# Genetic Example:

## Problem: find over [0…31]

* Data presentation: binary code
* Fixed population size: 4
* Evolution: roulette wheel selection, 1-point crossover, bitwise mutation
* Run 1 one circle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Initialization | | | Selection | | |
| Initial Population | Value x | Fitness f(x)= | Probability | Expected Count | Actual Count |
| 0 1 1 0 1 | 13 | 169 | 0.14 | 0.58 | 1 |
| 1 1 0 0 0 | 24 | 576 | 0.49 | 1.97 | 2 |
| 0 1 0 0 0 | 8 | 64 | 0.06 | 0.22 | 0 |
| 1 0 0 1 1 | 19 | 361 | 0.31 | 1.23 | 1 |
| Max | 576 | Average | 293 |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Crossover | | | | |
| Population after selection | Cross-over point | Offsprings | Value x | Fitness f(x)= |
| 0 1 1 0 1 | 4 | 0 1 1 0 0  1 1 0 0 1 | 12 | 144 |
| 1 1 0 0 0 | 25 | 625 |
| 1 1 0 0 0 | 2 | 1 1 0 1 1  1 0 0 0 0 | 27 | 729 |
| 0 1 0 0 0 | 16 | 256 |
| Max | 729 | Average | 439 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Mutation with probability | | | |
| Population after crossover | Offsprings | Value x | Fitness f(x)= |
| 0 1 1 0 0 | 0 1 1 0 0 | 12 | 144 |
| 1 1 0 0 1 | 1 1 **1** 0 1 | 29 | 841 |
| 1 1 0 1 1 | **0** 1 0 1 1 | 11 | 121 |
| 1 0 0 0 0 | 1 0 0 0 0 | 16 | 256 |
| Max | 841 | Average | 340 |

Comment: Cross-over or mutation can both give better or worse offsprings but it give us a chance to find best elitism. Even in last mutation, our average population decrease and we obtain better result, it shows that loss of good population is possible and bigger population means lower chances to degenerate it.

## Hamming Cliff problem

* Problem Statemetn: for x = [-16,16]
* Data presentation:
* \*\*Optimum point , second best
  + Cross-over: cannot generate from good parent
  + Mutation: cannot flip all positions
* \*\* Neighbour:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

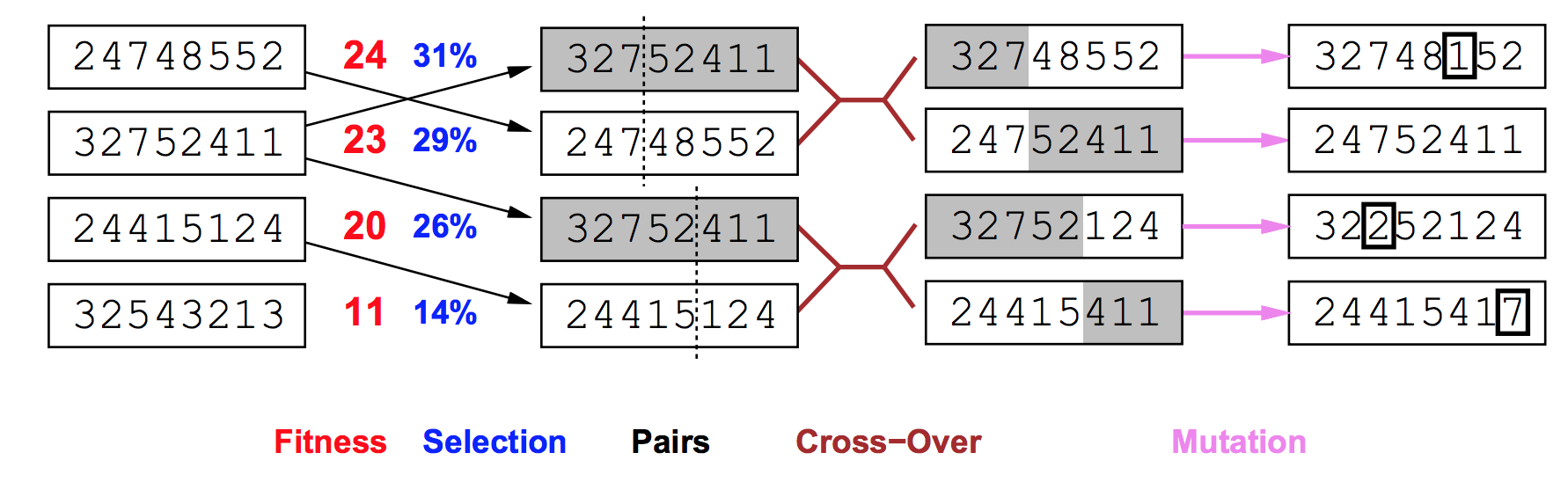
* + Neighbours have poor performance
* Problem solve by using new data presentation then g(i) and g(i+1) only differ 1 bit

## 8-Queen problem

* Problem: place 8 queens on a chessboard such that no queen attacks any others



* Data presentation: State/Position is position of each queen in each column
  + Ex: [46827135] represent a case in image above
* Fitness function: number of non-attacking queens (min=0, max=28)
  + Higher value better result



* Cross-over: swap segment
* Mutation: invert random segment (choose segment and permutation), switch 2 random positions or change it

# 2 Support theorem and some example for this algorithm

## Schema:

Definition:

* present a subset of D
* s contains chromosomes. = number of defined bit and is length of schema
* each chromosome belong to schemas

## Schema Theorem

* provide estimate of number of schemas during evolution (selection, cross-over and mutation) to analyse how chromosomes vary in schema
* Definition:
  + Population **P**, size **n**, at time **t**, schema **s**.
  + **m(s,t)** = number of chromosomes of **s** in **P** at time **t**

### Selection:

* probabilistic selection:
  + let (expect value of F over P and s)
* Assume that we choose only 1 best chromosome. Our current population is
  + Expect value of number of chromosomes after 1 selection
  + Expect value of number of chromosomes after n selections
  + Expect value of number of chromosomes of s after n selections
  + So if we have good schema, ratio and then number of good chromosomes increases

### Crossover:

* We only try applying 1 point crossover with probability that means (1- ) population (after selection step) will be in
  + **max distance** between defined binary values
  + suppose belongs to schema s:



* + - if i (a cross-over point) is outside d(s): schema preserved
    - if I is inside d(s): possible that both offspring are not in s. Schema can be destroyed.
  + We will find numbers of survive chromosomes after cross-over
    - Probability that a chromosome of s does not produce a chromosome of s is
    - E(number of chromosomes of s are destroyed)
    - Expect value of number of chromosomes of s after cross-over:

### Mutation:

* Each chromosome will be mutated with probability
  + Survival probability (probability that each bits should not be mulated)
  + Expect value of number of chromosomes of s in new population

### Conclusion:

* are less than 1, so quality of schema at beginning is quite important so that

## The Argument

* Theorem: Under reasonable assumptions, random population of size N **sample schemas** ( 100 chromo => 10^6 schemas)
* The proof: It is not very useful to understand this. You can look for it online
* Intuitive interpretation:
  + Increase a population, we also sample more schema so that we have greater chances to meet “good” schema
  + Since good schema is created during evolution, we have greater chance to get global maximum