Lab 3 - Compare different selection schemes for real-valued representation

CSE, SUSTech

Outline of This Lab

- Teaching Assistant
- How Does A Selection Sheme Work
- Does The Selection Schme Matter?
- Test Functions
- Illustrate The Results!

Teaching assistant

- Ms. Honglin Jin (12531321@mail.sustech.edu.cn)

How Does A Selection Sheme Work

- Parent Selection:
  - ✓Before search operators, from the current population of μ individuals, select λ individuals to reproduce/generate the next generation.
  - ✓$\lambda < \mu$
  - ✓The key issue here is to generate a probability distribution so that individuals can be selected according to such a distribution.
- Survivor Selection:
  - ✓After search operators, from the μ parents + λ generated offspring, select μ individuals to form the next generation.
  - ✓This is also called the (μ+λ) selection strategy.

Parent Selection Schemes

- Fitness proportional selection (FPS) schemes
  - ✓Roulette wheel selection (fitness proportional selection)
  - ✓FPS with scaling: simple scaling
- Rank-based selection schemes
  - ✓Simple rank-based selection
- Tournament selection

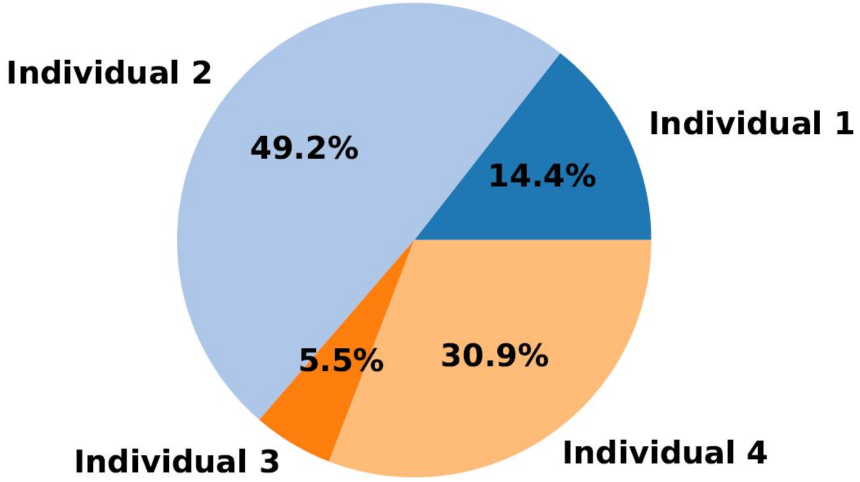
Roulette Wheel Selection

- Also known as “Fitness Proportional Selection”.
- Probability of selecting the individual x:

$$Prob(\mathbf{x}) = \frac{f(\mathbf{x})}{\sum_{\mathbf{x}' \in \mathcal{P}} f(\mathbf{x}')}.$$

- Example: We maximise f!

Idx	$\mathbf{x}$	fitness	$Prob(\mathbf{x})$
1	13	169	0.144
2	24	576	0.492
3	8	64	0.055
4	19	361	0.309



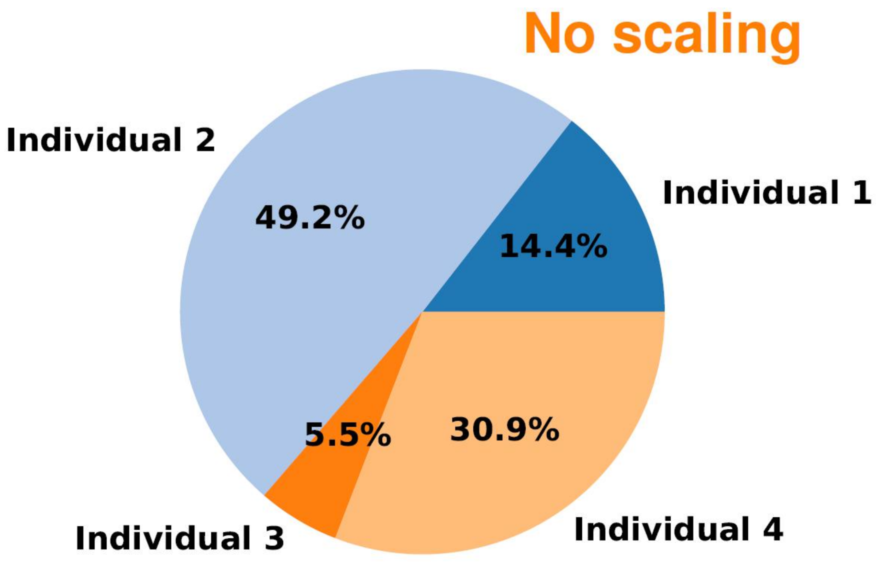
Fitness Scaling: Simple Scaling

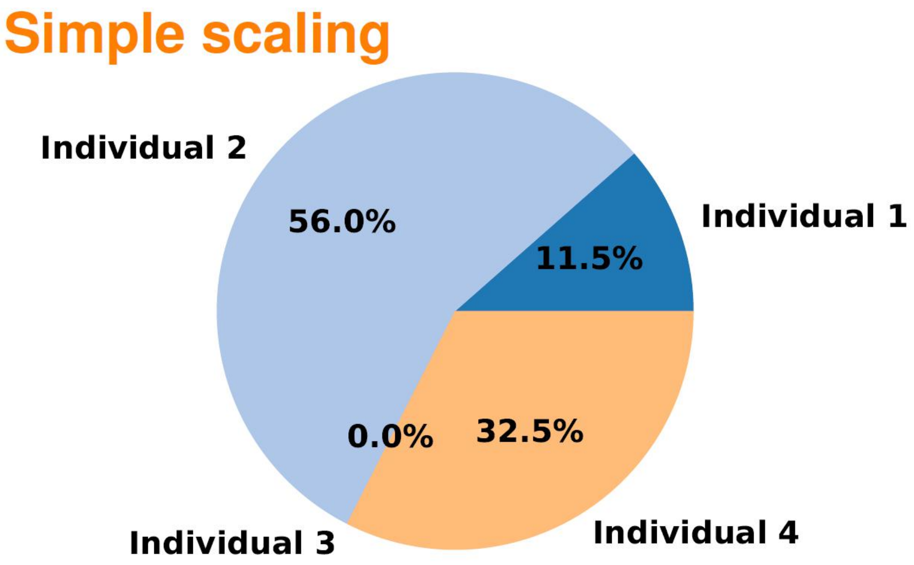
- The new fitness of an individual x is

$$f_{scaled}(x) = f(x) - f_{worst}$$

where f_{worst}, is the fitness of the worst individual so far.

- Example: We maximise f!





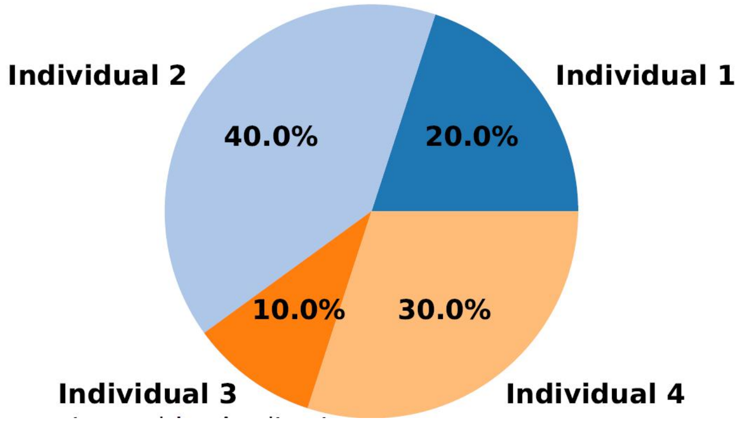
Rank-based Selection

- Rank the individuals from worst to best.
  - ✓ $\forall x \in P$, $rank(x) \in \{0, \ldots, \mu - 1\}$.
  - ✓ Larger rank refers to better individual!
- Probability of selecting the individual x:

$$Prob(\mathbf{x}) = \frac{rank(\mathbf{x}) + 1}{\sum_{\mathbf{x}' \in \mathcal{P}} (rank(\mathbf{x}') + 1)}.$$

- Example: We maximise f!

Idx	x	fitness	rank	$Prob(\mathbf{x})$
1	13	169	1	0.2
2	24	576	3	0.4
3	8	64	0	0.1
4	19	361	2	0.3



Tournament Selection

1. Select k individuals from the population P at random.
2. Select the best individual (the one with the highest fitness) from the k individuals selected above.

- k: tournament size.
- Increasing k → increased selection pressure.

Survivor Selection

- All the presented parent selection schemes can be applied for selecting survivors.
- However, there are some particular survivor selection schemes, in which replacement is explicit.

  → For this reason, the schemes are often referred to as "replacement strategies".

  ✓Age-based

  ✓Fitness-based

Fitness-based Replacement Strategies

- Round-robin tournament
- $(\mu + \lambda)$ selection
- (μ,λ) selection

Round-robin Tournament

- Introduced within Evolutionary Programming (EP).
- Used for
  - ✓choosing μ survivors, or
  - ✓choosing λ parents from a given population of μ.
- Pairwise tournament competitions in round-robin format.
  - ✓Each individual is evaluated against m others randomly chosen from the merged parent and offspring populations ($\mu + \lambda$).
  - ✓For each comparison, a "win" is assigned if the individual is better than its opponent.
  - ✓Finally, the μ individuals with the greatest number of wins are selected.
- m controls the selection pressure.

($\mu + \lambda$) Selection vs. (μ,λ) Selection

- Both come from Evolution Strategies (ES).
- Both for survivor selection
- λ offspring are generated using the μ parents.
- Differences:

($\mu + \lambda$) Selection:

- ($\mu + \lambda$) individuals ranked by fitness, the top μ to form the next generation.

(μ,λ) Selection:

- Mixture of age and fitness.
  - ✓ Age: all parents are discarded.
  - ✓ Fitness: λ Offspring ranked by fitness, the top μ form the next generation.

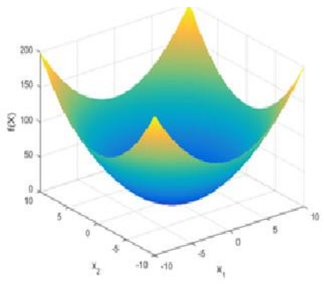
Does The Selection Schme Matter?

- Modify the simple evolutionary algorithm (EA) in Lab 1 with the real-valued representstion and the following Selection Schemes:
  - ✓Parant Selection Schemes:
    - ➢Roulette wheel selection (fitness proportional selection)
    - ➢FPS with scaling: simple scaling
    - ➢Simple rank-based selection
    - ➢Tournament selection
  - ✓Survivor Selection Schemes:
    - ➢$(\mu + \lambda)$ selection

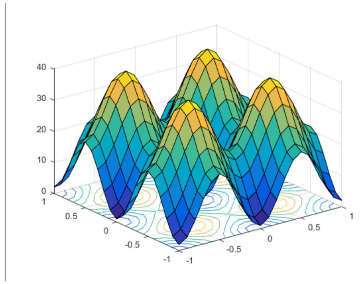
Test Functions
-7 unimodal benchmark functions

- Unimodal functions: f_1-f_5
- f_6 is the step function (one minimum, discontinuous).
- f_7 is a noisy quartic function, where random[0, 1) is a uniformly distributed random variable in [0, 1).

Test function	n	S	f_{min}
$f_1(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$	30	$[-100, 100]^n$	0
$f_2(\mathbf{x}) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	$[-100, 100]^n$	0
$f_3(\mathbf{x}) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	30	$[-10, 10]^n$	0
$f_4(\mathbf{x}) = \max_i \{\lvert x_i \rvert, 1 \leq i \leq n\}$	30	$[-100, 100]^n$	0
$f_5(\mathbf{x}) = \sum_{i=1}^{n} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-100, 100]^n$	0
$f_6(\mathbf{x}) = \sum_{i=1}^{n} (x_i + 0.5)^2$	30	$[-30, 30]^n$	0
$f_7(\mathbf{x}) = \sum_{i=1}^{n} i x_i^4 + random[0, 1)$	30	$[-1.28, 1.28]^n$	0



Test Functions
-8 multimodal benchmark functions



Test function	n	S	f_{min}
$f_8(\mathbf{x}) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	30	$[-500, 500]^n$	-12569.5
$f_9(\mathbf{x}) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10)]$	30	$[-5.12, 5.12]^n$	0
$f_{10}(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i\right) +20 + e$	30	$[-32, 32]^n$	0
$f_{11}(\mathbf{x}) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]^n$	0
$f_{12}(\mathbf{x}) = \frac{\pi}{n}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] +(y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4),$ $y_i = 1 + \frac{1}{4}(x_i + 1)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a, \\ 0, & -a \le x_i \le a, \\ k(-x_i - a)^m, & x_i < -a. \end{cases}$	30	$[-50, 50]^n$	0
$f_{13}(\mathbf{x}) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2[1 + sin^2(3\pi x_{i+1})] +(x_n - 1)^2[1 + \sin^2(2\pi x_n)]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	$[-50, 50]^n$	0
$f_{14}(\mathbf{x}) = \left[\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i - a_{ij})^6}\right]^{-1}$	2	$[-65.536, 65.536]^n$	1
$f_{15}(\mathbf{x}) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2+b_i x_2)}{b_i^2+b_i x_3+x_4}\right]^2$	4	$[-5, 5]^n$	0.0003075

Experimental Setup

- 15 Test Minimization Functions (7 unimodal + 8 multimodal benchmark functions).
- Population size 30.
- Maximum function evaluation 500, 000.
- 50 independent runs for each function.

Illustrate The Results!

Plot 15 figures with one for each test function

- ✓x-axis: current generation number.
- ✓y-axis: average fitness value of the best individual of current population over 50 runs.