosome of each child a chance to mutate, with probability $\frac{1}{\#\text{ chromosomes}}$.

Ex: Knapsack

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

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

$C = [T \ F \ F \ T \ F \ F \ F \ F]$ prob $\frac{1}{8}$

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

Ex: Blue lights

$C = [\ell_1, \ell_2, \cdots \ell_{10}]$

* move 1 coordinate of ℓ_2
* move ℓ_2 a little bit
* delete it and randomly pick a new one