Extra Credit Opportunities

- Tu, Feb 27, 4:15pm, Gates G01: Nika Haghtalab, CMU
  "Machine learning by the people, for the people"
- Th, Mar 1, 4:15pm, Gates G01: Jacob Steinhardt, Stanford
  "Provably Secure Machine Learning"
- Fr, Mar 2, 12:15pm, Gates G01: Wei-Lun (Harry) Chao, USC
  "Transfer learning towards intelligent systems in the wild"
- Th, Mar 8, 4:15pm, Gates G01: Bo Zhu, MIT
  "Exploring and Understanding Limits of Physical Systems"
- Fr, Mar 23, 12:15pm, Gates G01: Jesse Thomason, UT Austin
  "Continuously Improving Embodied Natural Language Understanding through Human-Robot Conversation"

Greedy Search (aka Hillclimbing)

```
Greedy(s,problem)
    current = s
    loop
        S = {s' | a∈problem.actions(current) and s'=child(a,current)}
        if there exists node∈S such that problem.goal(node)
            then return problem.solution(node)
            else
                node = best(S)
                if problem.f(node) is better than problem.f(current)
                    then current = node
                    else return problem.solution(current)
```

Simulated Annealing Search

```
SA(s,problem)
    current = s
    loop
        S = {s' | a∈problem.actions(current) and s'=child(a,current)}
        if there exists node∈S such that problem.goal(node)
            then return problem.solution(node)
            else
                if rand(0:1) < problem.temp then node = random(S)
                else
                node = best(S)
                if problem.f(node) is better than problem.f(current)
                    then current = node
                    else return problem.solution(current)
```

Genetic Algorithms

Vocabulary:

- Fitness function: Evaluation function f(x)
- Population: Set of states
- Two actions:
    - Mutation: States $\rightarrow$ States
    - Crossover: States x States $\rightarrow$ States
- Selection: Picking states to which actions are applied

Genetic Algorithm Template

```
GA(populationsize,mutationrate):
    S = a set of states of size populationsize /* the "initial population" */
    Loop
        S' = {}
        for i = 1 to populationsize
            x = selectfit(S)
            y = selectfit(S)
            new = crossover(x,y)
            if rand(0:1) < mutationrate then new = mutate(new)
            S' = S' + new
```

selectfit(S): /* example of how to select an element biased by their f values */

totalfitness = $\sum_{s \in S} \mathrm{f(s)}$

Select an element of S at random, each with probability $\frac{\mathrm{f(s)}}{\mathrm{totalfitness}}$.

/* this converts the f values into a probability distribution */