

# Topic 19 - Genetic Algorithms

Friday, May 6

①

## Announcements:

- HW 5 due the last day of class <sup>Mon, May 9</sup> 11:59pm
- Final will be takehome, due Mon, May 16, 11:59pm

This topic is about a group of population MTHs inspired by evolution.

Unlike PSO, Firefly, Cuckoo Search, the solution candidates don't need to be points in space.

First group of MTHs: "Evolutionary Strategies"

Summary: Do some tweaks to all the things in our population, and keep the best ones.

## "( $\mu, \lambda$ ) Evolution Strategy"

- \* Start with a population of  $\mu$  solutions.
- \* Tweak each one  $\frac{\lambda}{\mu}$  times, and keep all the results whether better or worse.
- \* Out of those  $\lambda$  tweaks, keep the best  $\mu$  of them.
- \* Repeat

Ex:  $\mu=10, \lambda=50, \text{TSP}$

Start with 10 tours  $\rightarrow$  tweak each 5 times

Out of the 50 tours, keep the best 10.

Pseudocode:

pop = [ $\mu$  random solutions] (or greedyish)

while True:

best = best sol in pop

next\_gen = []

for sol in pop:

repeat  $\lambda/\mu$  times:

new\_sol = tweak(sol)

next\_gen.append(new\_sol)

pop = [ $\mu$  best things in next\_gen]

## Variant: "( $\mu + \lambda$ ) Evolutionary Strategy"

In this version, when the next generation of  $\mu$  things, both the parents and the children compete.

### Pseudocode:

```
pop = [ $\mu$  random solutions] (or greedyish)
```

```
while True:
```

```
    best = best sol in pop
```

```
    next_gen = []
```

```
    for sol in pop:
```

```
        repeat  $\lambda/\mu$  times:
```

```
            new_sol = tweak(sol)
```

```
            next_gen.append(new_sol)
```

```
    pop = [ $\mu$  best things in next_gen + pop]
```

This one keeps good solutions around for longer:

- \* more exploitation
- \* less exploration

These days: common to do these with a tweak function whose intensity

can be dialed up or down to find a good balance of exploration / exploitation.

### "One-Fifth Rule"

- \* Aim for about  $1/5$  of children to be better than their parents
- \* If more, too much exploitation, dial up the exploration with bigger tweaks
- \* If less, too much exploration, dial down with smaller tweaks.

### Examples:

continuous problems — Gaussian walk with bigger or smaller standard deviation.

TSP — use k-opt tweaks with larger or smaller values of k.

### Genetic Algorithms

This is tied for most well known MH with Sim. Ann.

New idea: crossover

- \* Single parents can create children with a tweak (called a "mutation")
- \* Two parents can combine (aka. crossover) to produce one or more offspring that take some qualities from each parent.

Main Idea:

We'll start with a population of  $n$  solutions.

To form the next gen, we'll pick two sols. to be parents, then cross them over to form children. (Usually one child, sometimes more). Each child then has some probability of mutating on their own. After a while, we have a bunch of children, and they become the next generation of parents.

- \* The hope is that two parents that are pretty good can make a child that has the best qualities from each.
- \* Random mutation allows new good qualities to appear on their own

## Pseudocode:

pop = [  $\mu$  random solutions ]

while True:

best = best solution we've ever seen

next\_gen = [ ]

while len(next\_gen) < len(pop):

select two parents  $P_1$  and  $P_2$  in pop (how?)

perform crossover on  $P_1$  and  $P_2$  (how?)

to get some children

allow each child to mutate with (how?)

some probability

add the children to next\_gen

pop = next\_gen