# Intelligent Systems Principles and Programming

Xiaoqing Zheng
zhengxq@fudan.edu.cn


# Evolution

*"... no limit to this power of slowly and beautifully adapting each form to the most complex relations of life ... "*

——— ***Charles Darwin***

# Example

**Maximum** $f(x) = x^2$, $x \in [1, 31]$

- **Representation**

  $x \in \{0,1\}^5$

- **Initialization**

| | | | | |
|---|---|---|---|---|
| 1st generation | 01101, | 11000, | 01000, | 10011 |
| Interpretation | 13, | 24, | 8, | 19 |
| Fitness | 169, | 576, | 64, | 361 |

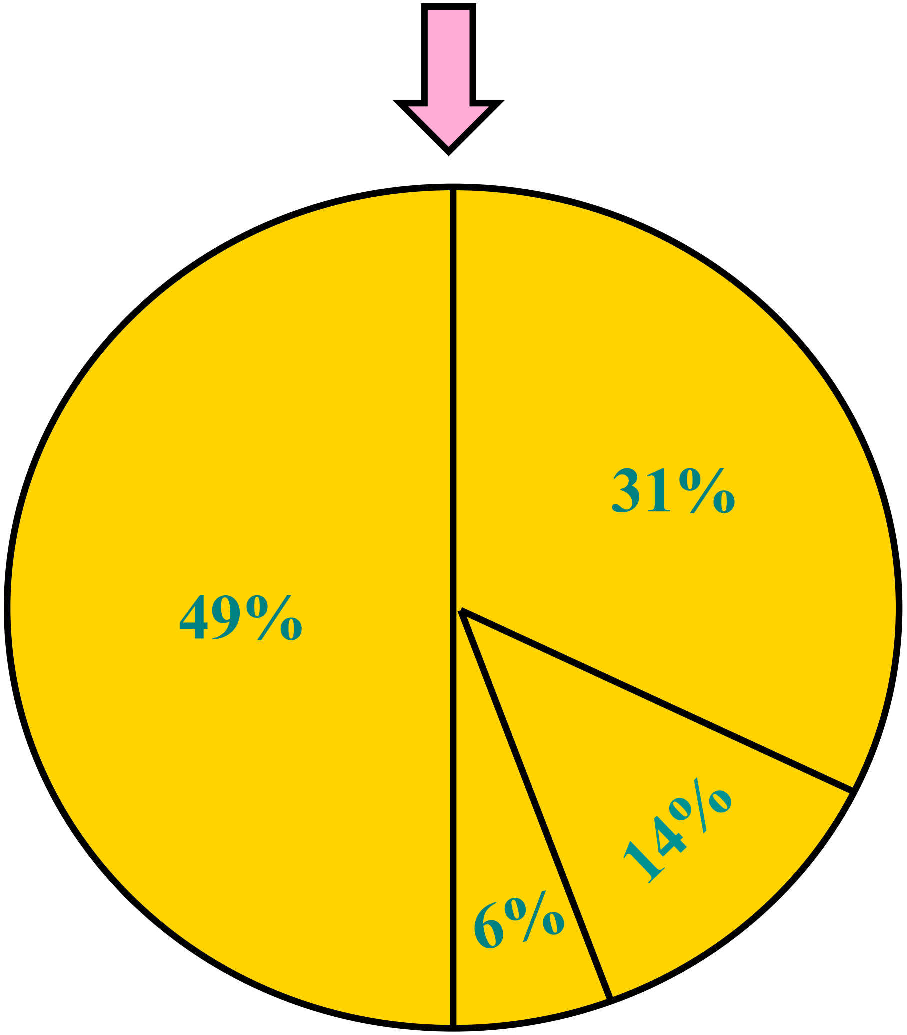
# Example

**Maximum** $f(x) = x^2$, $x \in [1, 31]$

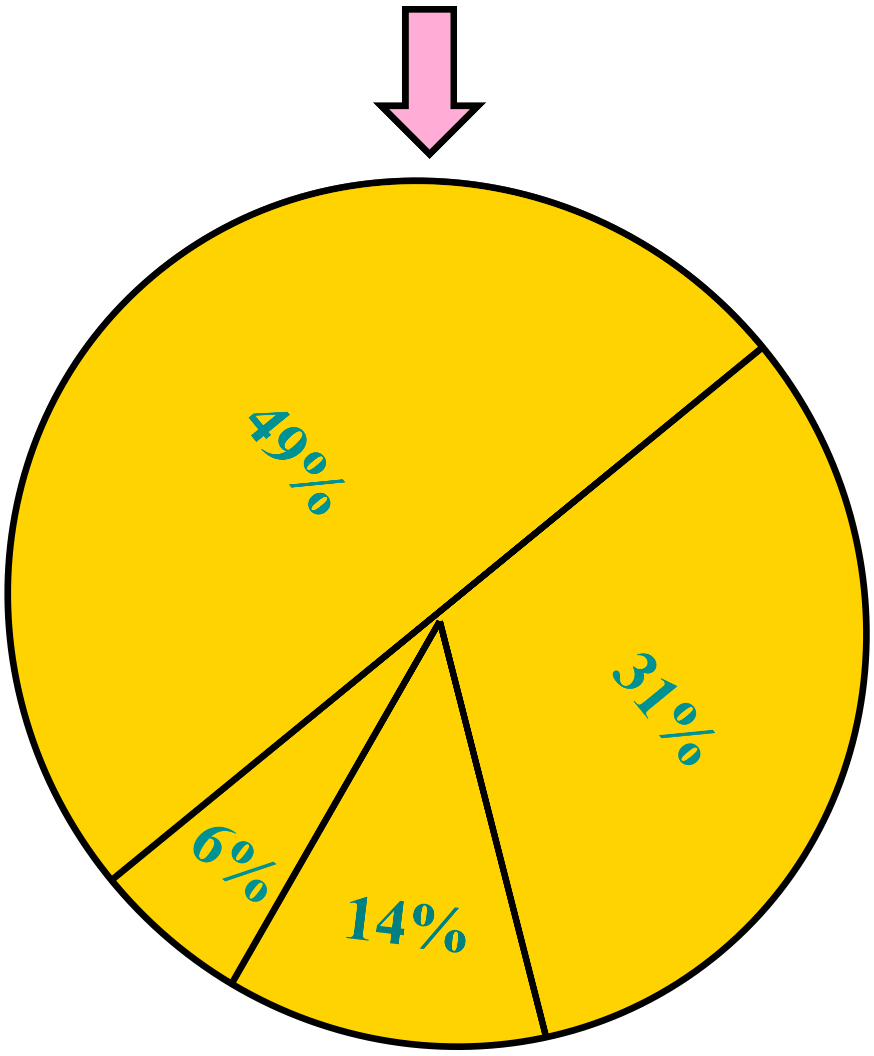
- **Selection**

| | |
|---|---|
| Individual | 01101, 11000, 01000, 10011 |
| Fitness | 169, 576, 64, 361 = 1170 |
| Probability | 0.14, 0.49, 0.06, 0.31 = 1.0 |
| Result | 01101, 11000, 11000, 10011 |

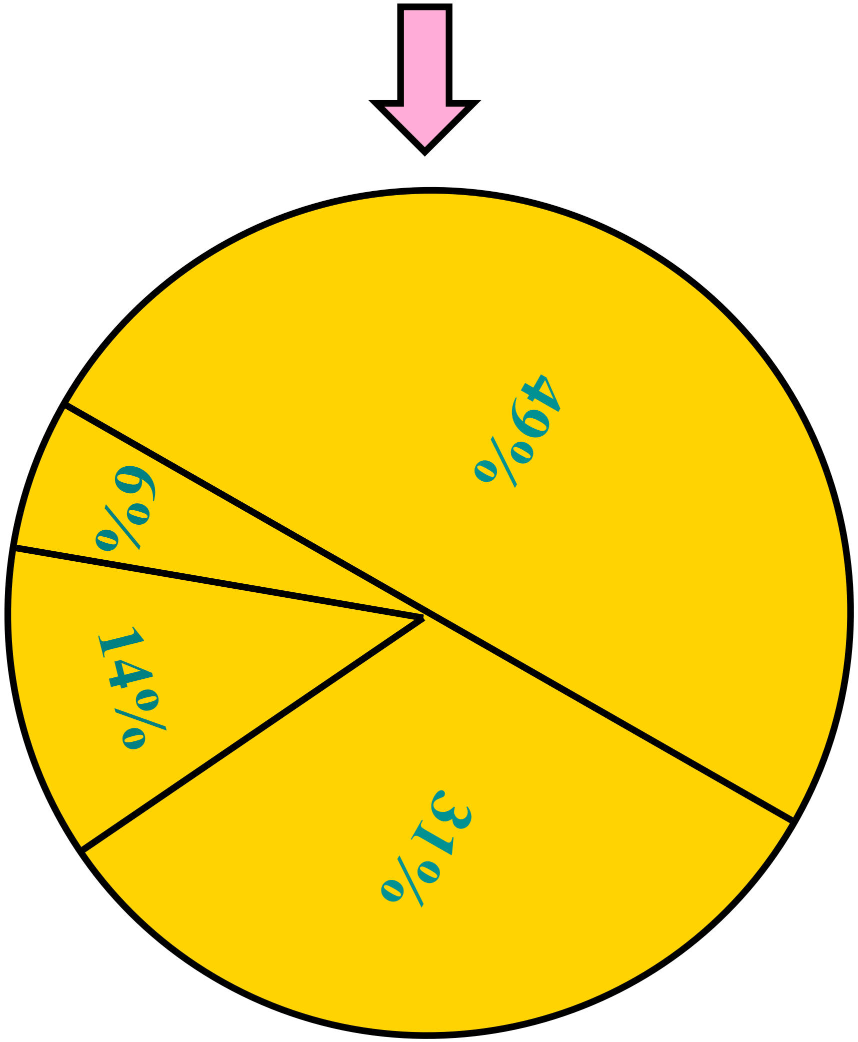
# Russian roulette



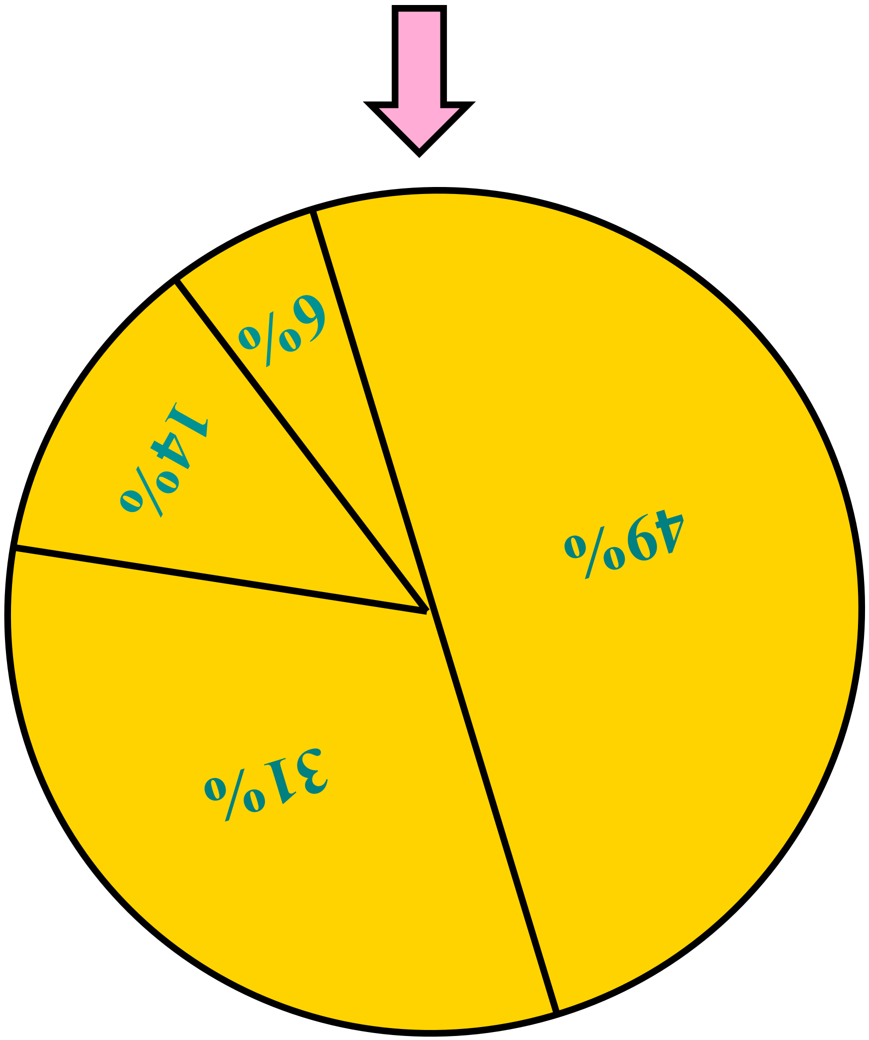

# Russian roulette



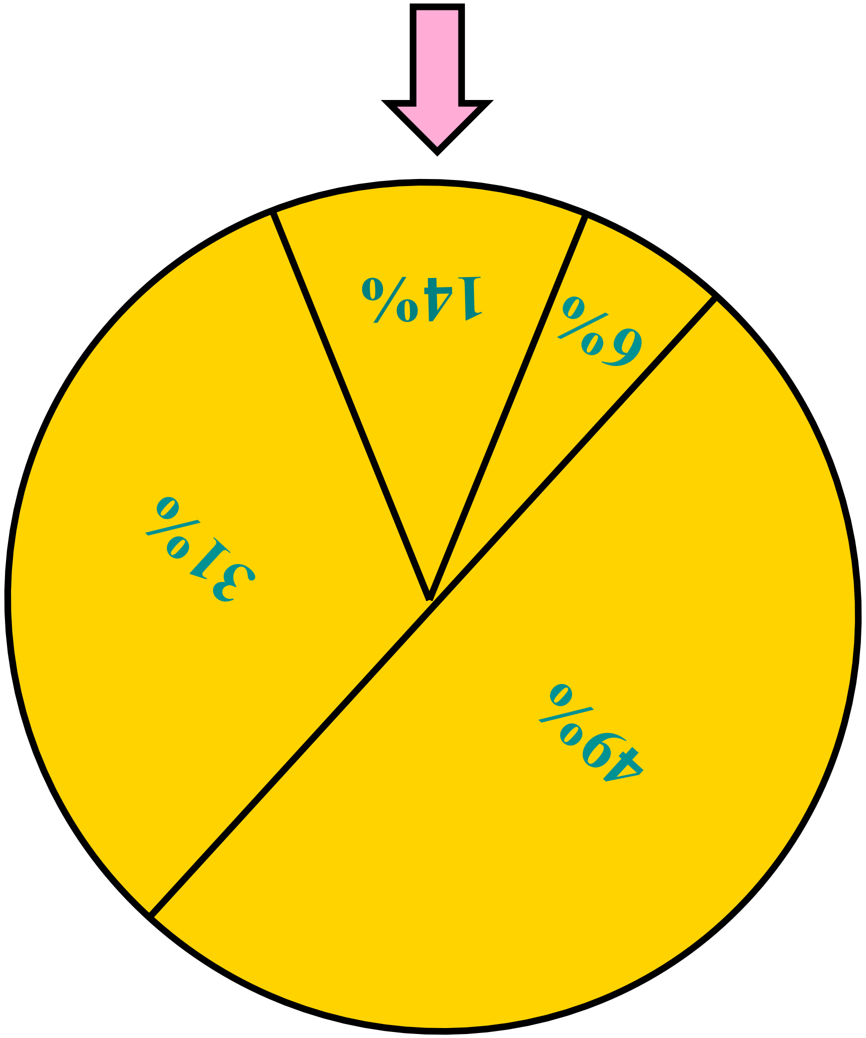

# Russian roulette




# Russian roulette




# Russian roulette




# Example

**Maximum** $f(x) = x^2$, $x \in [1, 31]$

- **Selection**

| | | | | |
|---|---|---|---|---|
| Individual | 01101, | 11000, | 01000, | 10011 |
| Fitness | 169, | 576, | 64, | 361 = 1170 |
| Probability | 0.14, | 0.49, | 0.06, | 0.31 = 1.0 |
| Result | 01101, | 11000, | 11000, | 10011 |

- **Crossover**

0110 | 1 ⇨ 01100      11 | 000 ⇨ 11011

1100 | 0    11001      10 | 011    10000

- **Mutation**

<u>0</u>1100 ⇨ <u>1</u>1100

# Genetic algorithm

**begin**
  **set** time $t = 0$
  initialize the ***population*** $P(t)$
  **while** the termination condition is not met **do**
    **begin**
      evaluate fitness of each member of the population $P(t)$;
      select members from population $P(t)$ based on ***fitness***;
      produce the ***offspring*** of these pairs using ***genetic operators***;
      replace  candidates of $P(t)$, with these offspring;
      **set** time $t = t + 1$
    **end**
**end**

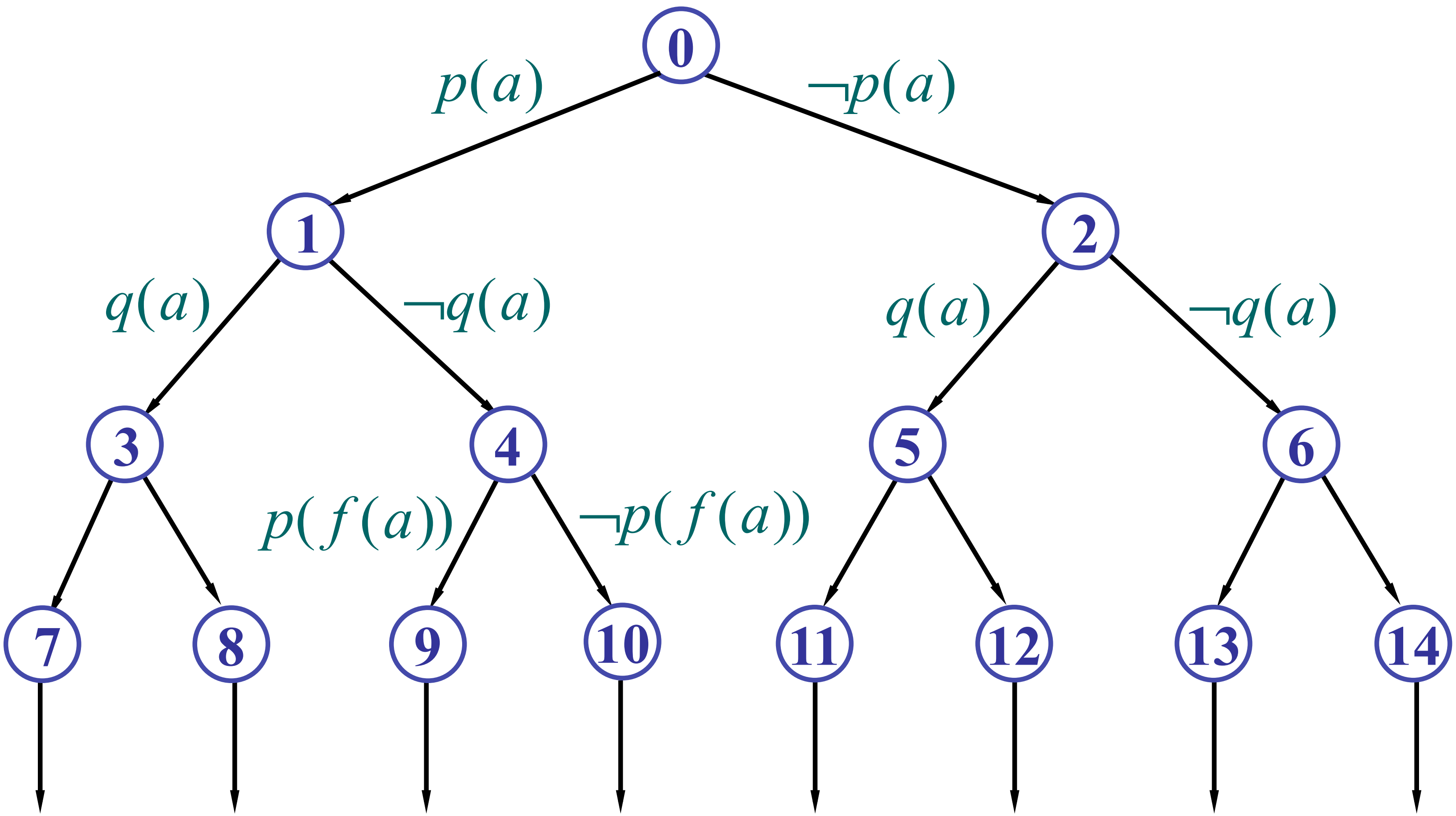
# Knapsack problem

A thief robbing a store finds $n$ items; the $i$th item is worth $v_i$ dollars and weights $w_i$ pounds, where $v_i$ and $w_i$ are integers. He wants to take as valuable a load as possible, but he can carry at most $W$ pounds in his knapsack for some integer $W$.

***Which items should he take?***

# Semantic tree

$\{p(x), \neg p(x) \vee q(x), \neg q(f(a))\}$



# CNF-satisfaction problem

The ***conjunctive normal form*** (CNF) satisfiability problem is straightforward: an expression of propositions is in conjunctive normal form when it is a sequence of clauses joined by an **AND** relation. Each of these clauses is in the form of a disjunction, the **OR** of literals.

$$(\neg a \vee c) \wedge (\neg a \vee c \vee \neg e) \wedge (\neg b \vee c \vee d \vee \neg e) \wedge$$
$$(a \vee \neg b \vee c) \wedge (\neg e \vee v)$$

# Analysis of traveling salesman problem

Given a finite number of "cities" along with the cost of travel between each pair of them, find the ***cheapest way*** of visiting each city exactly once and finishing at the city he starts from.

- 19 ***Cities***
- ***Possible routes*** $= 18! = 6.40237 \times 10^{15}$
- ***life time*** $= 80 \times 365 \times 24 \times 60 \times 60$
  $= 2.52288 \times 10^{9}$
- ***Computer speed*** = 10000 ***routes/second***

253.77 ***Generation!***

# Traveling salesman problem

**Crossover**

$p_1 = (\,1\;\;9\;\;2 \mid 4\;\;6\;\;5\;\;7 \mid 8\;\;3\,)$

$p_2 = (\,4\;\;5\;\;9 \mid 1\;\;8\;\;7\;\;6 \mid 2\;\;3\,)$

Start from the second cut point of one parent, the cities from the other parent are copied in the same order, omitting cities already present. When the end of the string is reached, continue on from the beginning. The sequence of cities from $p_2$ is :

2 3 4 5 9 1 8 7 6

$c_1 = (\,2\;\;3\;\;9 \mid 4\;\;6\;\;5\;\;7 \mid 1\;\;8\,)$

$c_2 = (\,3\;\;9\;\;2 \mid 1\;\;8\;\;7\;\;6 \mid 4\;\;5\,)$

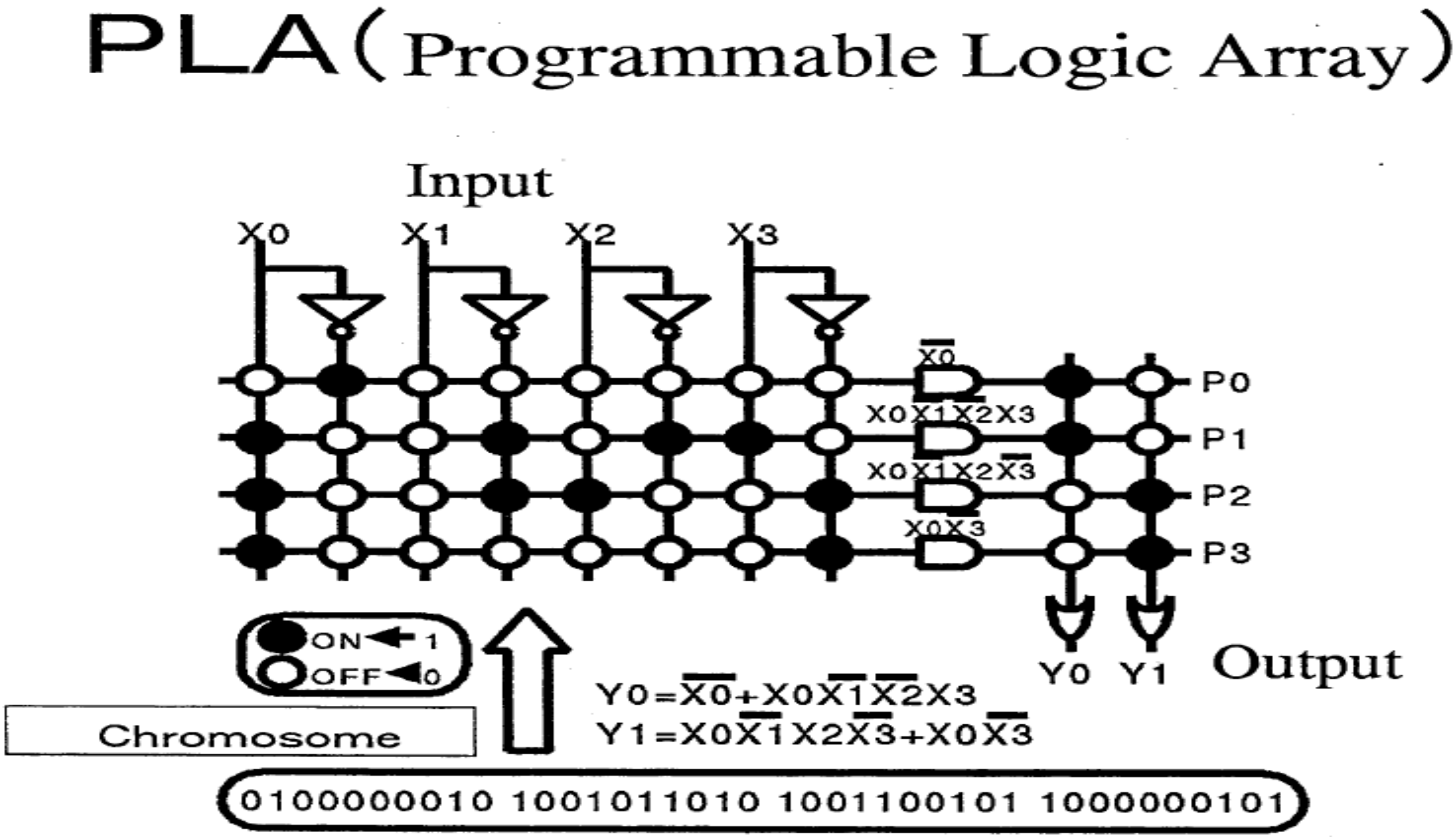
# Traveling salesman problem

**Mutation**

$$c_1 = (\,2\ \ 3\ \ 9\ |\ 4\ \ 6\ \ 5\ \ 7\ |\ 1\ \ 8\,)$$

$$c_1 = (\,2\ \ 3\ \ 9\ |\ \underbrace{7\ \ 5\ \ 6\ \ 4}_{\textit{Inversion}}\ |\ 1\ \ 8\,)$$

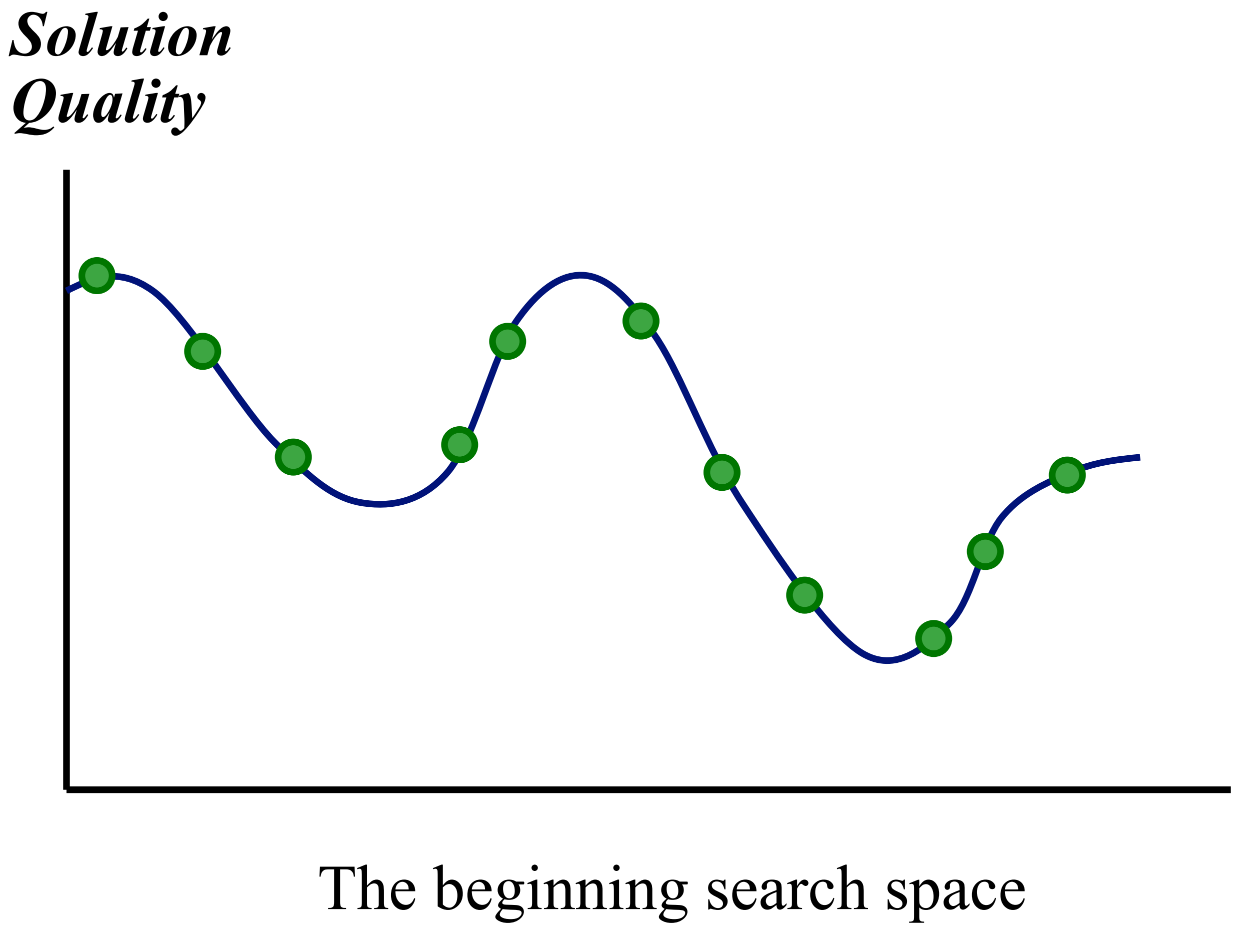
# Evolutionary hardware



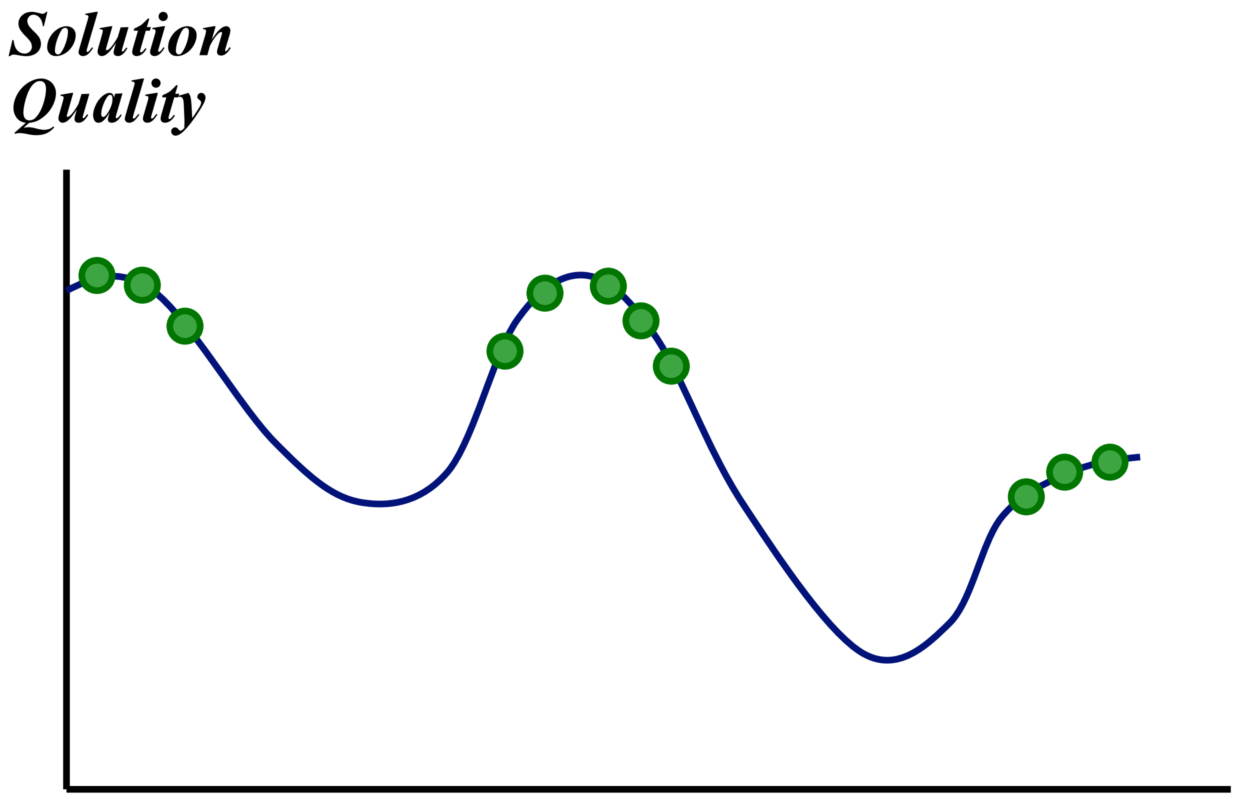
# Gray coding

| Binary | Gray | Binary | Gray |
|---|---|---|---|
| 0000 | 0000 | 1000 | 1100 |
| 0001 | 0001 | 1001 | 1101 |
| 0010 | 0011 | 1010 | 1111 |
| 0011 | 0010 | 1011 | 1110 |
| 0100 | 0110 | 1100 | 1010 |
| 0101 | 0111 | 1101 | 1011 |
| 0110 | 0101 | 1110 | 1001 |
| 0111 | 0100 | 1111 | 1000 |

# Strength of the genetic algorithms



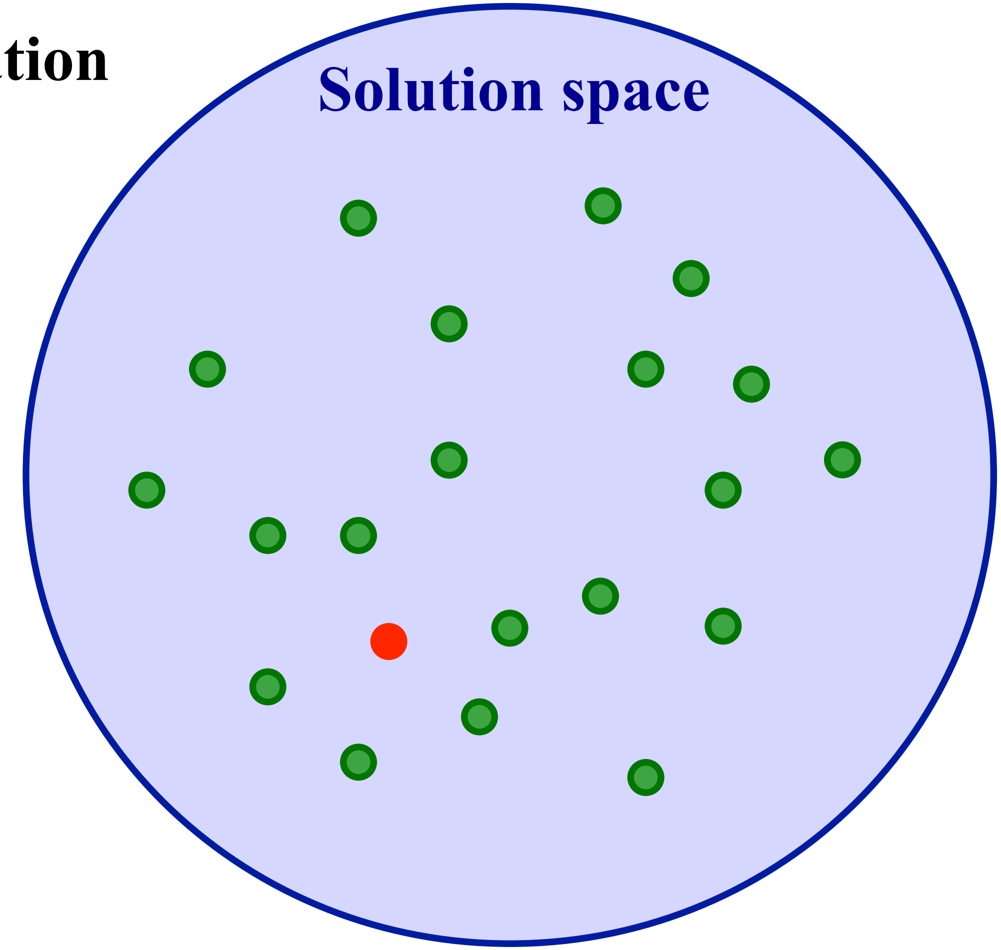
# Strength of the genetic algorithms




The search space after *n* generations

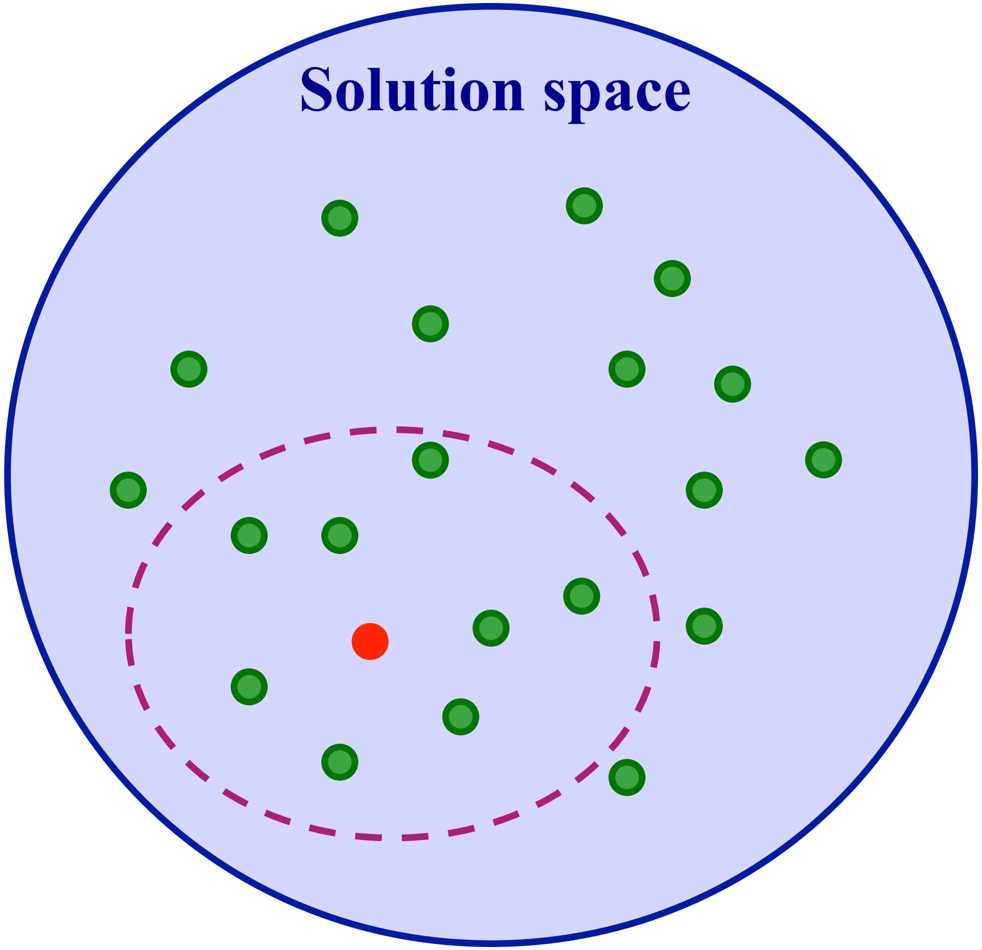
# Random research

**Initialization**



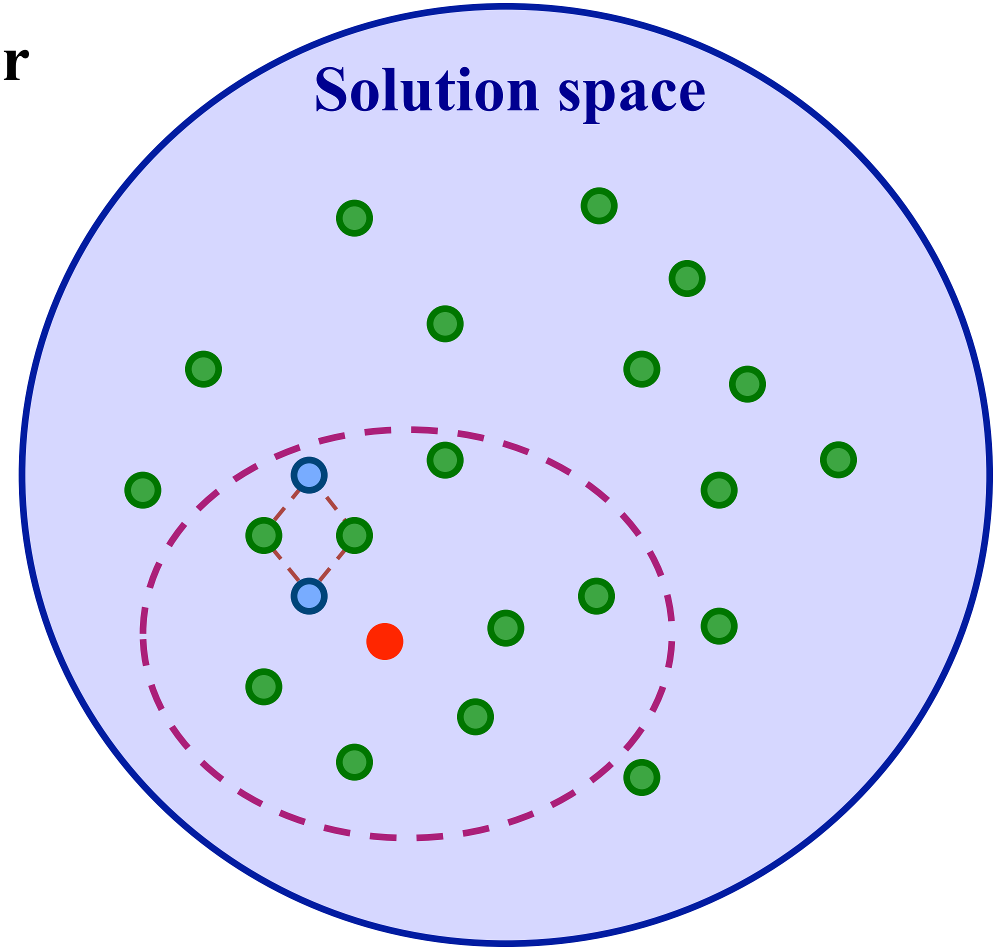
# Random research

**Selection**



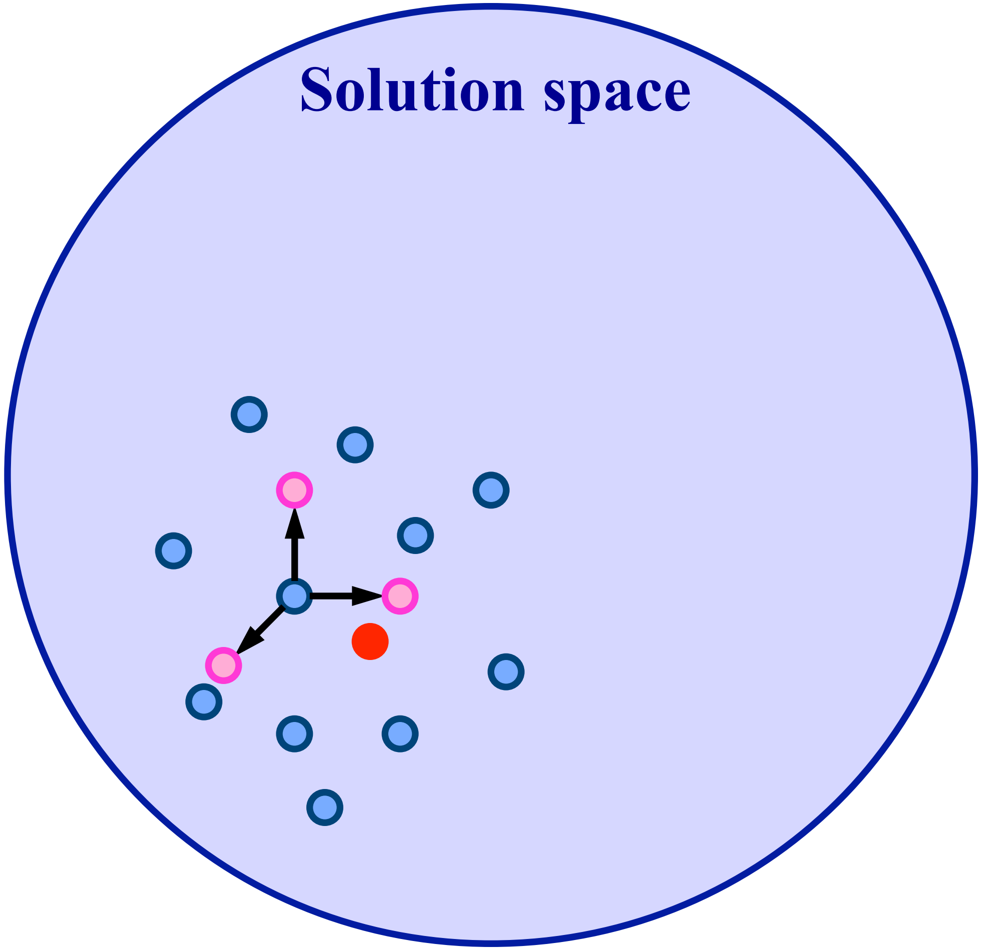
# Random research

**Crossover**



# Random research

**Mutation**



# Schema

- The schema ( 1 * * 0 * 1 ) describes the set of all strings of length 6 with 1's at positions 1 and 6 and a 0 at position 4. The * is a ***wildcard*** symbol, which means that positions 2, 3 and 5 can have a value of either 1 or 0.
- The ***order*** of a schema, $O(s)$, is defined as the number of ***fixed positions*** in the template, while the defining ***length*** $\delta(s)$ is the distance between the first and last specific positions. The order of ( 1 * * 0 * 1 ) is 3 and its defining length is 5.
- The ***fitness*** of a schema is the ***average*** fitness of all strings matching the schema. The fitness of a string is a measure of the value of the encoded problem solution, as computed by a problem-specific evaluation function.

# Schema theorem

The probability of the string $i$ that is selected:

$$p_i = \frac{f(i)}{\sum_{j=1}^{n} f(j)}$$

$m(s,t)$: the number of strings belonging to schema $s$ at generation $t$.

$$E\left[m(s,t+1)\right] = m(s,t) \cdot n \cdot \frac{\overline{f}(s)}{\sum_{j=1}^{n} f(j)}$$

# Schema theorem

$\bar{f}$ is the average fitness at generation $t$.

$$\bar{f} = \frac{\sum_{j=1}^{n} f(j)}{n}$$

Then,

$$E\left[m(s,t+1)\right] = m(s,t) \cdot \frac{\bar{f}(s)}{\bar{f}}$$

Let $\bar{f}(s) = (1+c)\bar{f}$

$$E\left[m(s,t+1)\right] = m(s,t) \cdot \frac{(1+c)\bar{f}}{\bar{f}}$$

$$= m(s,t) \cdot (1+c) = m(s,0) \cdot (1+c)^{t+1}$$

# Schema theorem

The probability of disruption $p_s$ is the probability that crossover will not destroy the schema $s$.

$$p_s \geq 1 - \frac{\delta(s)}{l-1}$$

$p_c$ is the probability of crossover.

$$p_s \geq 1 - p_c \cdot \frac{\delta(s)}{l-1}$$

Then,

$$E\left[m(s,t+1)\right] = m(s,t) \cdot \frac{\overline{f}(s)}{\overline{f}}\left[1 - p_c \cdot \frac{\delta(s)}{l-1}\right]$$

# Schema theorem

The probability of disruption $p_s$ is the probability that mutation will not destroy the schema $s$, and $p_m$ is the probability of mutation.

$$p_s = \left(1 - p_m\right)^{O(s)}$$

Assuming $p_m << 1$

$$p_s \approx 1 - p_m \cdot O(s)$$

Then,

$$E\left[m(s,t+1)\right] = m(s,t) \cdot \frac{\overline{f}(s)}{\overline{f}} \cdot \left[1 - p_c \cdot \frac{\delta(s)}{l-1}\right] \cdot \left[1 - O(s) p_m\right]$$

$$\approx m(s,t) \cdot \frac{\overline{f}(s)}{\overline{f}} \cdot \left[1 - p_c \cdot \frac{\delta(s)}{l-1} - O(s) \cdot p_m\right]$$

# Schema theorem

- Holland's schema theorem is widely taken to be the foundation for explanations of the power of ***genetic algorithms***.
- A ***schema*** is a template that identifies a ***subset*** of strings with similarities at certain string positions.
- The schema theorem states that ***short***, ***low-order***, ***schemata*** with ***above-average fitness*** increase exponentially in successive generations.

# Any question?

Xiaoqing Zheng
Fundan University