Genetic & Evolutionary Algorithms:
Implementation, Investigation on Selection

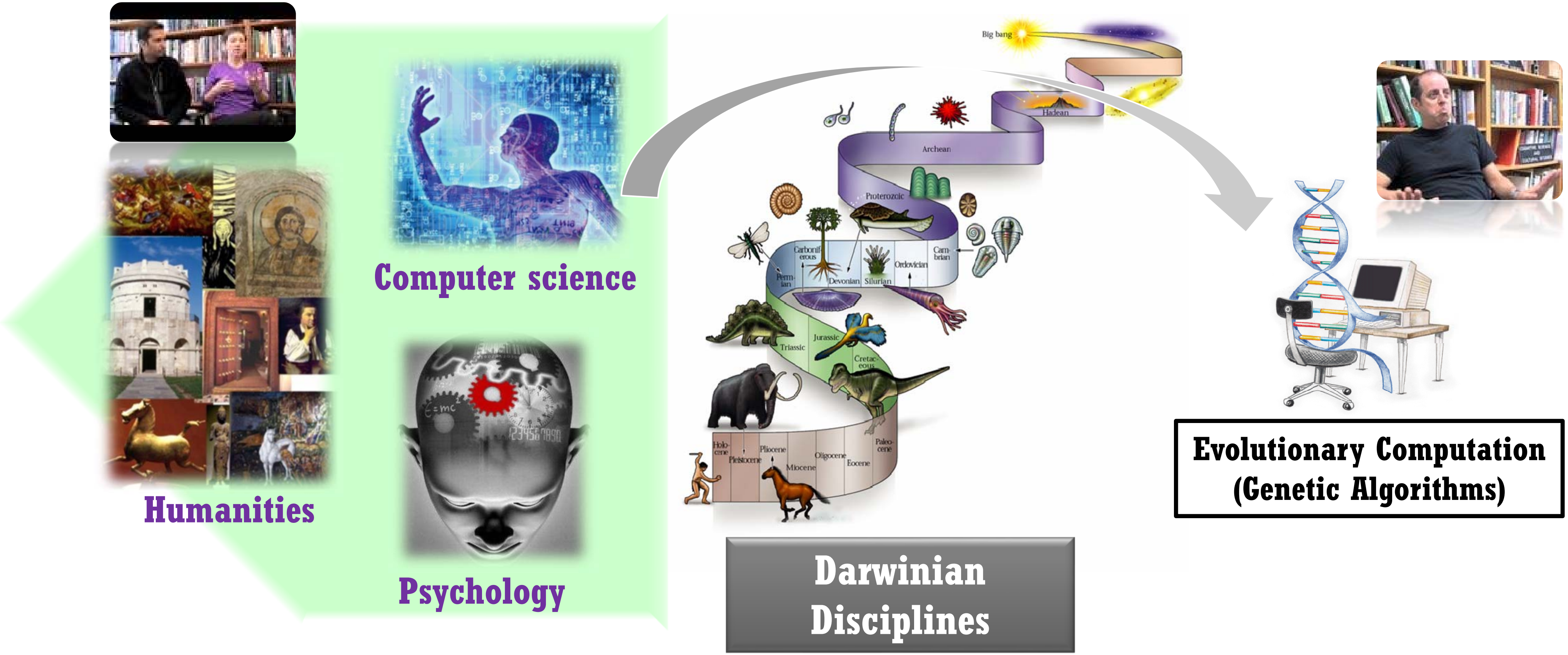
September 18, 2019

Prof. Chang Wook Ahn

Meta-Evolutionary Machine Intelligence (MEMI) Lab.
Electrical Eng. & Computer Sci.
Gwangju Inst. Sci. & Tech. (GIST)

GAs - Review (1)

- "Evolution" is still evolving in places.
  - Biological Evolution gives the inspiration to do new research
    - ✓ Psychology, The Humanities, Computer Science, etc.
  - GAs are an outcome of the Darwinian + the computing algorithm



GAs - Review (2)

❖ What's the Target of Interest?

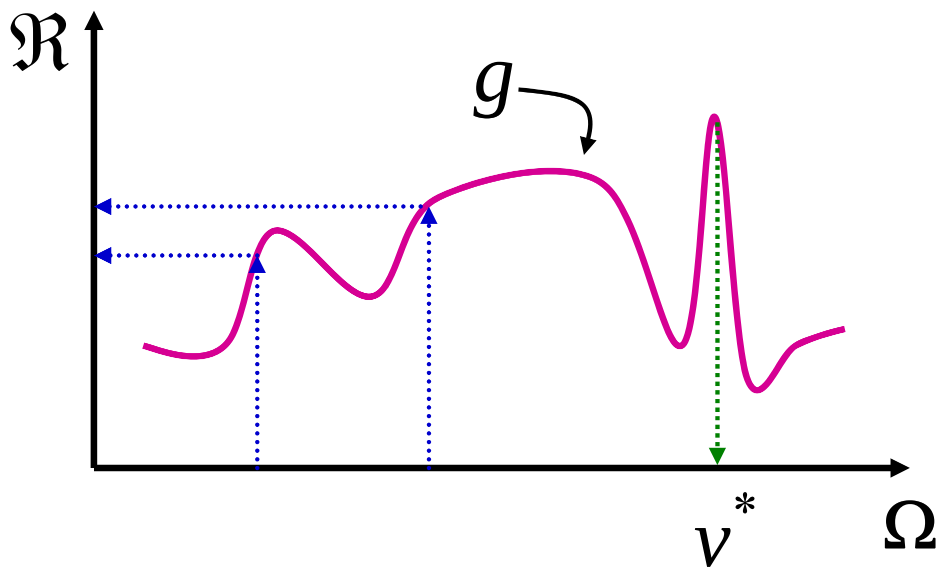
- **Optimization Problems**
  - ✓ Can be defined by specifying the set of all feasible candidates
  - ✓ The goal is to find the best solution(s)

Formal Definition

For a search space Ω

There is a function $g : \Omega \mapsto \Re$

The task is to find $v^* = \arg\max_{v \in \Omega} g$

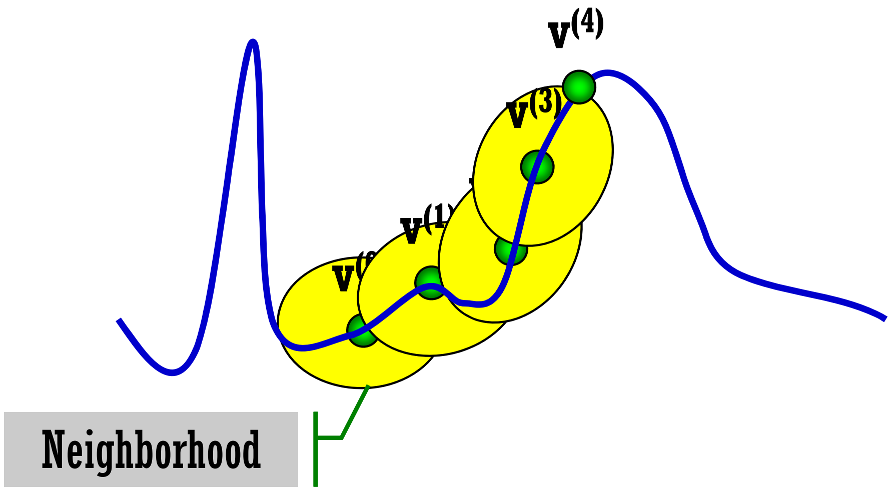


Here, v is a vector of decision variables, and g is the objective function

Conventional Stochastic Approach

❖ Neighborhood Search

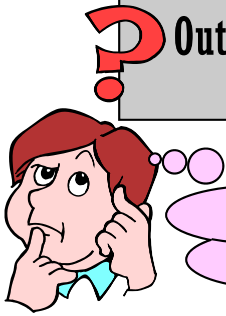
- Also, called '**Hill-climbing**'
- Widely used in various **COPs**
- Simple procedures as follows:
  1. All neighbors are evaluated
  2. The best one is selected
  3. Iterate until no more improvement



(* Pseudo-code of NS *)

Generate an initial solution v;
Specify a neighborhood function N(v);
Store v^* as current best v and evaluate $g^*=f(v)$;
WHILE termination condition are not satisfied
  select a solution $v' \in N(v)$;
  evaluate $g' = f(v')$;
  IF $g' > g^*$ then
    store v' as current best v^* and g' as g^*
    // $v^* := v'$; $g^* := g'$;
  END
END
Output v^* and g^*

It is prone to be converged into the sub-optimum.
It cannot escape from the sub-optimum.

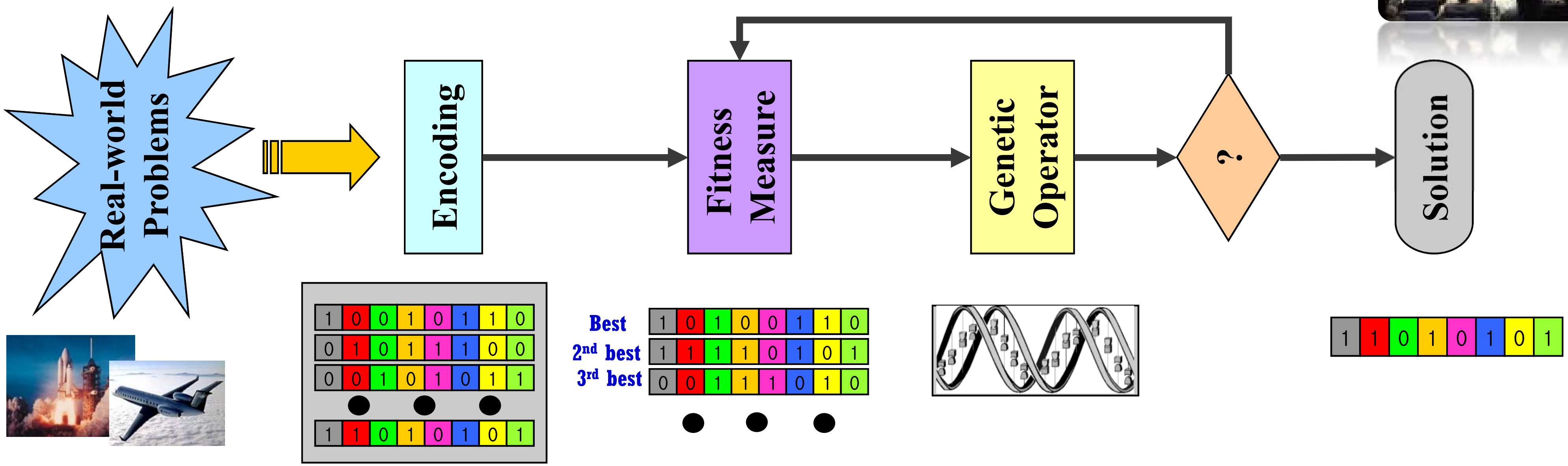


What if GA and NS are compared in a fair manner?

GAs - Review (3)

❖ Key Components & Terminology

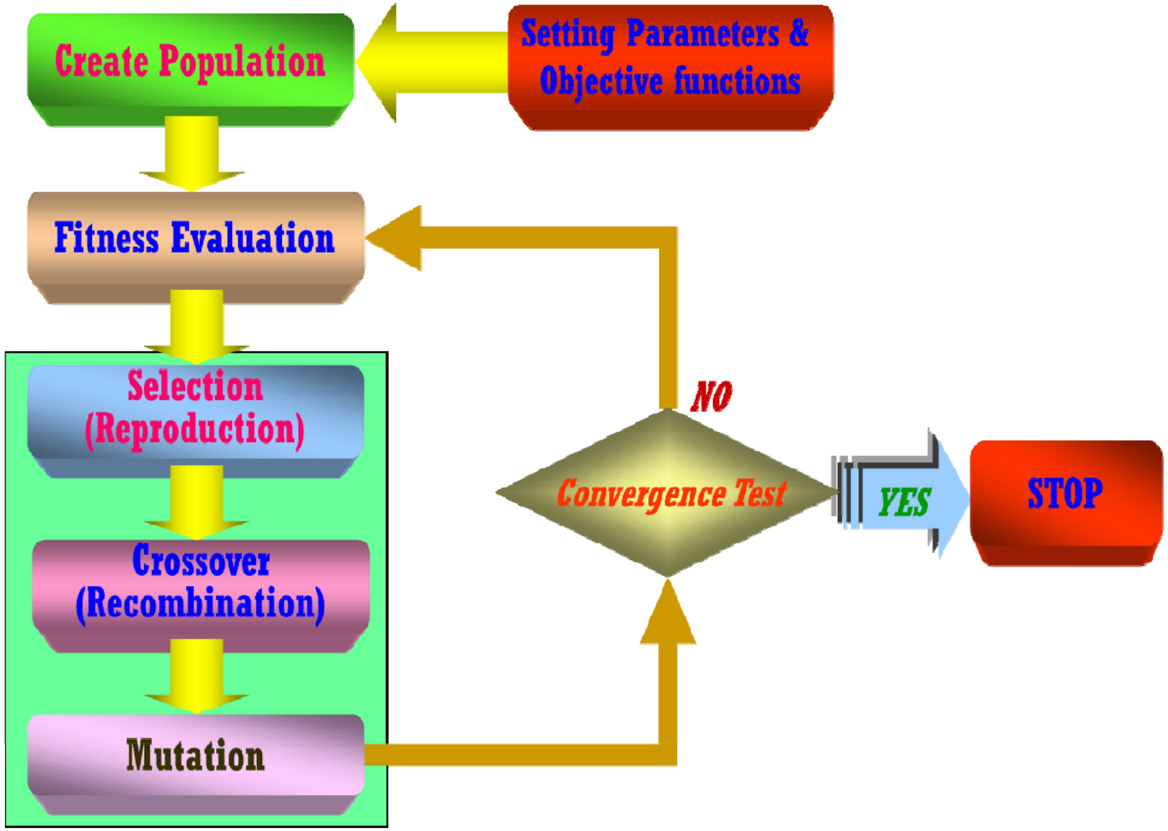
- **Encoding:** variables (phenotype) are encoded into a chromosome (genotype)
- **Population:** a set of chromosomes (i.e., individuals or candidate solutions)
- **Fitness function:** measure the goodness of each candidate solution: it can be mathematical terms, computer simulation, human evaluation
- **Genetic operators:** boosting chromosomes up towards the optimum
  - ✓ **Selection:** realize the survival of the fittest
  - ✓ **Crossover:** realize the genetic inheritance
  - ✓ **Mutation:** realize the genetic mutation



GAs - Pseudocode (1)

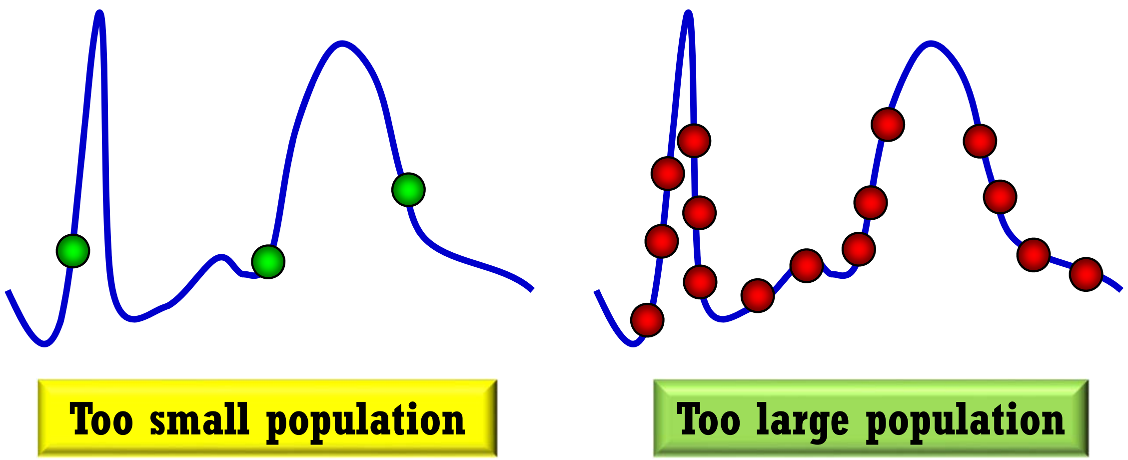
❖ Possible Implementation of GAs

```
(* Overall Procedures of GAs *)
t := 0;
/*Create an N individuals as a population*/
P(t) = initialize( N );
fitness = evaluation( P(t) );
WHILE stopping condition not fulfilled DO
    t = t+1;
    tmp_P(t) = selection ( fitness, P(t) );
    tmp_P(t) = crossover ( tmp_P(t) );
    P(t) = mutation( tmp_P(t) );
    fitness = evaluation( P(t) );
END
Solution: the best individual
```



GAs - Pseudocode (2)

```
(* Initialization *)
FOR i := 1 TO N        //as to population size
   FOR i := 1 TO L   //as to individual length
      IF ( random[0,1] > 0.5 )
         P_ij^(t) := 1;
      ELSE
         P_ij^(t) := 0;
END  END  END
```



❖ How Many Individuals (i.e., population size)?

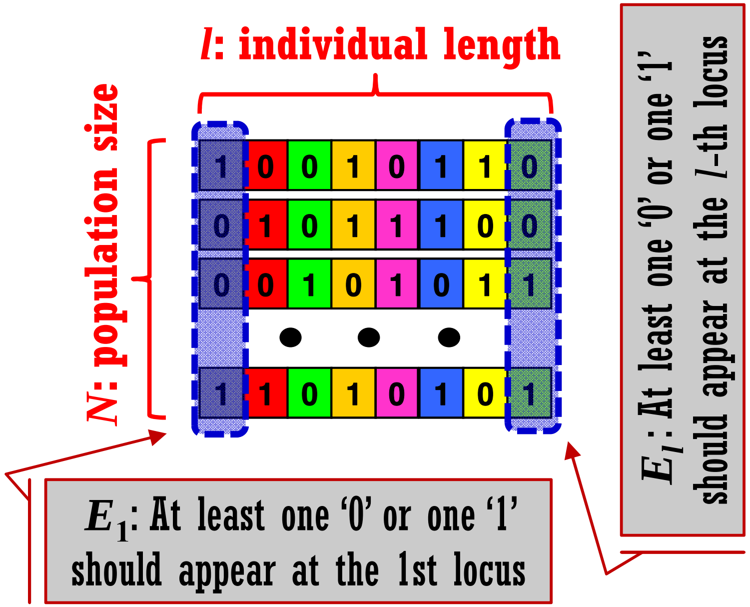
- ➢ Intuitively, there should be some *optimal value* on the grounds that
  - ✓ Too small population is not sufficient for exploring the effective search
  - ✓ Too large population impairs the computational efficiency
- ➢ It is plausible to grow the population size with the *string length* (i.e., problem size)
  - ✓ But even a linear growth rate would lead to a quite large population in some cases

♣ This issue would be further investigated later in detail.

GAs – Initial Population

❖ Proper Initial Population Size?

- A **minimum population size** for a meaningful search to take place is required
  - ✓ At the very least, every point in the search space should be reachable from the initial population by crossover only
  - ✓ That is, at least one instance of every allele should be at each locus in the population



$$P[E_1] = 1 - \{P[\text{all } x_1{=}0] + P[\text{all } x_1{=}1]\}$$

$$P[\text{all } x_1{=}0] = (1/2)^N = P[\text{all } x_1{=}1]\}$$

$$P[E_1] = \{1 - (1/2)^{N-1}\} = P[E_2] = \ldots = P[E_l]$$

$$P[E] = P[E_1] \cdot P[E_2] \ldots P[E_l] = \{1 - (1/2)^{N-1}\}^l$$

$$P[E] \approx \exp(-l / 2^{N-1})$$

$$N \approx \mathrm{ceil}(1 + \log_2(-l / \ln \boxed{P[E]}))$$

Confidence

Let E be an event that at least one allele is present at every locus

e.g.) A population size 17 is enough to ensure that the confidence exceeds 99.9% for individuals of length 50!

GAs - Pseudocode (3)

(* Roulette Wheel Selection *)

FOR i := 1 TO N

  x := random[0,1];

  k := 1;

  WHILE k < N && $x > \sum_{j=1}^{k} f\left(P_j^{(t)}\right) \Big/ \sum_{l=1}^{N} f\left(P_l^{(t)}\right)$

    k := k+1;

  tmp_$P_i^{(t)}$:= $P_k^{(t)}$

END

(* Tournament Selection *)

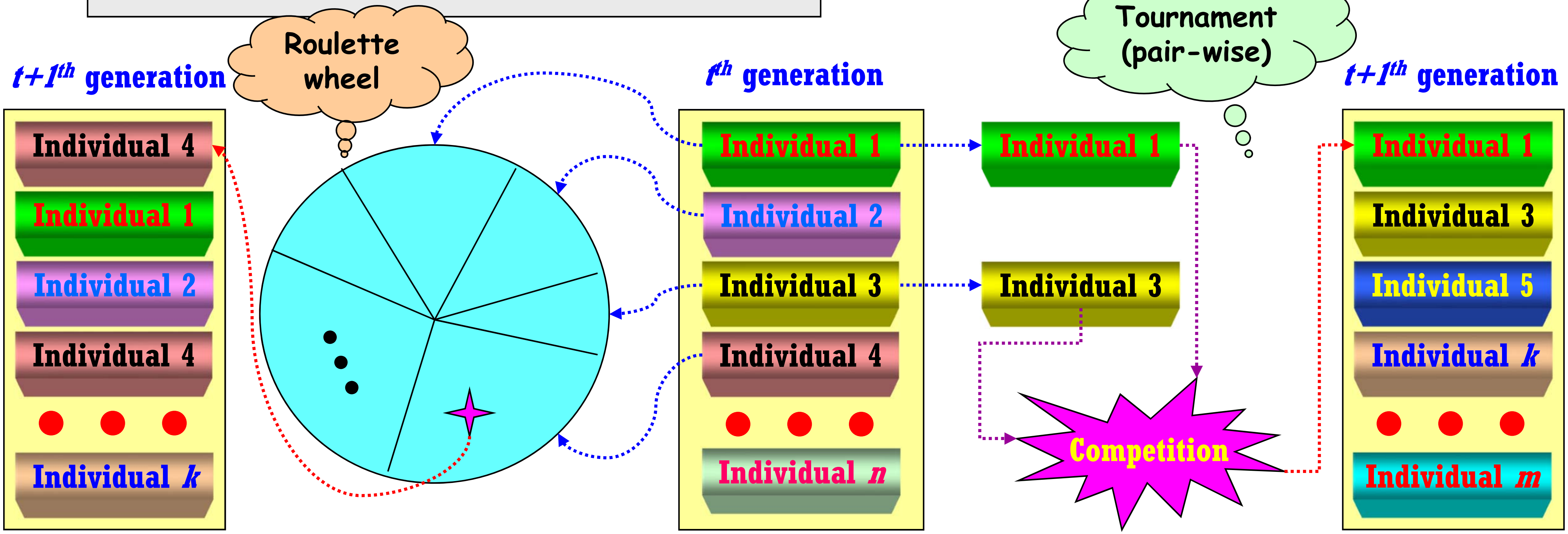
FOR i := 1 TO N

  x := random_int[1, N];

  IF $f(P_i^{(t)}) < f(P_x^{(t)})$

    tmp_$P_i^{(t)}$:= $P_x^{(t)}$;

  ELSE tmp_$P_i^{(t)}$:= $P_i^{(t)}$;



GAs - Pseudocode (4)

At first, Shuffling needs!

Selected population

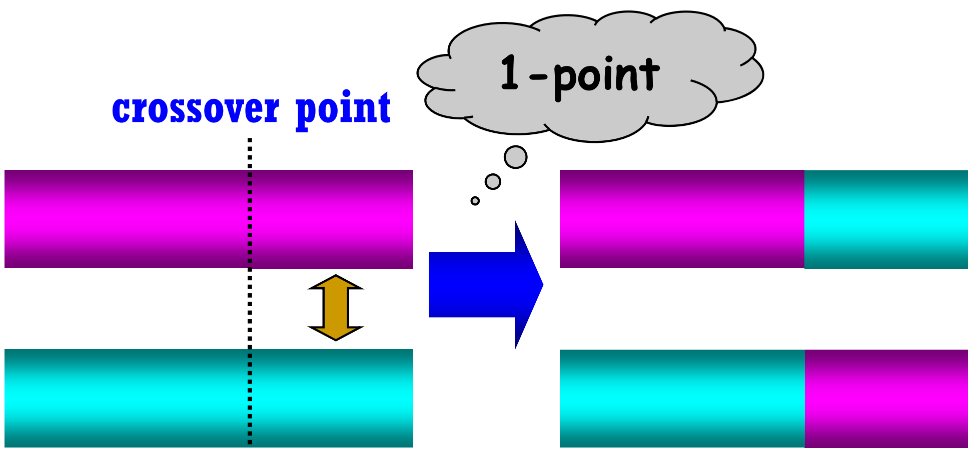
1	0	0	1	0	1	1	0
0	1	0	1	1	1	0	0
0	0	1	0	1	0	1	1
1	1	0	1	0	1	0	1
0	1	0	1	1	1	0	0
1	1	0	1	0	1	0	1

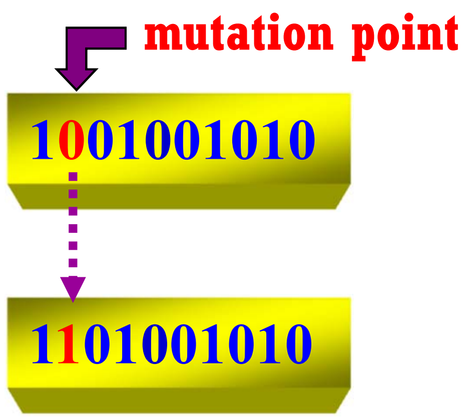
```
(* One-point Crossover *)
FOR i := 1 TO N/2
  IF random[0,1] ≤ P_C
    pos := random_int[1, n-1];
    FOR k := pos+1 TO n
      aux := tmp_P_i^(t)[k];
      tmp_P_i^(t)[k] = tmp_P_{i+N/2}^(t)[k];
      tmp_P_{i+N/2}^(t)[k] = aux;
  END   END
END
```

Crossover probability

```
(* Bit-wise Mutation *)
FOR i := 1 TO N
  FOR k := 1 TO n
    IF random[0,1] < P_M
      invert( tmp_P_i^(t)[k] );
    END
  END
END
P^(t) := tmp_P^(t);
```

Mutation probability

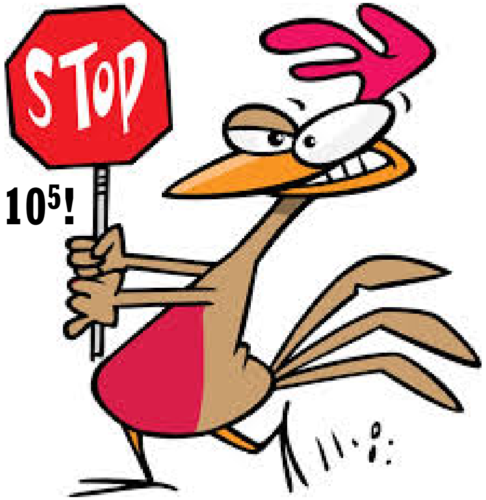




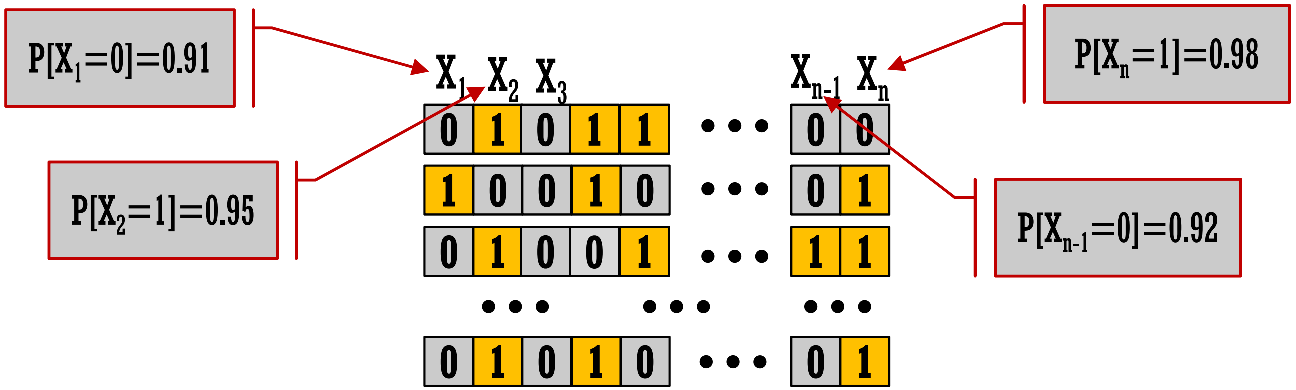
GAs - Termination

❖ When Do GAs Terminate?

- GAs are a stochastic search method that could in principle run forever.
- In practice, a **termination condition** is required
  1. **Set a limit on the number of fitness evaluations**
     - ✓ e.g., Stop the run when the number of evaluations exceeds 10^5!
  2. **Set the computer clock time**
     - ✓ e.g., Stop the run when the CPU time exceeds 1.5 seconds!
  3. **Track the population diversity/convergence**
     - ✓ e.g., Stop if at every locus the portion of one particular allele rises above 90%.

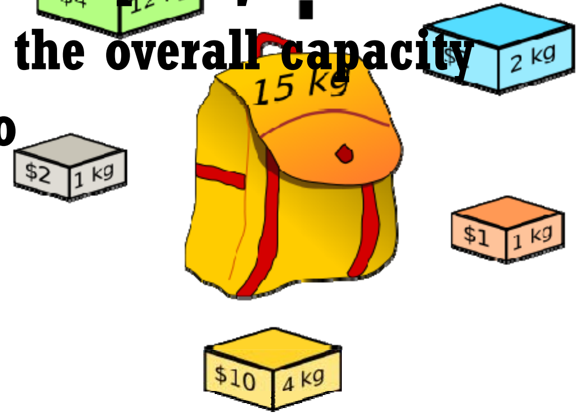


♣ The choice depends on its own purpose!



Example: 0-1 Knapsack Problem

- A set of ***n* items** is available to packed into a knapsack with **capacity *C* units**.
- Item i has a **value w_i** (e.g., \$) and uses up **c_i units** (e.g., kg) of capacity
- The aim is to maximize the amount of values while keeping the overall capacity
- That is, determining the **subset *I*** of items to pack in order to



$$\max \sum_{i \in I} w_i \quad \text{subject to} \sum_{i \in I} c_i \le C$$

Problem formulation!

\- If we define

$$x_i = \begin{cases} 1, & \text{if item } i \text{ is packed} \\ 0, & \text{otherwise} \end{cases}$$

How to handle it?

\- The knapsack problem is given as

$$\max \sum_{i=1}^{n} w_i x_i \quad \text{subject to} \sum_{i=1}^{n} c_i x_i \le C$$