Pictorial representation of evolution via mutation and selection

Evolution by updating whole population every generation

A-type B-type

Time=1 Copy parent pop. to offspring pop.

Time=1 Offspring pop. after **mutations** ➔

mutations

Time=1 Pick a Random member

Suppose RN=3

Not OK

Offspring pop. Array (O)

B picked for reproduction

Compare with fitness for reproduction: Pick RN between 0&1. If RN< normalized_fitness_of_B, then OK

normalized_fitness_of_B = f1/(f0+f1)

OK

Time=2

B becomes the second member of the Offspring pop. at t=2

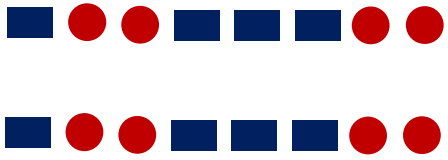
Parent pop. Array (P)

B becomes the second member of the Offspring pop. at t=2

Repeat above steps to fill up the parent pop. at t=2

Copy offspring array at t=2 to parent array such that it becomes the parent pop. For choosing offspring at t=3

Copy



Offspring pop. Array (O) at t=2

Parent pop. Array (P) at t=3

Evolution via mutation *and* selection (Proof of Error Threshold): Algorithm

Create two arrays one for the parent population and one for the offspring population, each of size N. Both the arrays should initially contain N 0's and 0 1's.

Start Loop over generations (Total = T)

Loop over populations (Size=N)
Copy parent population array to offspring population array to make mutations in.
Close loop over populations.

Start new loop over offspring population
Generate a uniformly distributed random number x lying between 0 and 1
If type 0 : if x < u change individual from 0 --> 1
Close loop over offspring population.

Start new loop over parent population
Select parent population for the next time-step from the offspring population in this time-step with a probability proportional to the fitness of the offspring i.e.
Generate another uniformly distributed random number RN2 lying between 1 and N
Generate a random number z between 0 and 1
If **P[RN2]=0** and **z < f0/(f0+f1)** replace P[RN1] by 0
If **P[RN2]=1** and **z < f1/(f0+f1)** replace P[RN1] by 1
Else Generate new RN2 by picking a random no. between 1 and N
Continue until new parent population is generated by selection of all N members
Close loop over parent population.

Start new loop over population
Calculate frequency x0 of type 0 in the population.
x1=1-x0 – frequency of type 1 in the population.
Close loop over population.

Write frequencies x0 and x1 vs time in a file.

Close loop over generations