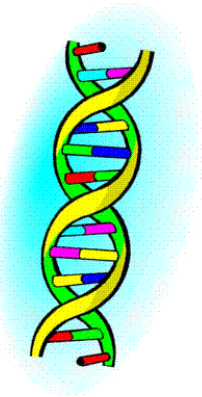
Genetic Algorithms

- **Chromosome**
  - An encoded representation of a candidate solution to the problem (typically a sequence of numbers or bits)
- **Population of chromosomes**
  - A pool of candidate solutions, initially generated at random
- **Fitness function**
  - $f(chromosome) \Rightarrow$ numerical estimate of observed quality
- **Operators**
  - *Selection:* survival of the fittest
  - *Crossover:* genetic recombination
  - *Mutation:* random variation



Outline of a Genetic Algorithm

1. Create a new population of random chromosomes

000011101001100

110010011110011

100110100110100

000001001000010

111101011111101

101001101010101

000010000101011

110100100010001

Outline of a Genetic Algorithm

1. Create a new population of random chromosomes
2. Evaluate the fitness of each chromosome in the population

$f(000011101001100) = 6$

$f(110010011110011) = 9$

$f(100110100110100) = 7$

$f(111101011111101) = 12$

$f(000001001000010) = 3$

$f(101001101010101) = 8$

$f(000010000101011) = 5$

$f(110100100010001) = 6$

Outline of a Genetic Algorithm

1. Create a new population of random chromosomes
2. Evaluate the fitness of each chromosome in the population
3. Repeat:

   (a) pick 2 chromosomes probabilistically, based on fitness

$f(000011101001100) = 6$

$f(110010011110011) = 9$

$f(10011010011010 0) = 7$

$f(111101011111101) = 12$

$f(000001001000 10) = 3$

$f(101001101010101) = 8$

$f(000010000101011) = 5$

$f(110100100010001) = 6$

Outline of a Genetic Algorithm

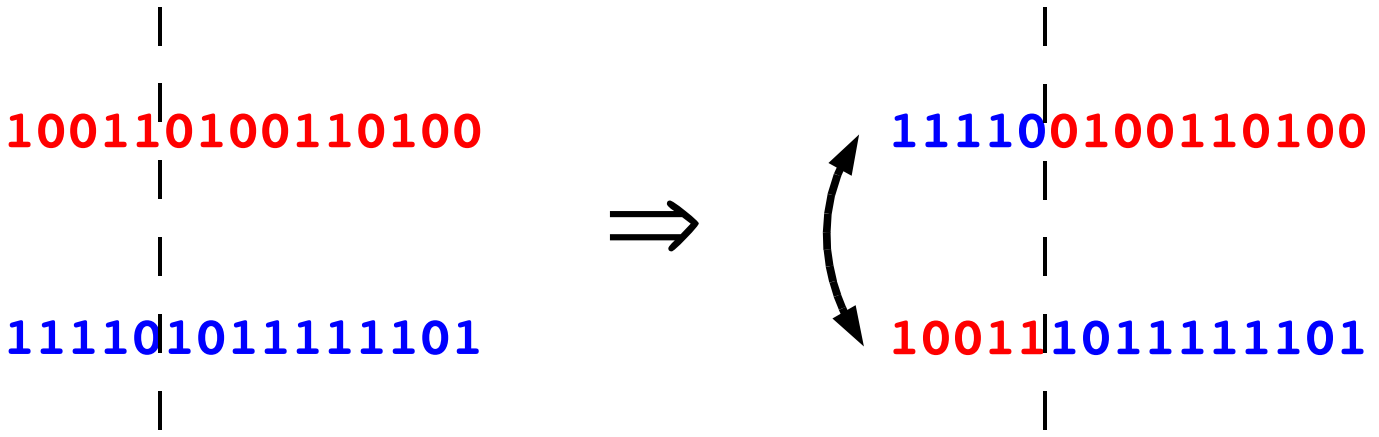
1. Create a new population of random chromosomes
2. Evaluate the fitness of each chromosome in the population
3. Repeat:
   (a) pick 2 chromosomes probabilistically, based on fitness
   (b) create 2 new offspring from them, using crossover

100110100110100

111101011111101

Outline of a Genetic Algorithm

1. Create a new population of random chromosomes
2. Evaluate the fitness of each chromosome in the population
3. Repeat:
   (a) pick 2 chromosomes probabilistically, based on fitness
   (b) create 2 new offspring from them, using crossover



Outline of a Genetic Algorithm

1. Create a new population of random chromosomes
2. Evaluate the fitness of each chromosome in the population
3. Repeat:
   (a) pick 2 chromosomes probabilistically, based on fitness
   (b) create 2 new offspring from them, using crossover

111100100110100

100111011111101

Outline of a Genetic Algorithm

1. Create a new population of random chromosomes
2. Evaluate the fitness of each chromosome in the population
3. Repeat:
   (a) pick 2 chromosomes probabilistically, based on fitness
   (b) create 2 new offspring from them, using crossover
   (c) mutate each offspring with some small probability

1111000001101 10

0 00111011111101

Outline of a Genetic Algorithm

1. Create a new population of random chromosomes
2. Evaluate the fitness of each chromosome in the population
3. Repeat:
   (a) pick 2 chromosomes probabilistically, based on fitness
   (b) create 2 new offspring from them, using crossover
   (c) mutate each offspring with some small probability
   (d) add the offspring to the new generation
   (e) when the new generation has reached the same size as the current population, replace the current population by the new generation
   (f) evaluate the fitness of each chromosome in the new population, and continue

Outline of a Genetic Algorithm

- Over time, the **average fitness** of the population will increase
- The best-fit chromosomes are not guaranteed to survive to the next generation
- Some GAs use **elitism** to ensure that the best chromosomes do survive
- Even the worst-fit chromosomes have some (small) probability of surviving
- Many ways of probabilistically selecting chromosomes
  - fitness-proportionate selection (“roulette-wheel sampling”)
  - rank selection
  - tournament selection