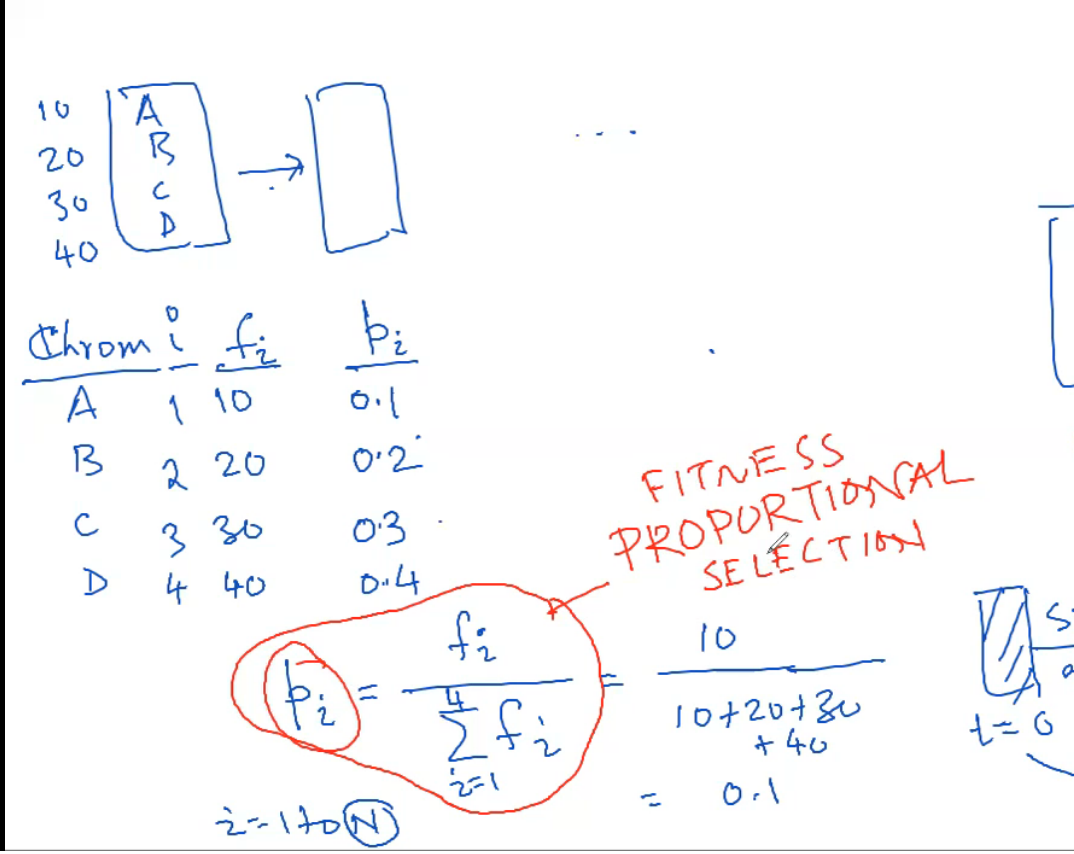
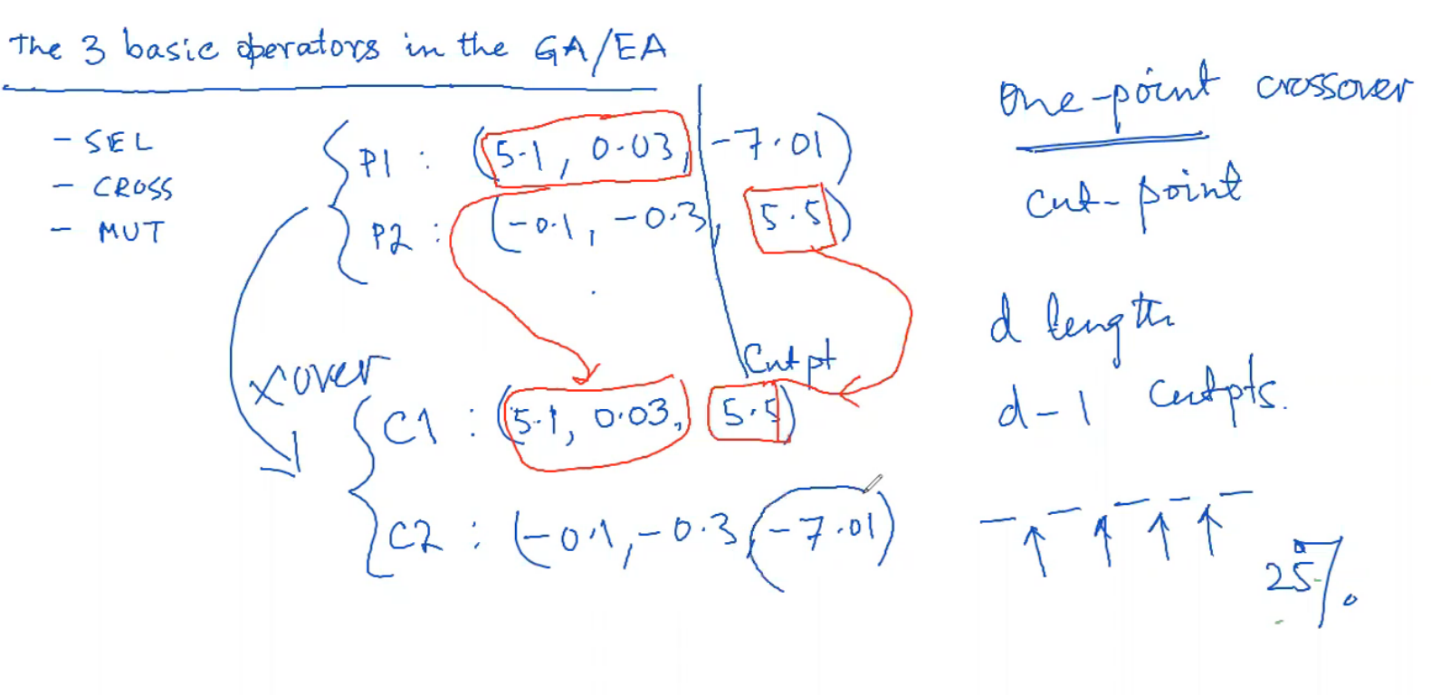
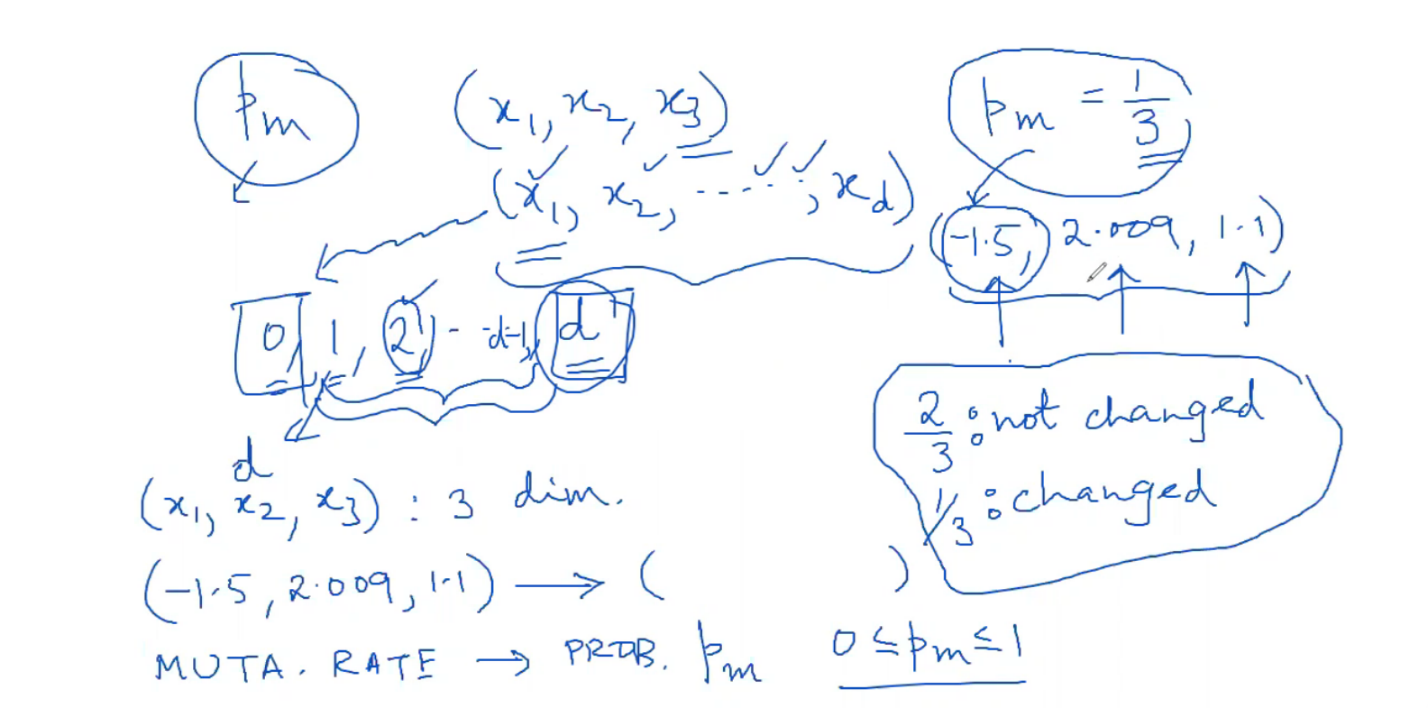
Proportional selection

Use rand from 0-1 and see which p values covers that number.

Single point crossover



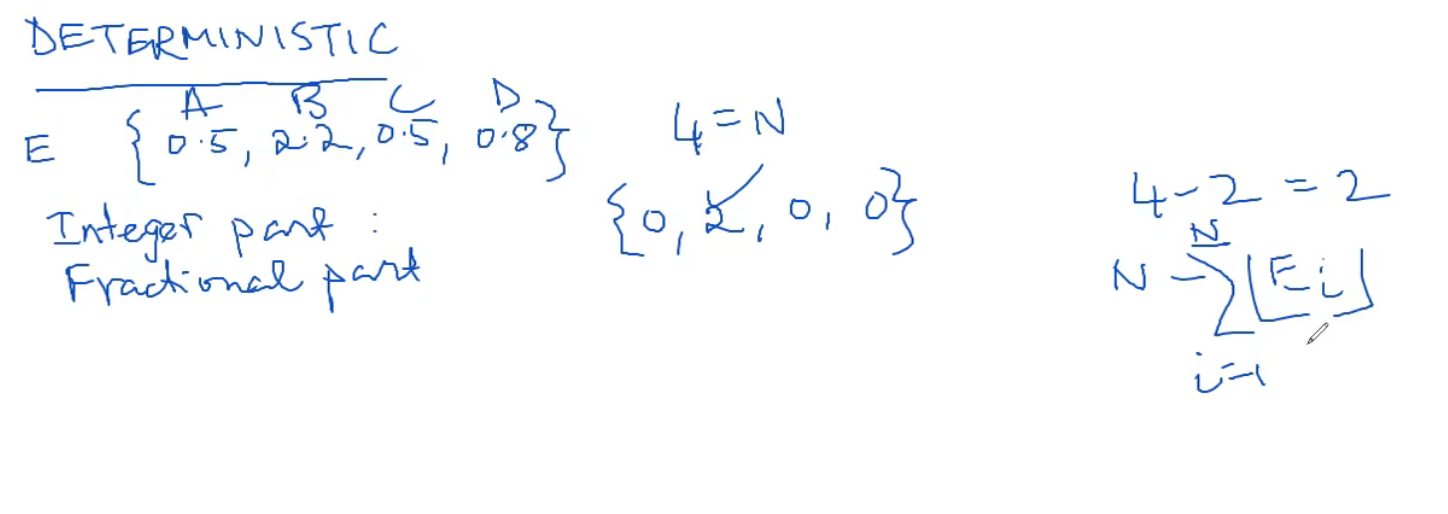
Genewise mutation



Stochastic universal sampling

Normalize the f values(f/n). Then generate a uniformly randomly generated probe. This probe is the first probe to be generated(u(0,1)) and all other probes are all equidistant from each other. When hitting boundaries, wrap around back to 0. The distance between each probe is 1/n.

Deterministic sampling

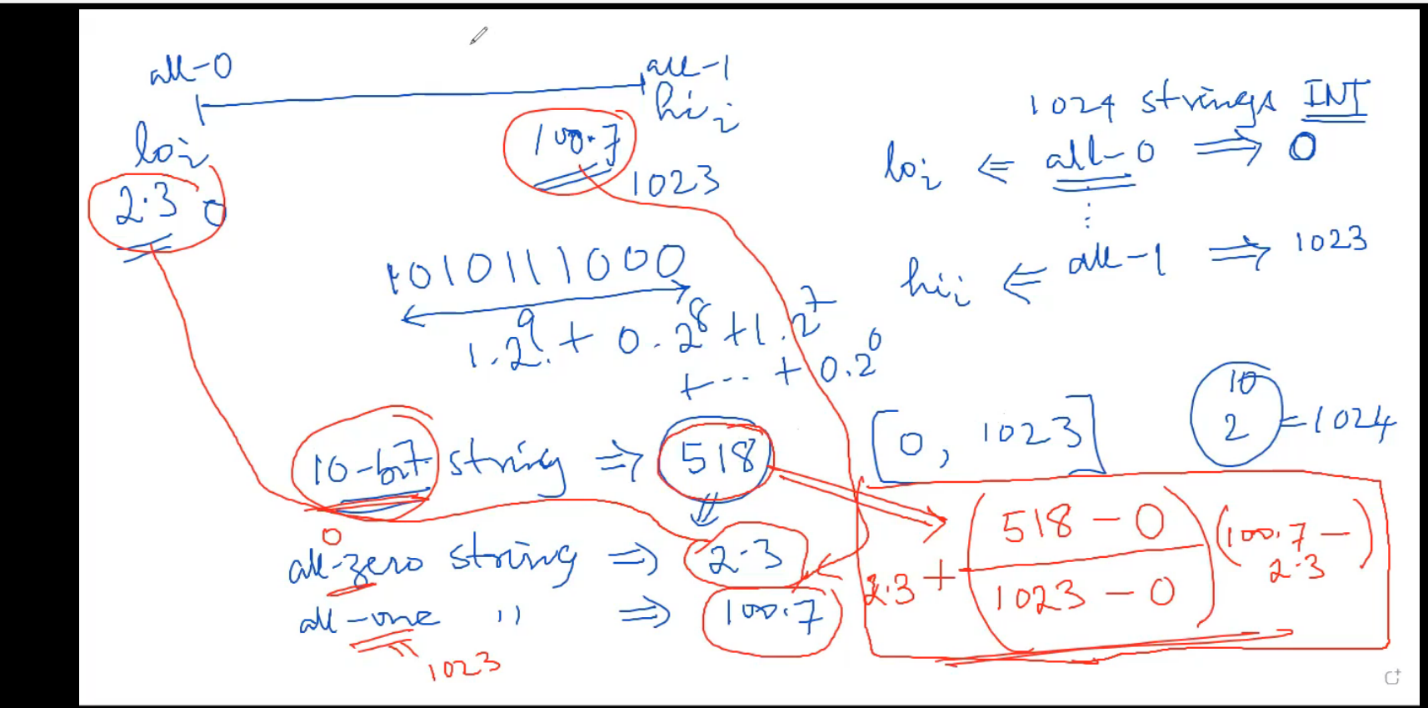


Ei = pi\*n. Take all integer parts of Ei, add them up and subtract total from n. This gives you the number of individuals that need to be filled(r).

Sort the population. Find the highest (r) numbers. If one of those numbers have multiple instances, choose one randomly

Bit string

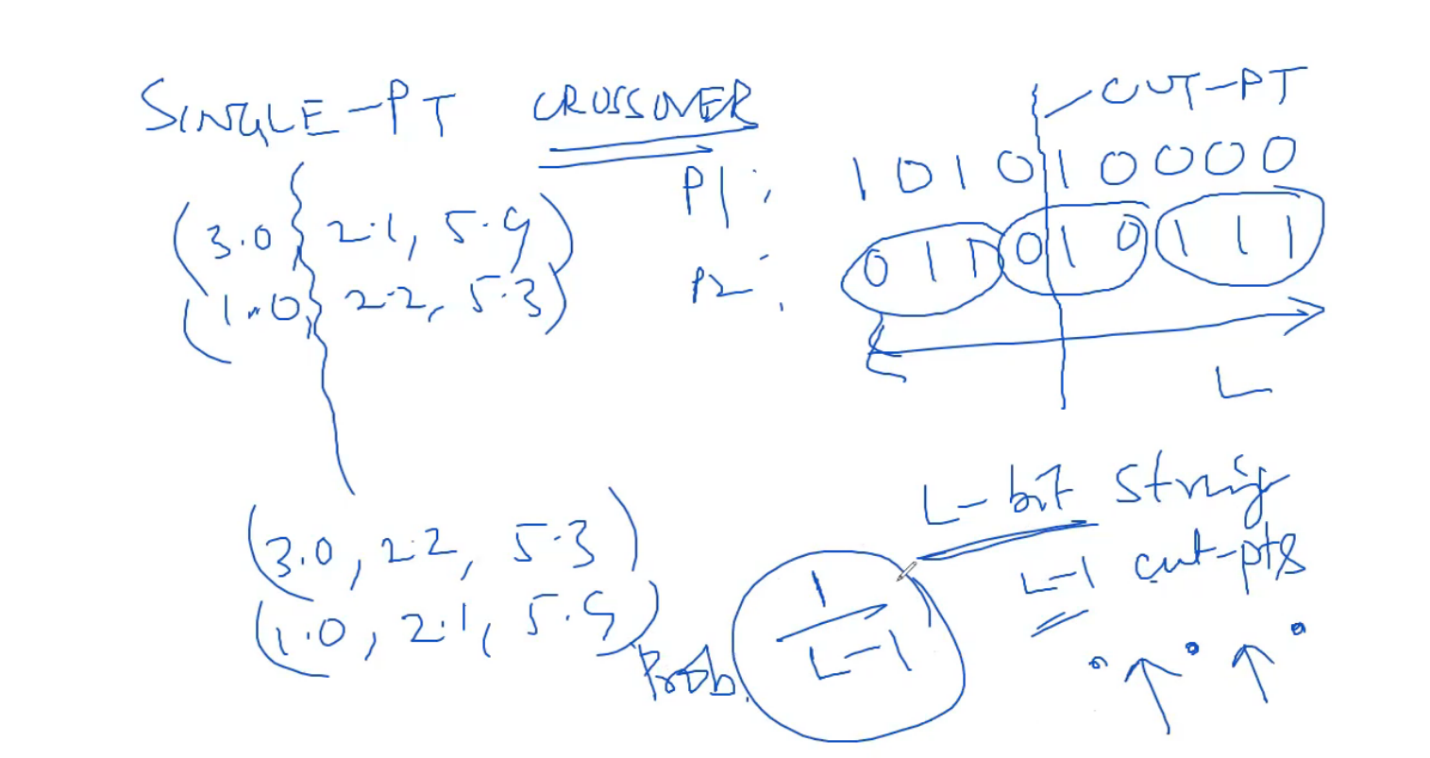
Evaluate fitness



L bits make up one individual so L=(x1x2x3), ex 1010110

Selection sampling- decode the bits into xi for fitness calculations

Single point crossover- probability of point is 1/L-1

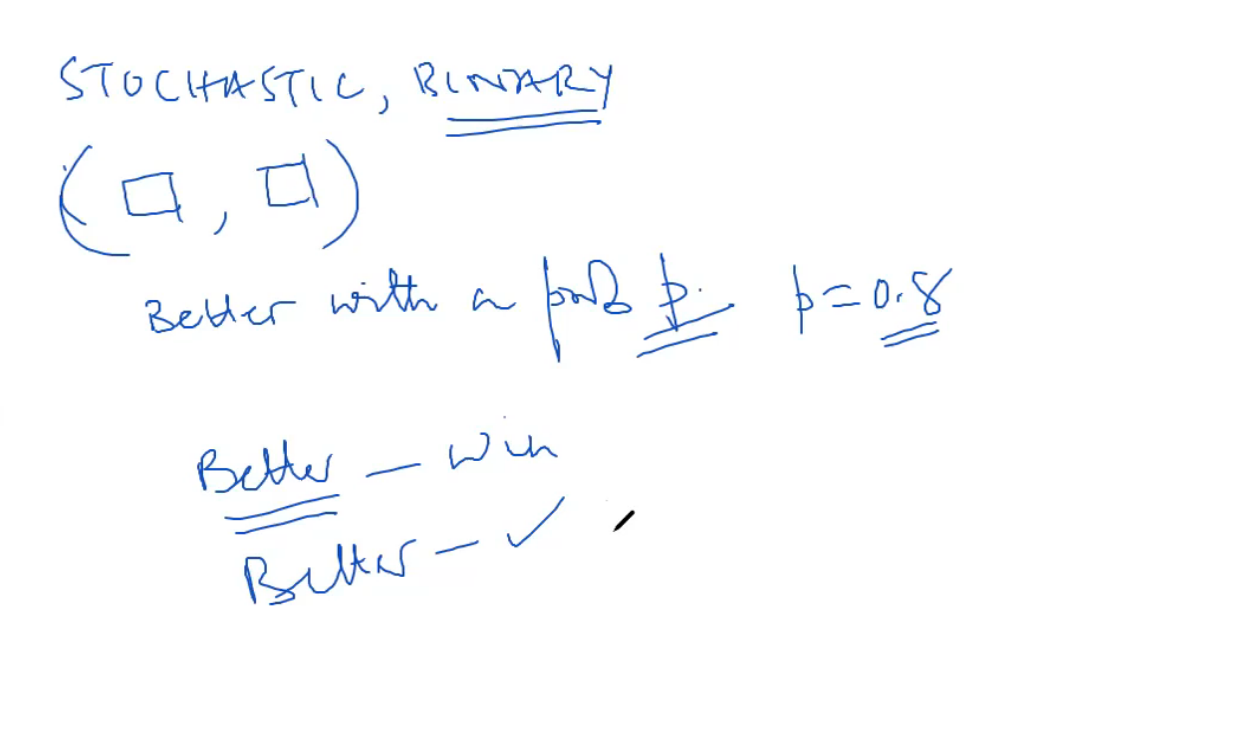


Mutation- bit wise mutation. Each bit will have a probability to mutate

**Proportional selection**

**Linear ranking selection**- sort population before selection

**Stochastic binary tournament**



This tournament is choosing two numbers from the population. Out of the two, find the better number. There is a 80%(p=.8) chance that the better number will go into selection. And a 20% that the smaller number will not.