Genetic & Evolutionary Algorithms:
More Investigation: Selection

March 19, 2013

Prof. Chang Wook Ahn

Sungkyunkwan Evolutionary Algorithms Lab.
School of Info. & Comm. Eng.
Sungkyunkwan University

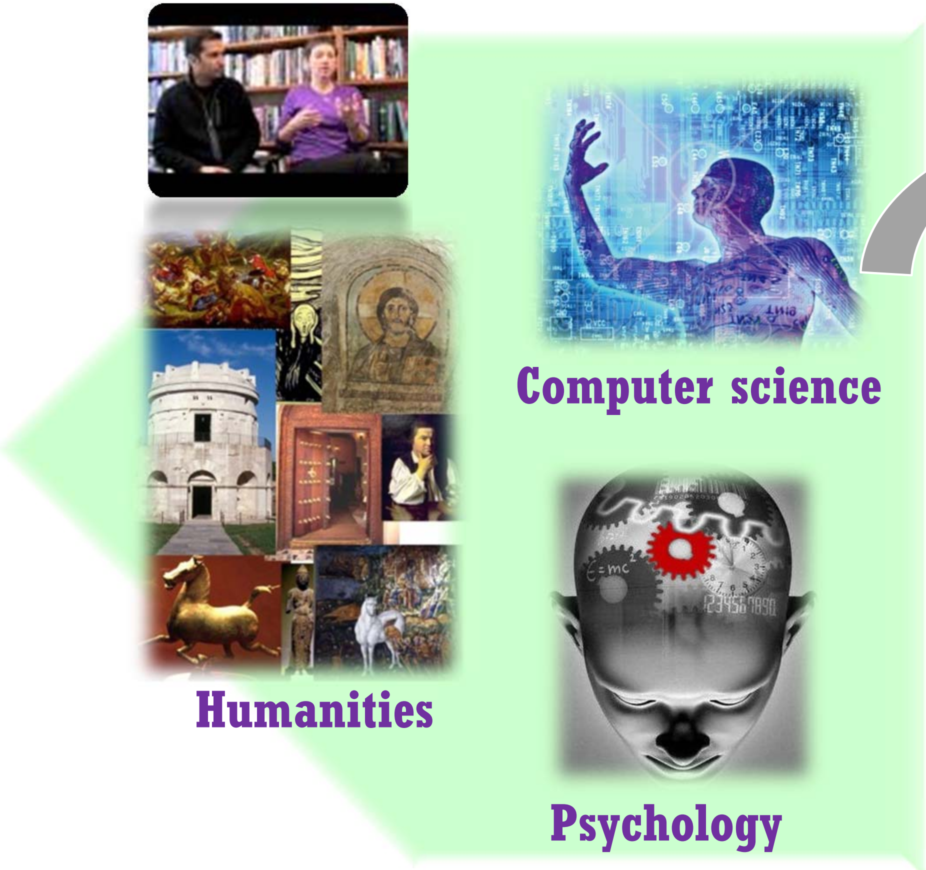
Contents

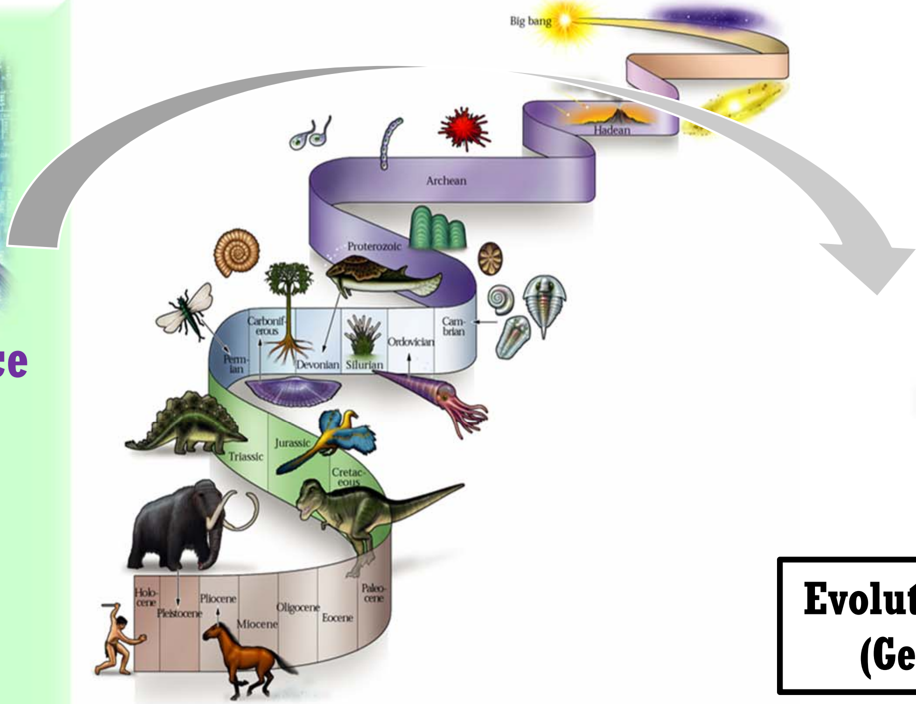
- GAs – Review
- GAs – Pseudo-code
- Conventional Approach
  - Neighborhood Search (NS)
- Application
  - 0-1 Knapsack Problem
  - Resource Allocation
- More Investigation: Selection
  - Proportional Selection
    - ✓Selection Noise, Scaling
  - Ordinal Selection
    - ✓ Rank selection, Tournament selection

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**





Darwinian Disciplines

Evolutionary Computation (Genetic Algorithms)

GAs - Review (2)

❖ What's the Target of Interest?

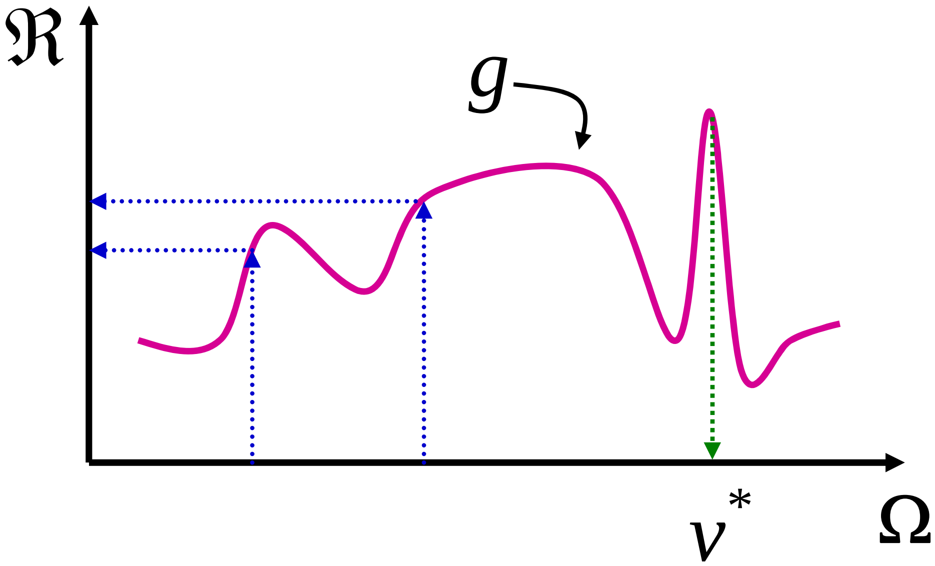
- **Optimization Problems**
  - ✓ Can be defined by specifying the set of all feasible candidates
  - ✓ The goal is to find the best solution(s)

Formal Definition

For a search space Ω

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

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

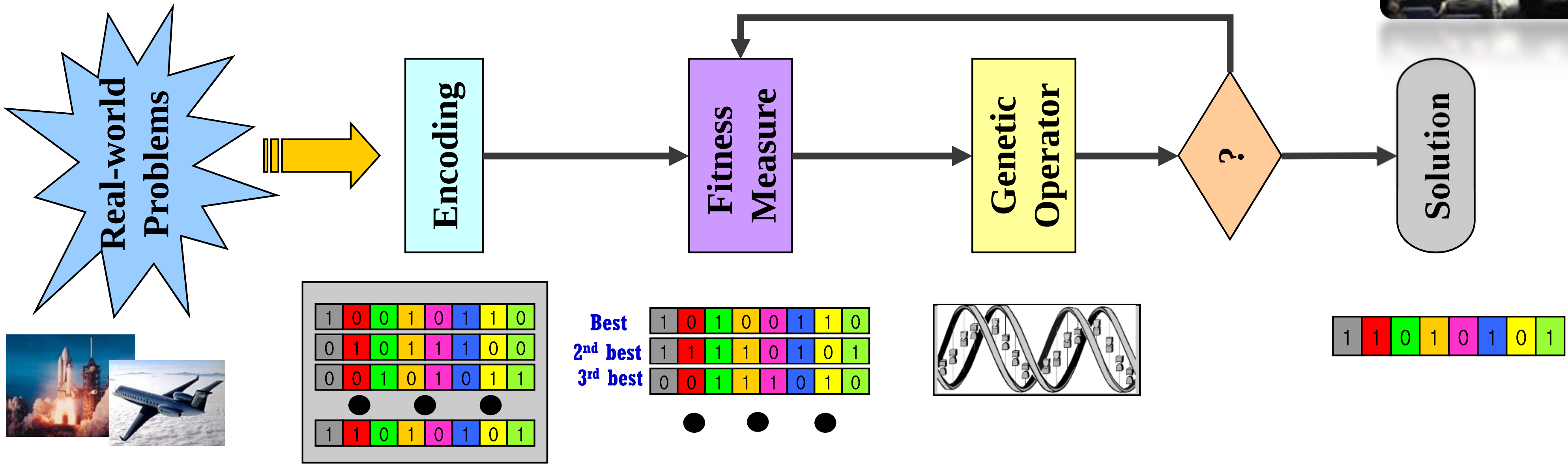


Here, v is a vector of decision variables, and g is the objective function

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

❖ Possible Implementation

```
t := 0;
Create initial population P(0) = (P_1(0),...,P_N(0))
WHILE stopping condition not fulfilled
```

Shuffling needs!

```
(* proportional selection - RWS *)
  FOR i := 1 TO N
    x := random[0,1];
    k := 1;
    WHILE k < N && x > Σ_{j=1}^{k} f(P_j(t)) / Σ_{l=1}^{N} f(P_l(t))
      k := k+1;
    tmp_P_i(t) := P_k(t)
  END
```

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

```
(* 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);
```

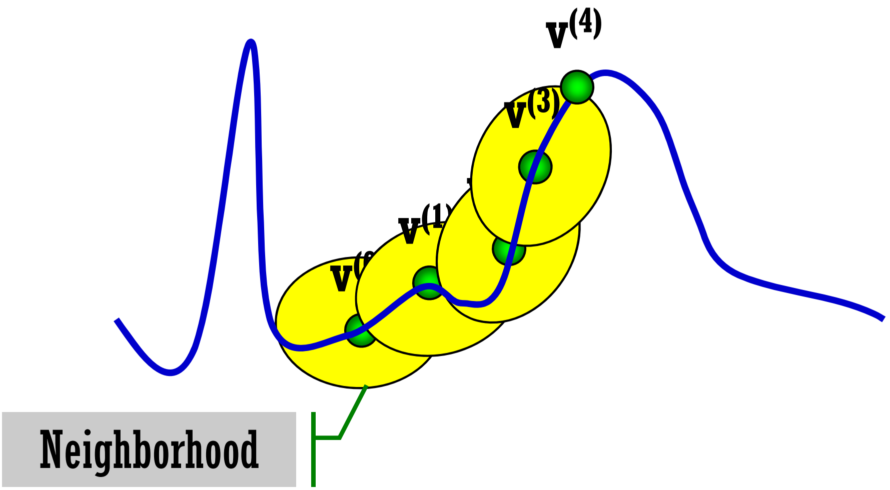
```
(* one-point crossover *)
  FOR i := 1 TO N/2
    IF random[0,1] ≤ P_C
      pos := random_int[1, n-1];
      FOR k := pos+1 TO n
        aux := tmp_P_i(t)[k];
        tmp_P_i(t)[k] = tmp_P_{i+N/2}(t)[k];
        tmp_P_{i+N/2}(t)[k] = aux;
  END   END   END
```

```
(* 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 := tmp_P;
  t := t+1;
END
```

Conventional 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^*

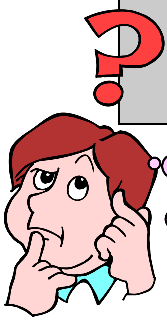
    // $v^* := v'$; $g^* := g'$;

  END

END

Output v^* and g^*

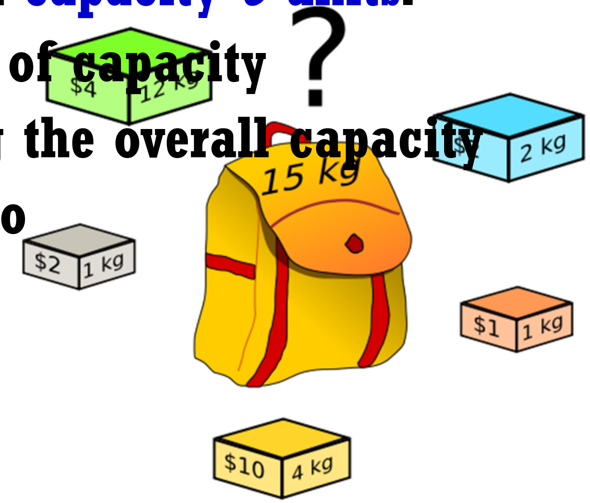
It is prone to be converged into the sub-optimum.
It cannot escape from the sub-optimum.



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

Application: 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



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

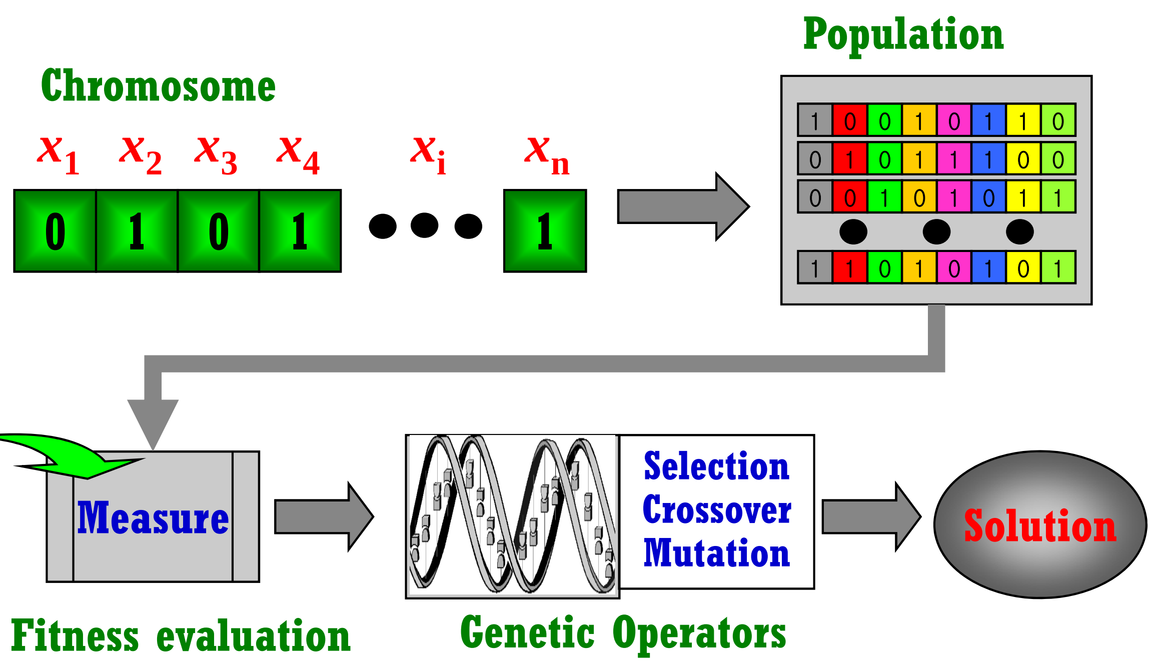
Problem formulation!

- If we define

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

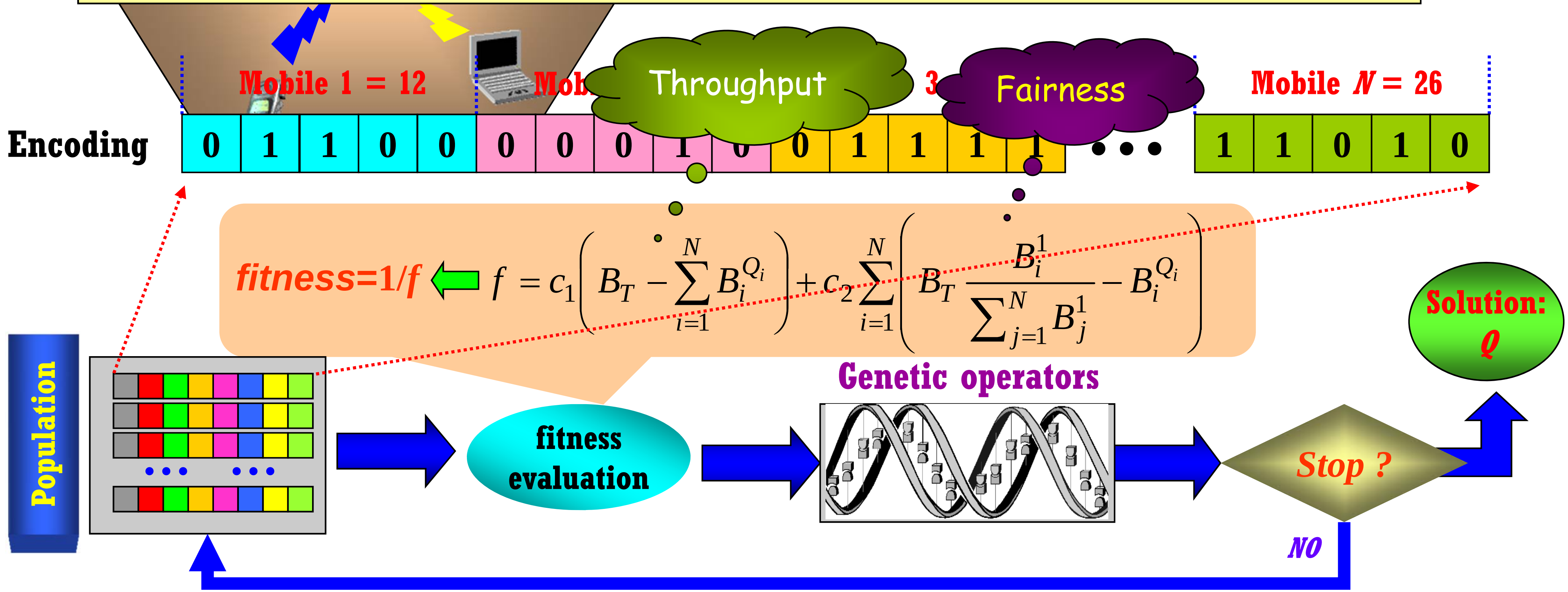
- The knapsack problem is given as

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



Application: Resource Allocation

QoS Index	1	2	…	14	15	…	26	27
Video	High	High	…	Mid.	Mid.	…	Low	Low
Audio	High	High	…	Mid.	Mid.	…	Low	Low
Data	High	Mid.	…	Mid.	Low	…	Mid.	Low

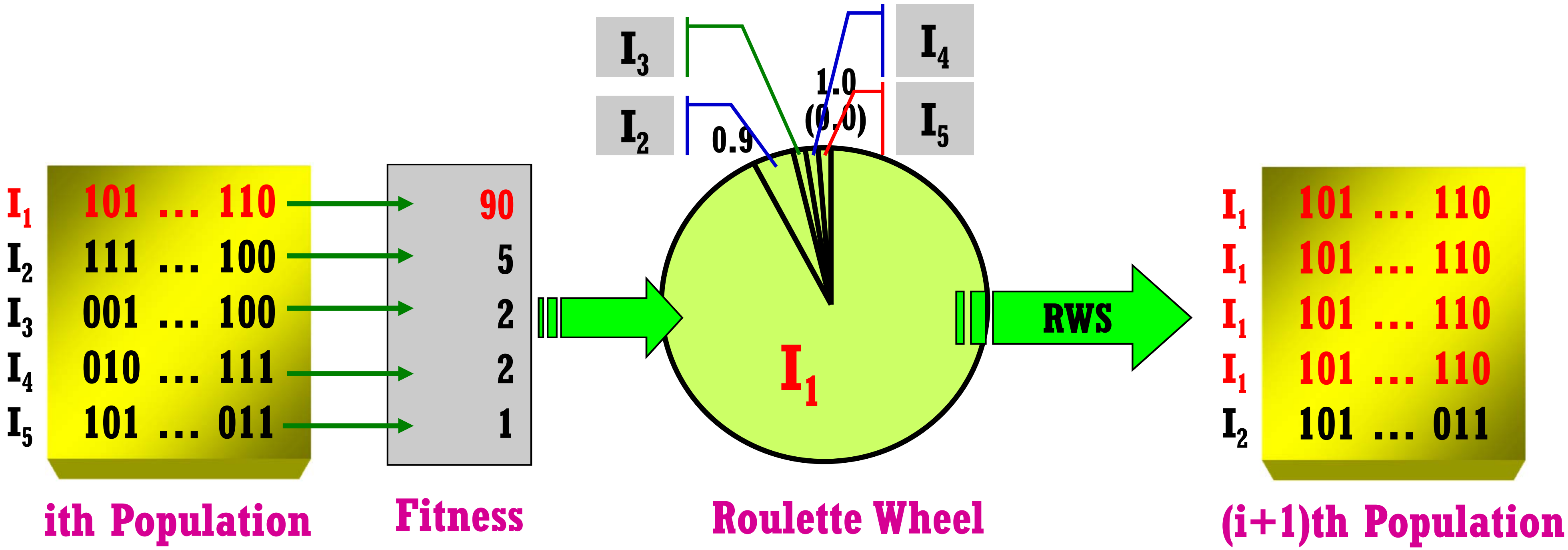


Further Detailed Investigation on Selection

Proportional Selection (1)

Selection Noise

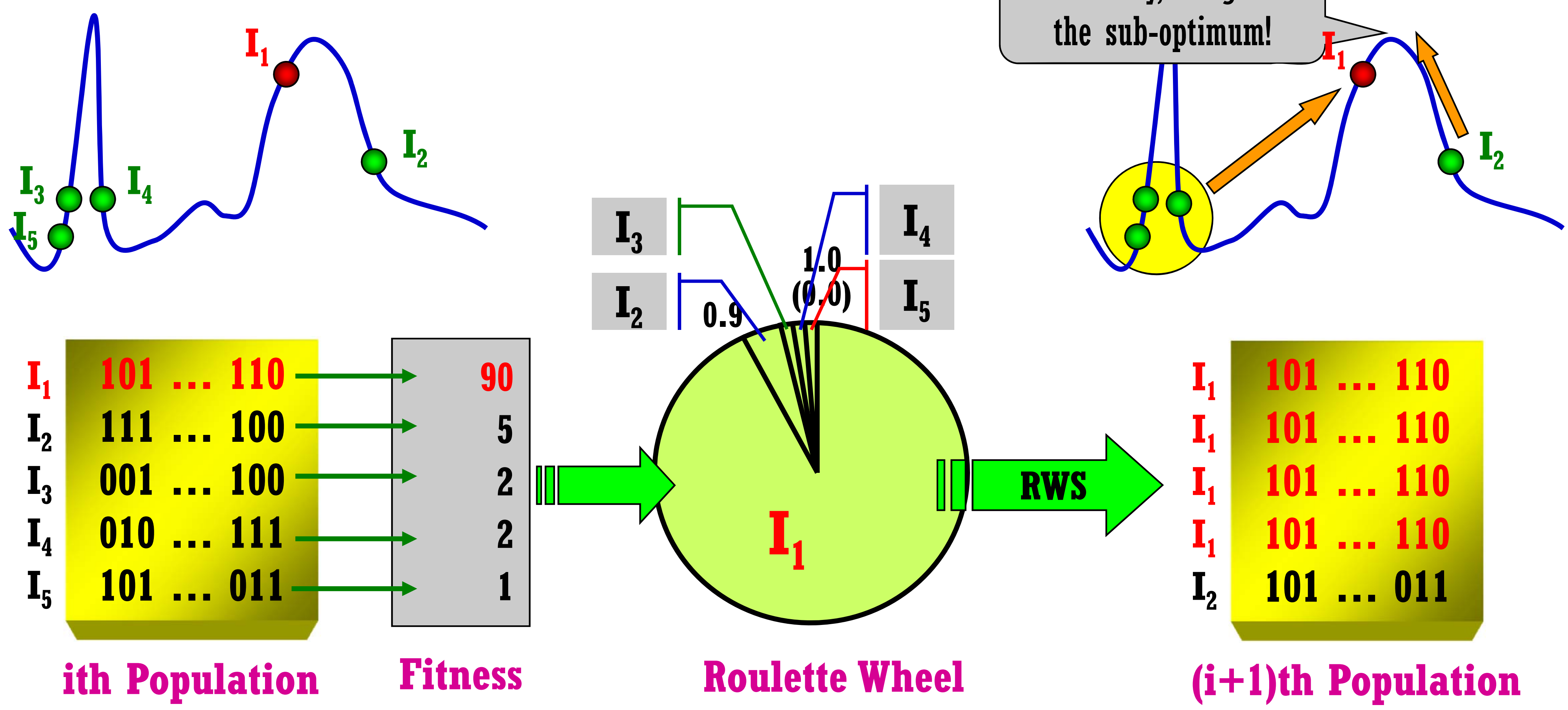
- **What problem** exists in the **proportional selection** (e.g., RWS) ?
  - It is prone to be attracted by the **selection noise**!
  - Thereby, the **premature convergence** can be taken place!



Proportional Selection (2)

Premature Convergence

- The whole population is **too early** converged (into **a sub-optimum**)
  - (Population) **diversity decreases** too **fast!**



Proportional Selection (3)

Scaling

❖ We can **relax** the weakness of the proportional selection

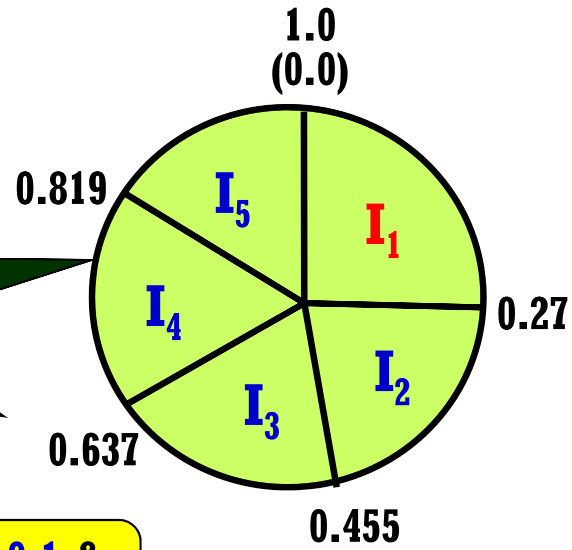
➢ Linear (or non-linear) **scaling** of the fitness

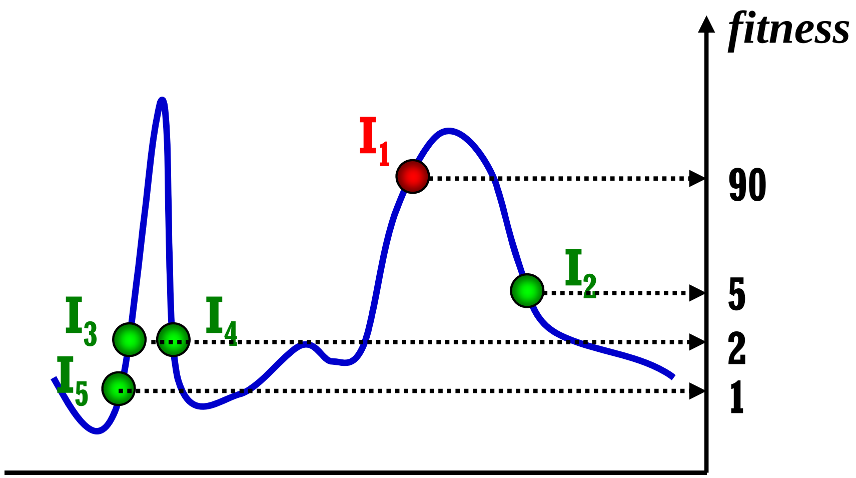
$$f = a \cdot fitness + b$$

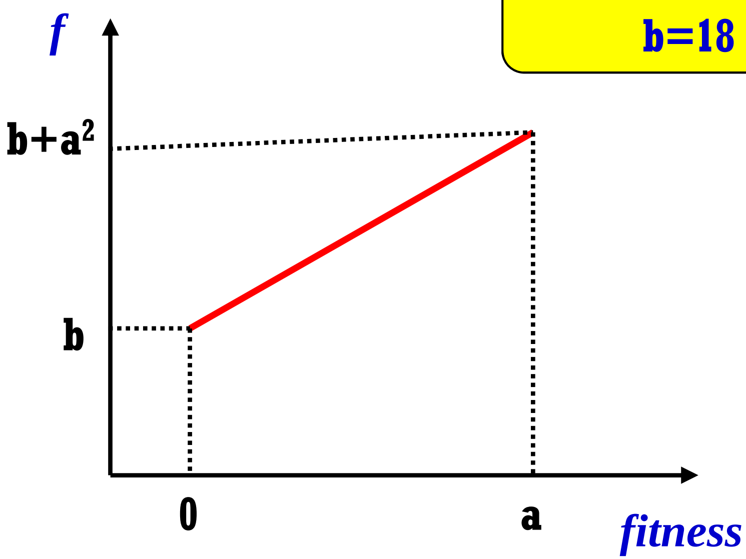
where $\overline{f} = \overline{fitness}$ and $f_{\max} = \varphi \cdot \overline{f}$

Continual re-scaling is needed.
How to set a & b ?

Roulette Wheel







With a=0.1 & b=18

	Fitness	f
I_1	90	27
I_2	5	18.5
I_3	2	18.2
I_4	2	18.2
I_5	1	18.1

Ordinal Selection (1)

Ranking Selection

- **Why** do we **resort to fitness** values?
  - The premature convergence has been brought forth from the fitness value itself
  - A key point in the selection is the **relative dominance** (i.e., **ranking**)!
- Ranking may lose some information, but simpler and more efficient

	ith Population	Fitness	Rank
I_1	101 … 110	80	5
I_2	111 … 100	10	4
I_3	001 … 100	5	3
I_4	010 … 111	3	2
I_5	101 … 011	2	1

- Suppose that the prob. of selecting the kth-rank individual

$$P[k] = \alpha + \beta \cdot k$$

- To be a probability distribution

$$\sum_{k=1}^{N} \alpha + \beta \cdot k = N\left(\alpha + \beta \frac{N+1}{2}\right) = 1$$

How to select individuals?
What criterion can be used for selection?

Ordinal Selection (2)

Ranking Selection (Cont.)

- Selection pressure is defined by

$$\phi = \frac{P[\text{selecting the fittest individual}]}{P[\text{selecting average individual}]}$$

$$\Rightarrow \quad \phi = \frac{\alpha + \beta \cdot N}{\alpha + \beta(N+1)/2}$$

$$\Rightarrow \quad \beta = \frac{2(\phi - 1)}{N(N-1)}, \quad \alpha = \frac{2N - \phi(N+1)}{N(N-1)}$$

which implies that $1 \le \phi \le 2N/(N+1) \simeq 2$

- The cumulative prob. distribution can be stated in terms of the sum of an arithmetic progress.

- With a random number r, finding the k is given by

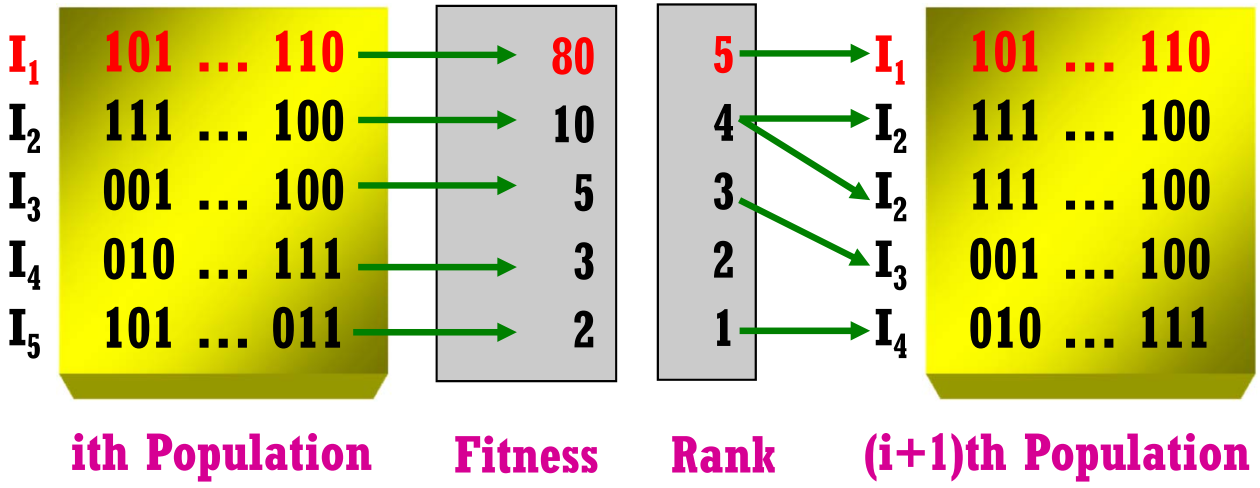
$$\sum_{i=1}^{k} (\alpha + \beta \cdot i) = \alpha \cdot k + \beta \frac{k(k+1)}{2} = r$$

$$\Rightarrow \quad k = \frac{-(2\alpha + \beta) + \sqrt{(2\alpha + \beta)^2 + 8\beta \cdot r}}{2\beta}$$

➔ With a given random number, the individual of the $\lceil k \rceil$-th rank can be selected!

Ordinal Selection (3)

Ranking Selection (Cont.)

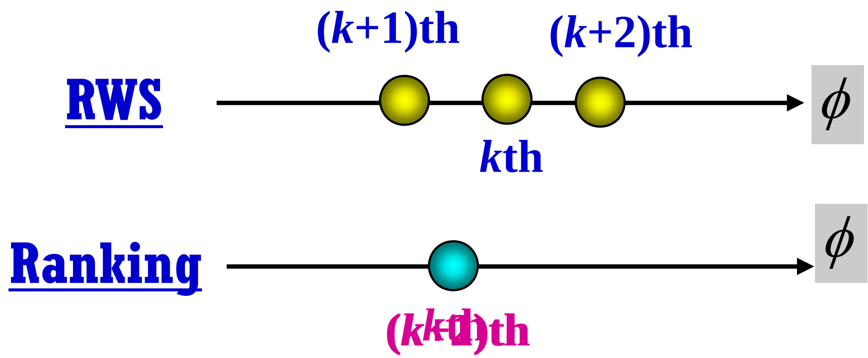


- Using the ordinary proportional selection, it takes *O(logN)*
- But the ranking takes *O(NlogN)* by the sorting algorithm.
- Nevertheless, the prob. of keeping a constant selection pressure without re-scaling is an attractive one!

- For N=5, ϕ =1.5, we can get $\alpha=\beta=1/20$

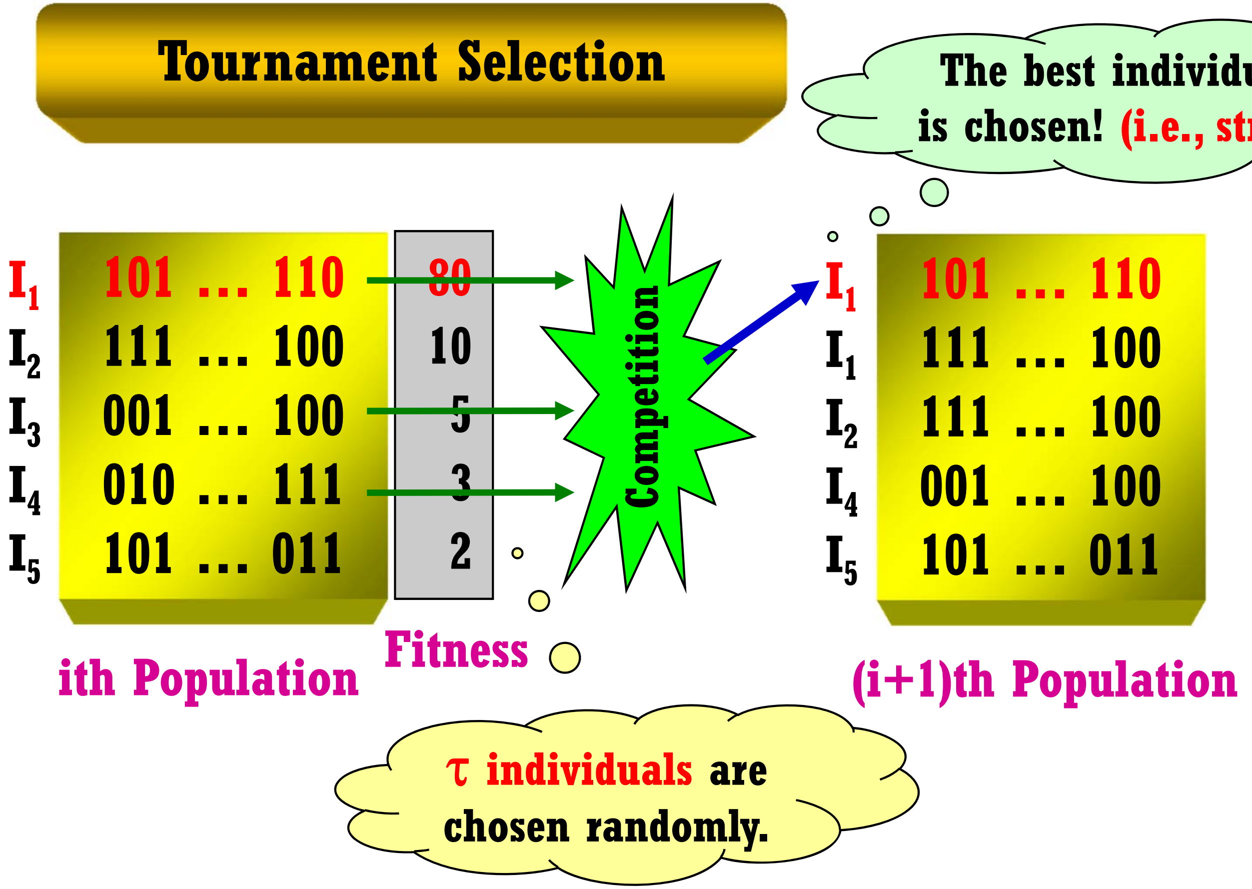
- For given random numbers (0.5, 0.9, 0.4, 0.1, 0.7), we have k=(3.22, 4.68, 2.77, 1.0, 4.0)

  ➔ (4, 5, 3, 1, 4) individuals are selected!



Ordinal Selection (4)

Tournament Selection



- In a complete cycle (i.e., generating N individuals), each individual will be compared τ times on average

- Every time it is compared, the best one is selected. (it is called 'strict' tournament.)

- The chance of the median individual being chosen is the prob. that the remaining (τ-1) ones are all worse: $(½)^{\tau-1}$

- Thus, the selection pressure is $\phi = 2^{\tau-1}$

- To obtain a selection pressures below 2, soft tournament can be used such that the chance of winning of the best is $p<1$

- We can get $\phi = 2^{\tau-1}\,p$

- The pair-wise soft tournament selection can produce the selection pressure as the ranking: $\phi = 2\,p$ that exists [0, 2]