Genetic Algorithm Search

Genetic Algorithms

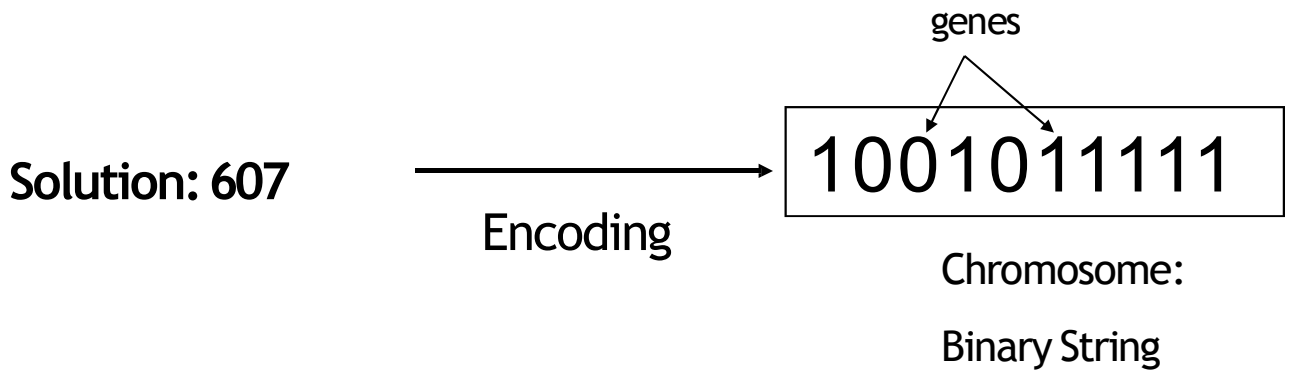
- Formally introduced in the US in the 70s by John Holland.
- GAs emulate ideas from genetics and natural selection and can search potentially large spaces.
- Before we can apply Genetic Algorithm to a problem, we need to answer:

  - How is an individual represented?
  - What is the fitness function?
  - How are individuals selected?
  - How do individuals reproduce?

Genetic Algorithms:

Representation of states (solutions)

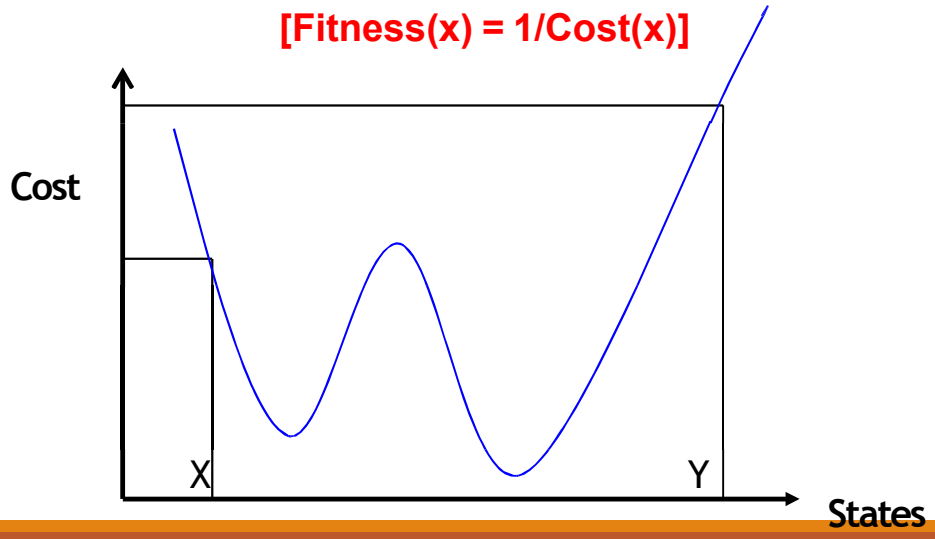
- Each state or individual is represented as a string over a finite alphabet. It is also called **chromosome** which Contains **genes**.



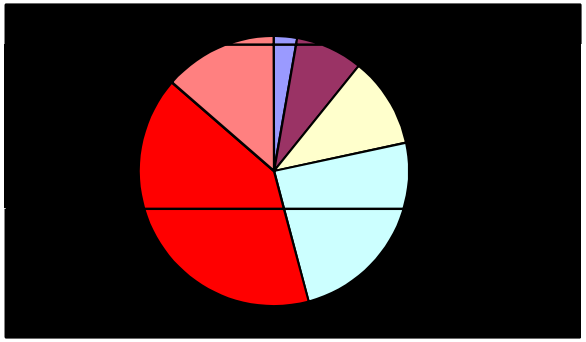
Genetic Algorithms: Fitness Function

- Each state is rated by the evaluation function called **fitness function**. Fitness function should return higher values for better states:

Fitness(X) should be greater than Fitness(Y) !!



GA Parent Selection - RouletteWhee



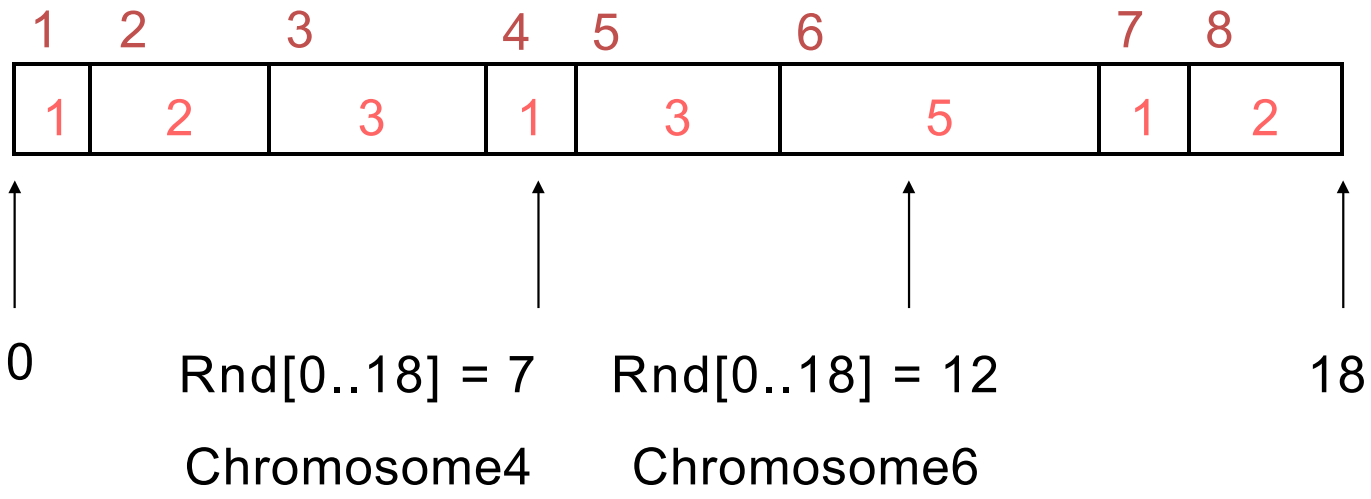
Roulette Wheel Selection

- Sum the fitnesses of all the population members, TF
- Generate a random number, m, between 0 and TF
- Return the first population member whose fitness added to the preceding population members is greater than or equal to m

Genetic Algorithms: Selection

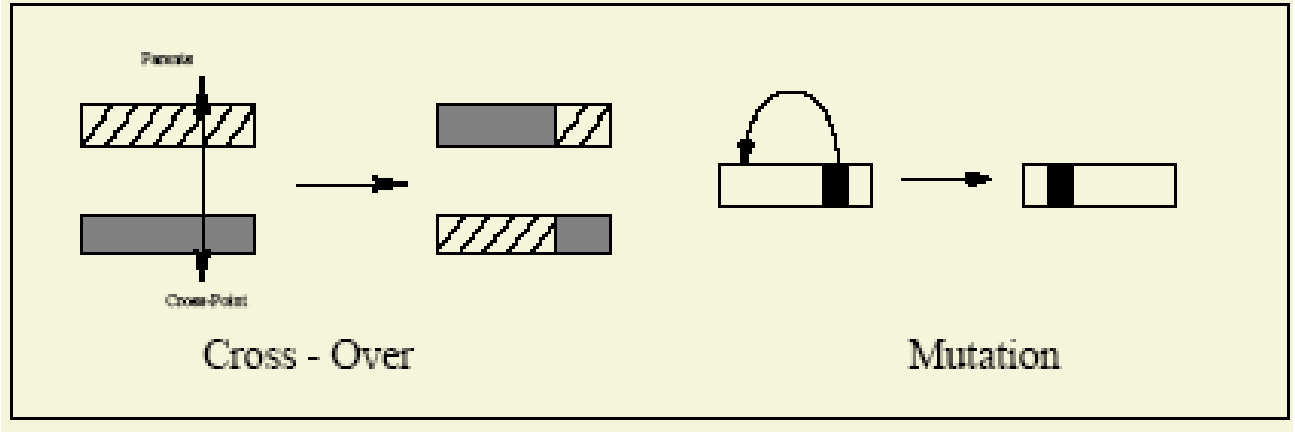
How are individuals selected?

Roulette Wheel Selection

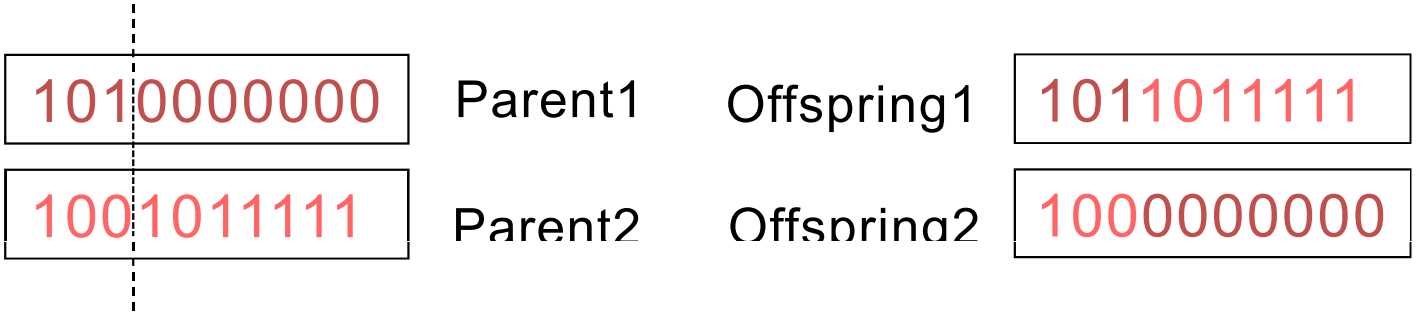


Genetic Algorithms: Cross-Over and Mutation

How do individuals reproduce ?

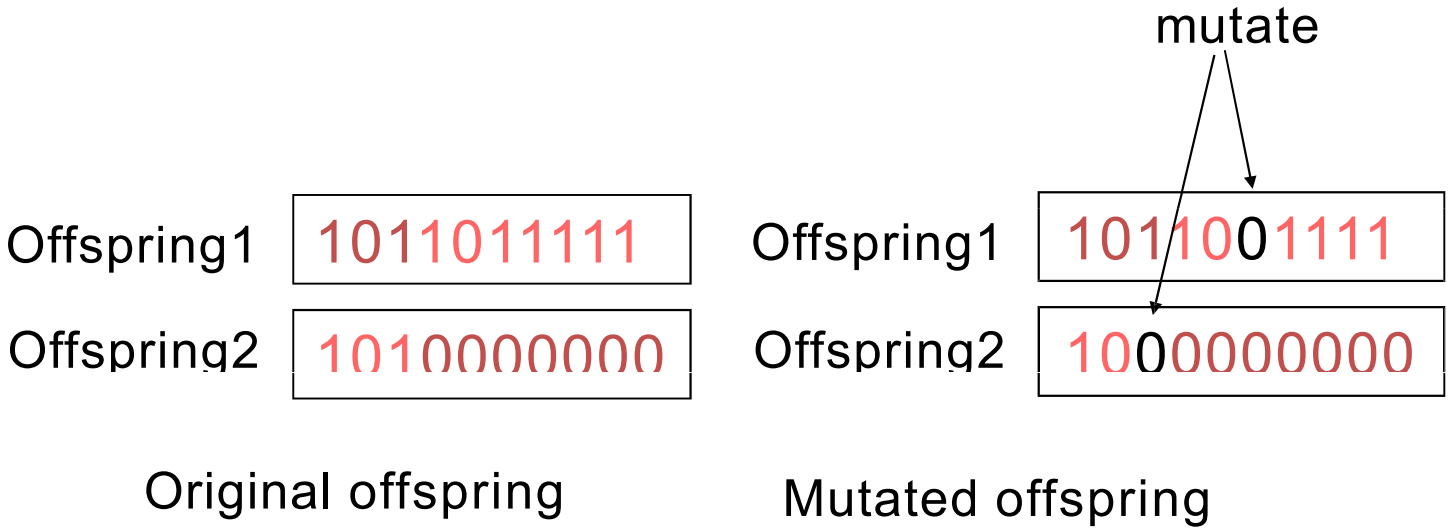


Crossover - Recombination



With some high probability (*crossover rate*) apply crossover to the parents. (*typical values are 0.8 to 0.95*)

Stochastic Search: Genetic Algorithms
Mutation



With some small probability (the *mutation rate*) flip each bit in the offspring (*typical values between 0.1 and 0.001*)

Genetic Algorithms

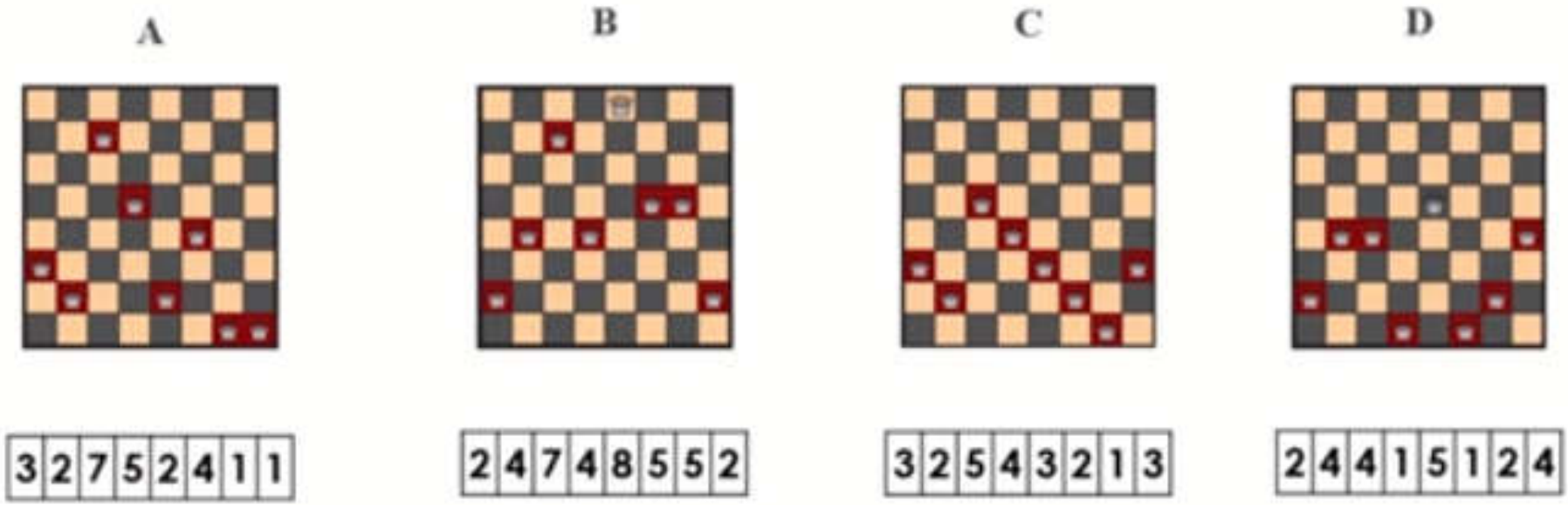
Algorithm:

1. Initialize population with **p** Individuals at random
2. For each Individual **h** compute its fitness
3. While max fitness < threshold
   do Create a new generation Ps
4. Return the Individual with highest fitness

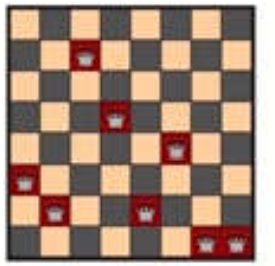
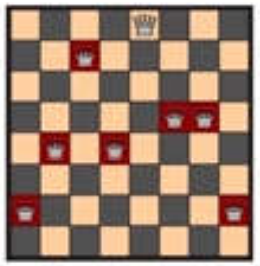
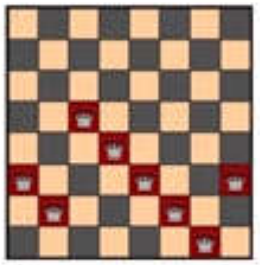
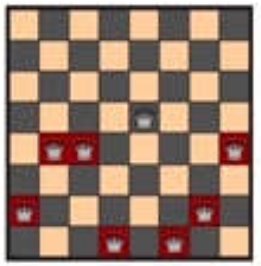
Genetic algorithms

Has the effect of "jumping" to a completely different new part of the search space (quite non-local)

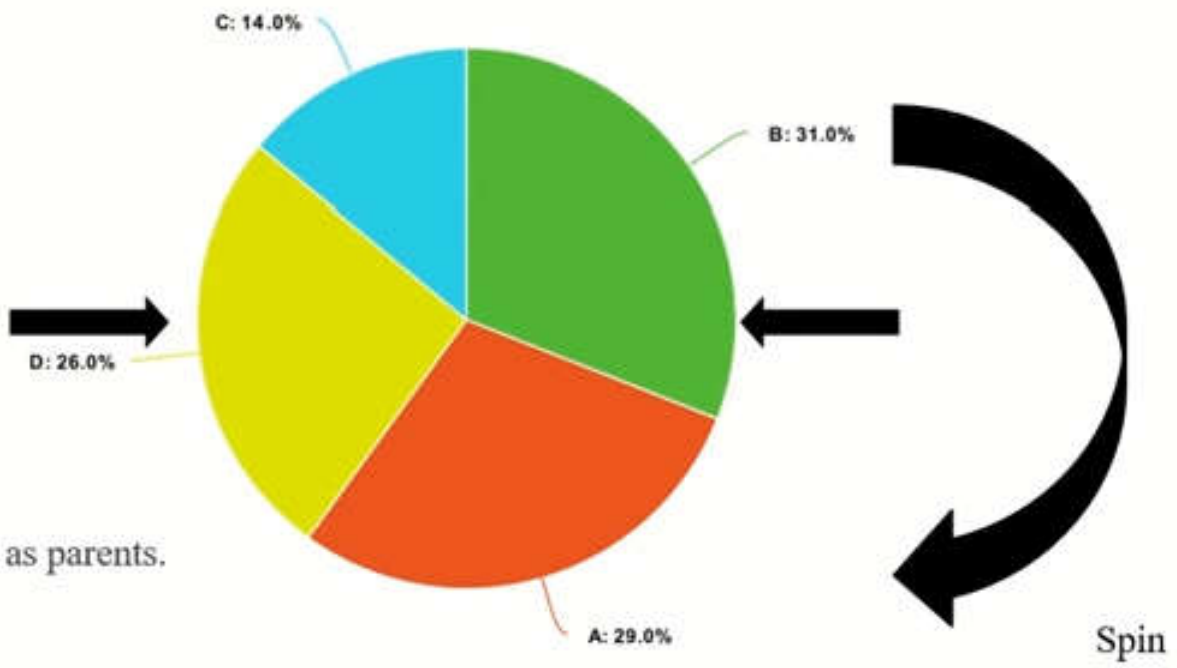
Genetic algorithms



Genetic algorithms – Fitness Calculation

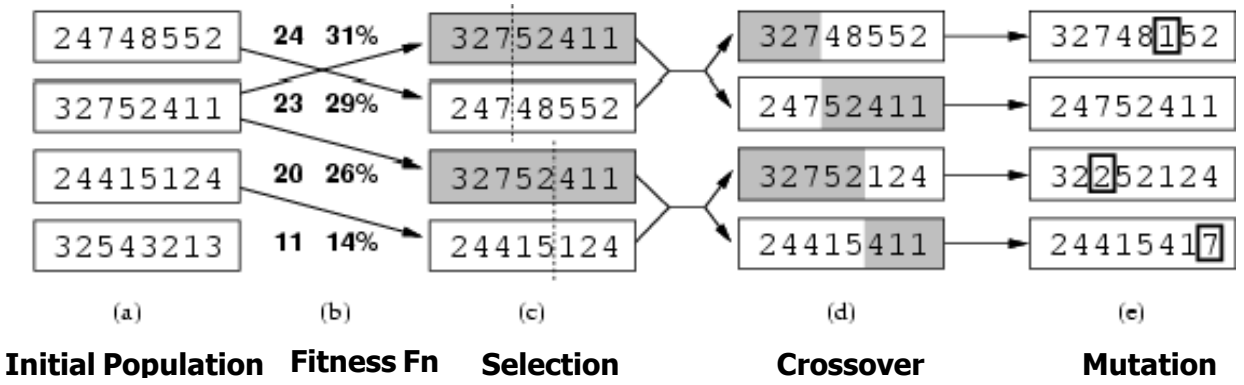
	A	B	C	D
Individuals	 3 2 7 5 2 4 1 1	 2 4 7 4 8 5 5 2	 3 2 5 4 3 2 1 3	 2 4 4 1 5 1 2 4
Fitness	23	24	11	20
Fitness %	29%	31%	14%	26%

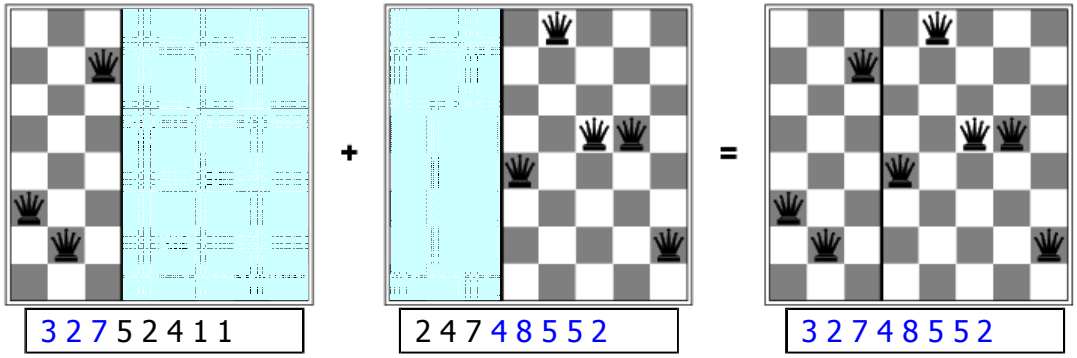
Genetic algorithms – Parent Selection (Stochastic Universal Sampling)



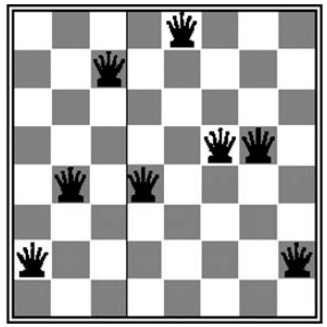
- B and D are selected as parents.

Genetic Algorithm (cont.)

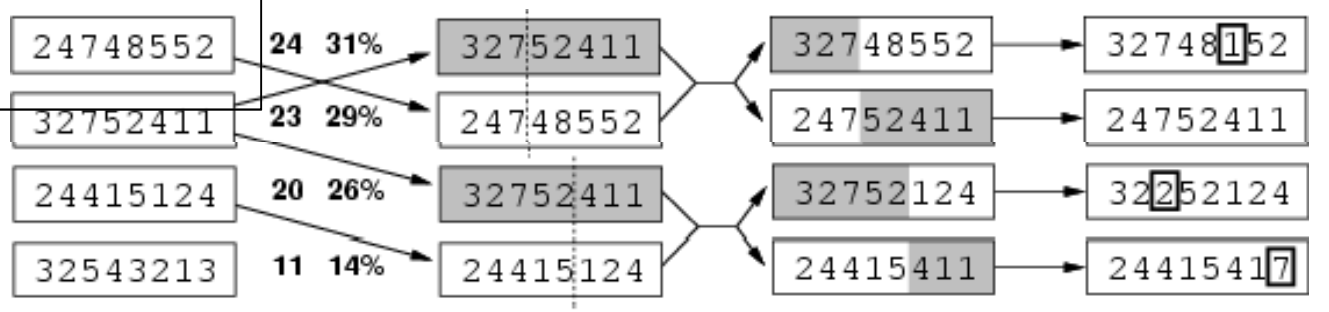




Genetic Algorithm



Operate on state representation.



production of next generation

random mutation

Fitness function: number of **non-attacking** pairs of queens (min = 0, max = 8 × 7/2 = 28 ☺ the higher the better)

24/(24+23+20+11) = 31%

23/(24+23+20+11) = 29% etc

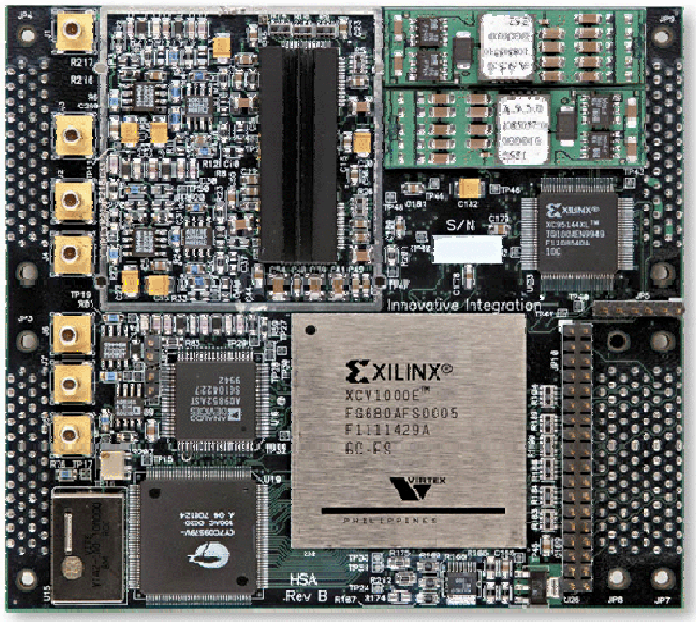
crossover point randomly generated

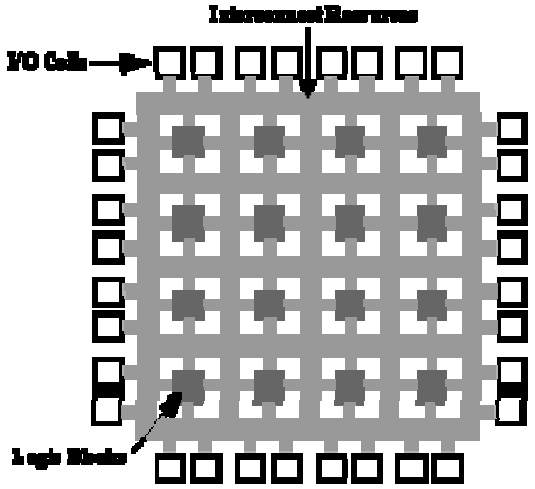
probability of a given pair selection proportional to the fitness (b)

GA is a good *no clue* approachto problem solving

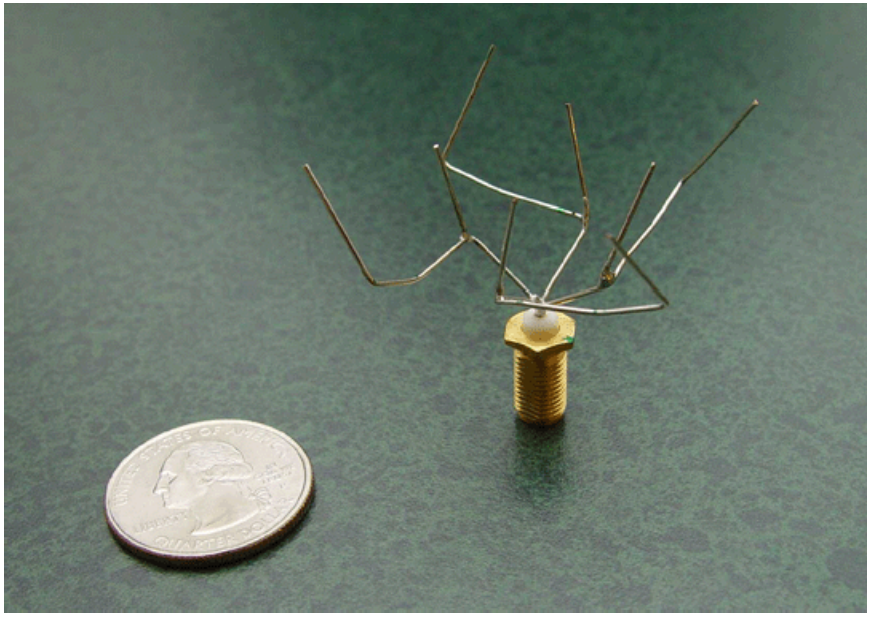
- GA is superb if:
  - Your space is loaded with lots of weird bumps and local minima.
    - GA tends to spread out and test a larger subset of your space than many other types of learning/optimization algorithms.
  - You don't quite understand the underlying *process* of your problem space.
  - You have lots of processors
    - GA's parallelize very easily!

Evolvable Circuits



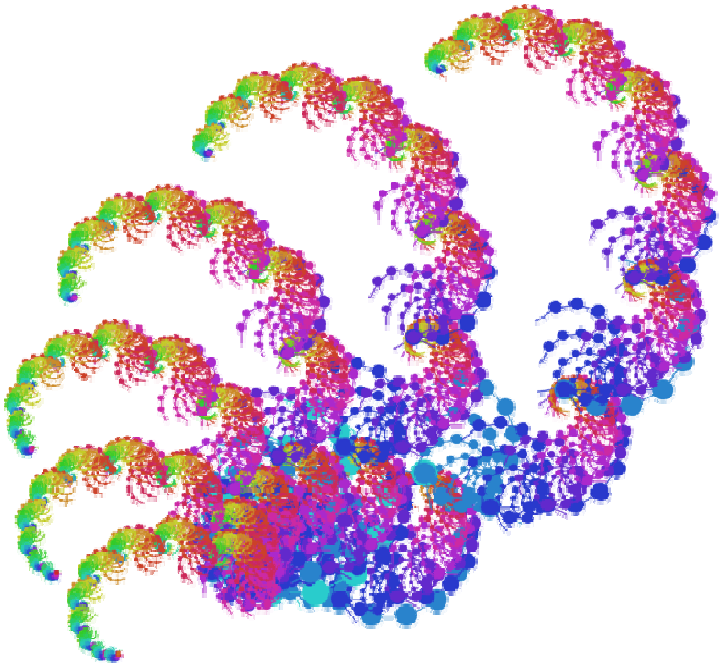


Antenna for NASA



Car Design

Evolutionary Arts



What is the major challenge?

Evolving Mona Lisa