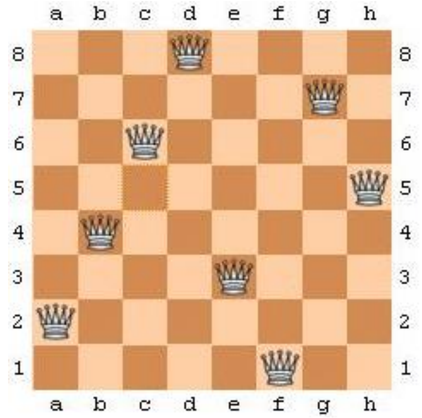
COMP815 Nature Inspired Computing

Evolution Strategy

Last Week

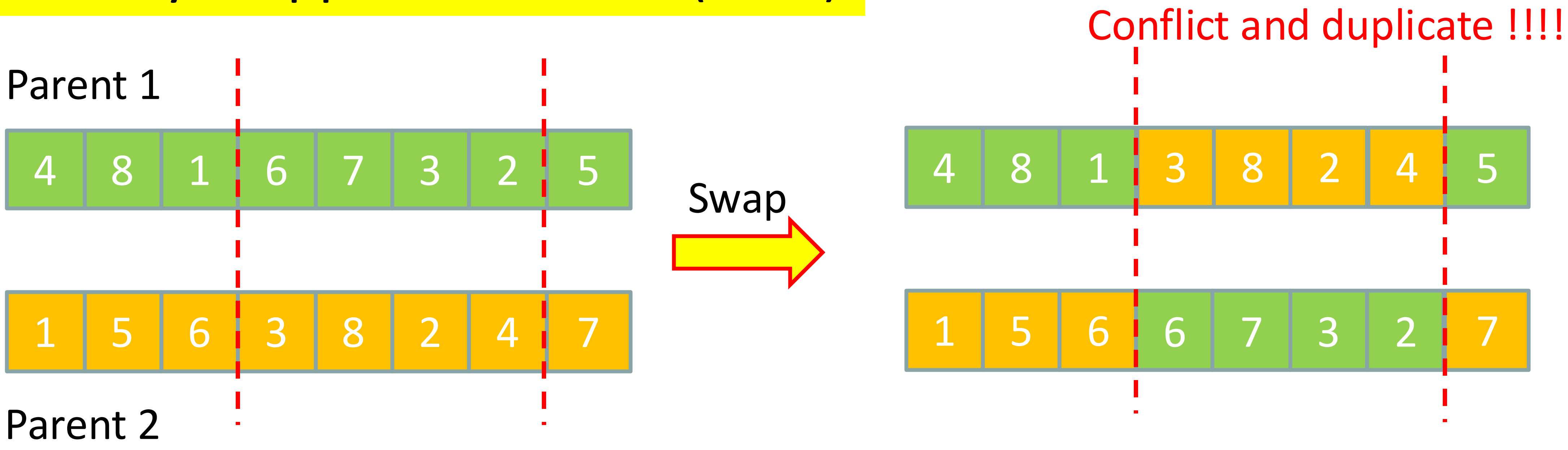
- N-Queen, 8-Queen



- Travelling Salesman Problem
- New Strategies for
  - Chromosome
  - Crossover
  - Mutation

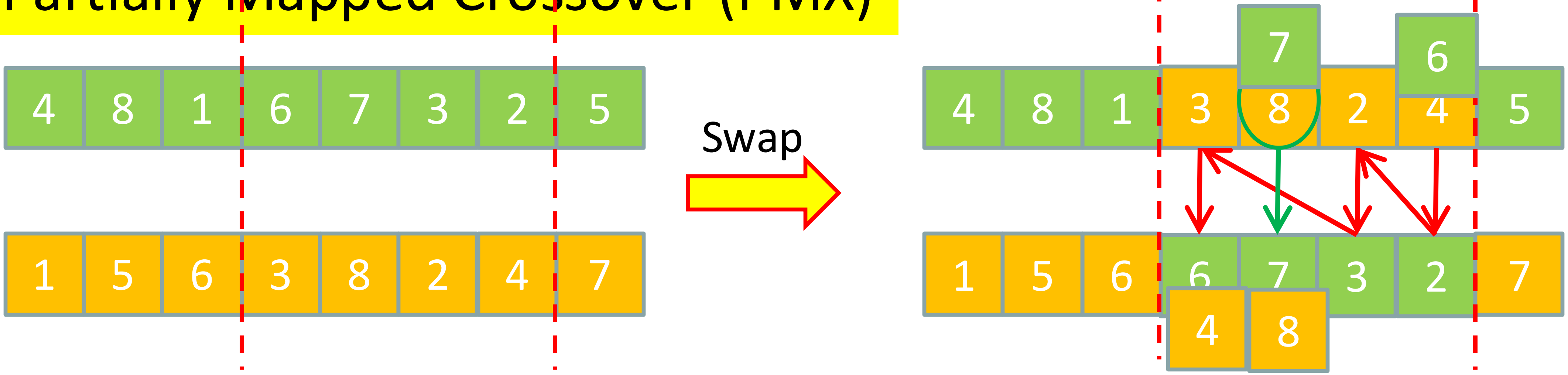
Crossover

Partially Mapped Crossover (PMX)



Crossover

Partially Mapped Crossover (PMX)



Repair with Replacement:

3: no need to replace

8 <-- --> 7

2 : no need to replace

4 <-- --> 2 <-- --> 3 <-- --> 6

Crossover

Partially Mapped Crossover (PMX)

2 Valid Offsprings !!

4	8	1	6	7	3	2	5
1	5	6	3	8	2	4	7

Offsprings

4	8	1	3	7	2	6	5
1	5	6	4	8	3	2	7

Repair with Replacement:

3: overlapping, no need to replace

8 <-- --> 7

2 : overlapping, no need to replace

4 <-- --> 2 <-- --> 3 <-- --> 6

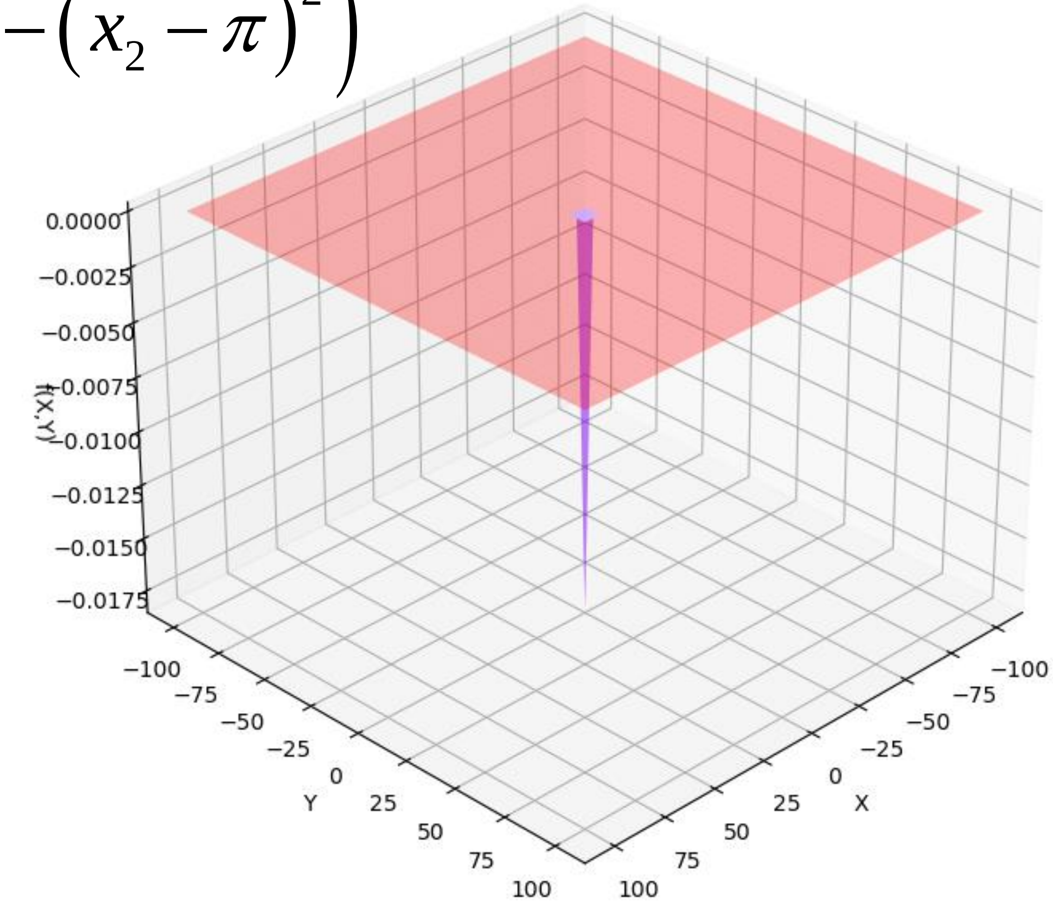
Evolution Strategy – Overview

- Typically used for solving continuous optimization problems
- Does not rely on gradient computation
- Better theoretic understanding

- ES does not always have crossover (recombination) – offsprings can be produced by mutating a single parent
- Use self-adaptation – change its rate of diversity generation (mutation step size) in response to feedback

Easom Function

$$f(x) = -\cos(x_1)\cos(x_2)\exp\left(-(x_1-\pi)^2-(x_2-\pi)^2\right)$$



Global minimum

$$x^* = (\pi, \pi)$$

$$f(x^*) = -1$$

Characteristics

	Evolution Strategy	Genetic Algorithm
Representation	Real-valued vectors	Chromosome
Recombination	Discrete or intermediary	Crossover of parents
Mutation	Gaussian random	Bit flip or swap
Parent selection	Uniform random	Rank-based
Survivor selection	(μ, λ) or $(\mu + \lambda)$	Remove worst individuals

Population size = μ λ offsprings produced

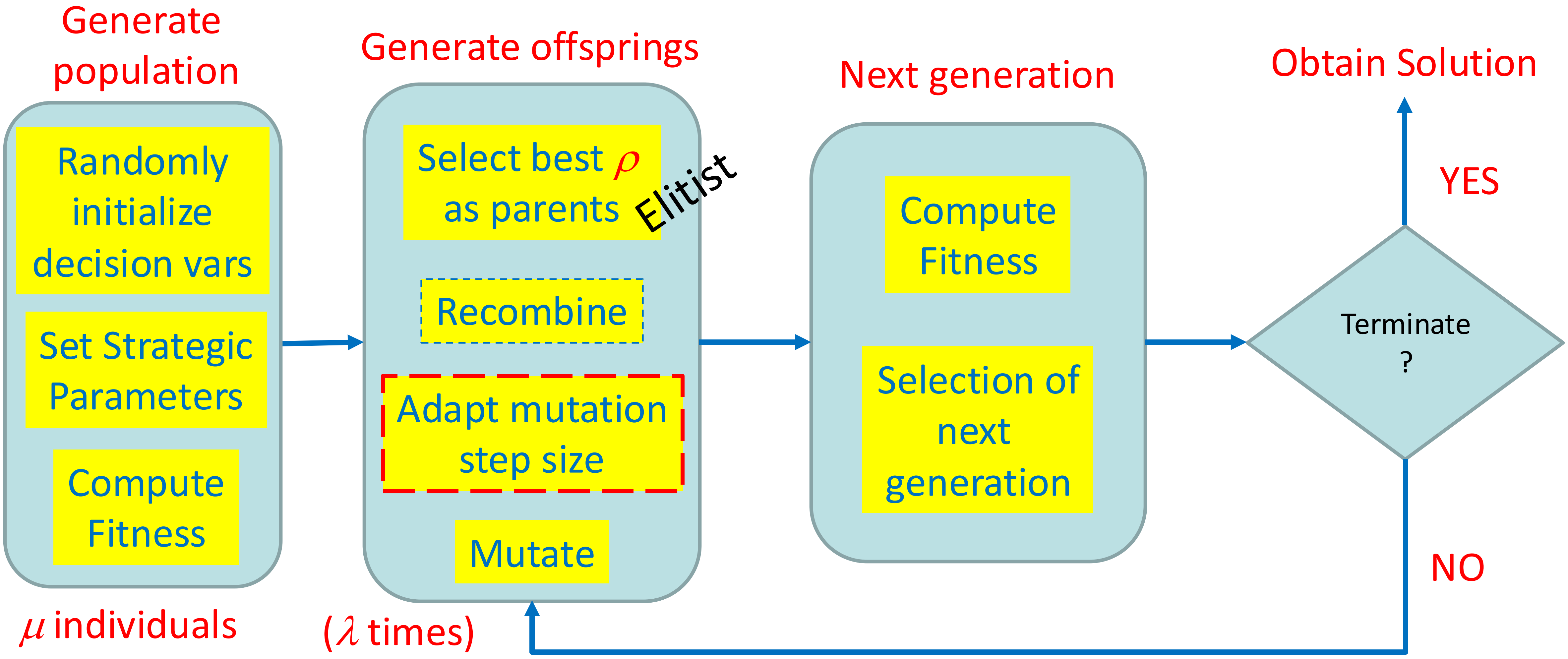
$(\mu, \lambda) \Rightarrow \mu$ offsprings out of λ chosen as new generation $\lambda > \mu$

$(\mu + \lambda) \Rightarrow \mu$ of the fittest $\mu + \lambda$ chosen as new generation

Individuals (Chromosome)

- Each individual k in a population of μ is composed of
  - Decision variables $\mathbf{x}_k = (x_1, \dots, x_D)_k$ *e.g.*, (x, y, z) axis of a drone
  - Fitness $f\left(\mathbf{x}_k\right)$
  - A set of evolvable strategy parameters $\mathbf{s}_k$
  - Similar to Chromosome
    But continuous and includes evolvable parameters

Basic ES Algorithm



Recombination / Crossover

Can use more than 2 parents

Each set of parents only produce one offspring

Intermediate recombination:

For each decision variable,

take the average of the values of that variable of the parents

Discrete dominant recombination

For each decision variable,

uniformly randomly choose one value from the parents

Mutation

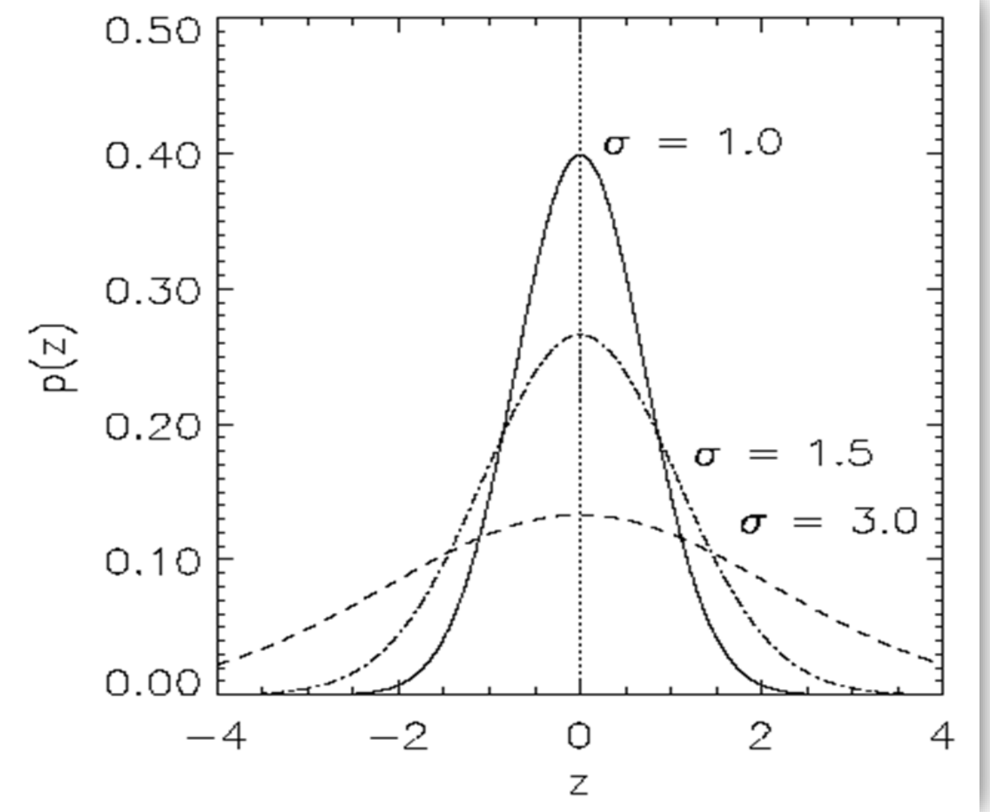
Add a normally distributed value to the current variable

$$x_i = x_i + z$$

Mutation step size (standard deviation)

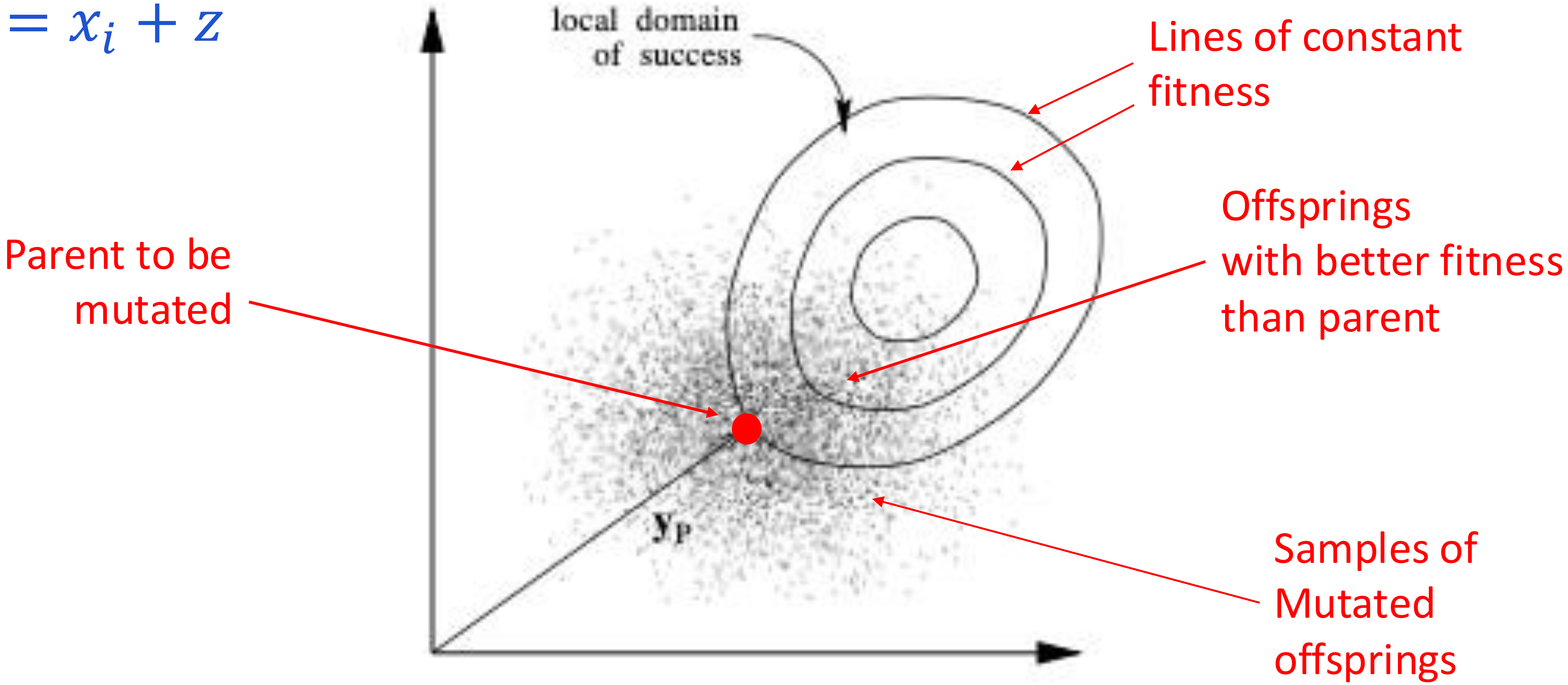
$$z := \mathcal{N}(0, \sigma)$$

Normal distribution



2D Example

$x_i = x_i + z$



Adaptation of Strategy Parameter $\boldsymbol{\sigma}$

- σ is varied on-the-fly by the 1/5th-rule:

$$\sigma = \begin{cases} \sigma / c & \text{if} \quad p_s > 1/5 \\ \sigma & \text{if} \quad p_s = 1/5 \\ \sigma \cdot c & \text{if} \quad p_s < 1/5 \end{cases}$$

Exploration (if $p_s > 1/5$)

Exploitation (if $p_s < 1/5$)

p_s is the success probability that an offspring replaces a parent

Keep $\boldsymbol{\sigma}$ constant for G generations

G_s = number of successful mutations

$$p_s = G_s / G$$

Change $\boldsymbol{\sigma}$ according to 1/5 rule

Self-adaptation

- Each individual's strategy parameters could undergo evolution
- May undergo recombination
- Always subject to mutation
- Strategy parameters of fitter individuals will also be passed down the generations
- Individuals could learn more optimal strategy parameters
- Most commonly adapted parameter is σ

Self-adaptation of σ

$$\hat{\sigma} = \begin{cases} \sigma\alpha, & \mathcal{U}(0,1] \leq 0.5 \\ \sigma/\alpha, & \mathcal{U}(0,1] > 0.5 \end{cases} \qquad \alpha > 1$$

A uniform random number

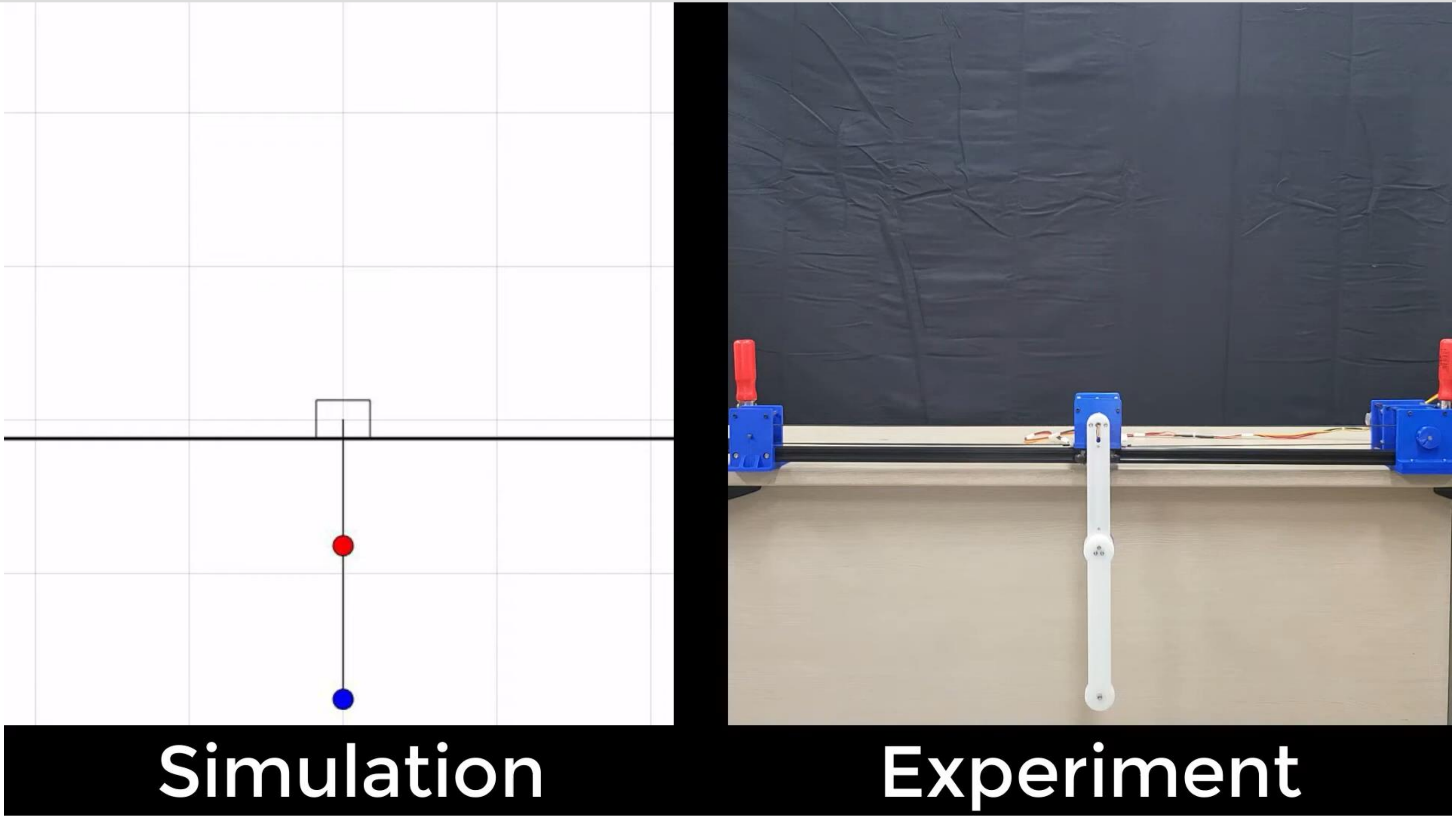
Commonly used values:

$$\alpha = 1 + \frac{1}{\sqrt{N}} \qquad \alpha = 1 + \frac{1}{\sqrt{2N}}$$

Recent Application

Recently applied to reinforcement learning to solve the inverted pendulum problem

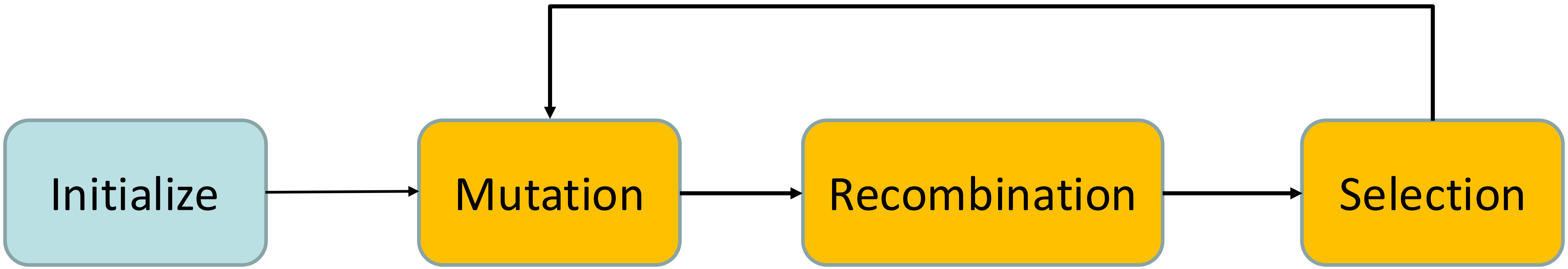
Recent Application



Differential Evolution

- Introduced in 1996
- Originally for continuous optimization problems
- Also does not require gradient calculation

Basic Algorithm



Individuals

- Suppose there are *D* decision variables in the optimization problem
- An individual *i* in generation *G* is represented by a vector of *D* real numbers

$$x_{i,G} = [x_{1,i,G} \ \ x_{2,i,G} \ \ \cdots \ \ x_{D,i,G}]$$

- Minimum population size is 4

Mutation

- For each $x_{i,G}$ (target vector) randomly select three other individuals $x_{r1,G}$ $x_{r2,G}$ $x_{r3,G}$

$$\boxed{i \neq r1 \neq r2 \neq r3}$$

- Compute donor vector by

$$v_{i,G+1} = x_{r1,G} + F\left(x_{r2,G} - x_{r3,G}\right)$$

- F is a constant

$$\boxed{0 < F \leq 2}$$

Recombination

- Obtain trial vector from elements of target vector and donor vector:

$$u_{j,i,G+1} = \begin{cases} v_{j,i,G+1} & \text{if } \mathrm{rand}_{j,i} \leq \mathrm{CR} \text{ or } \quad j = I_{\mathrm{rand}} \\ x_{j,i,G} & \text{if } \mathrm{rand}_{j,i} > \mathrm{CR} \text{ and } \quad j \neq I_{\mathrm{rand}} \end{cases}$$

$i = 1,2, \dots, N$ (*Individual*) $\quad j = 1,2, \dots, D$ (*Variable*) $\quad \mathrm{rand}_{j,i} \sim U[0,1]$

$x_{i,G} = \left[x_{1,i,G} \;\; x_{2,i,G} \;\; \cdots \;\; x_{D,i,G} \right]$

I_{rand} is an integer from $[1,2,\dots,D]$

Ensures that $u_{i,G+1} \neq x_{i,G}$

Recombination

- Obtain trial vector from elements of target vector and donor vector:

$$u_{j,i,G+1} = \begin{cases} v_{j,i,G+1} & \text{if } rand_{j,i} \leq CR \text{ or } j = I_{rand} \\ x_{j,i,G} & \text{if } rand_{j,i} > CR \text{ and } j \neq I_{rand} \end{cases}$$

$v_{j,i,G+1}$	$j = 1$	$j = 2$	...				$j = D$
$u_{j,i,G+1}$		?	...				
$x_{j,i,G}$	$j = 1$	$j = 2$	...				$j = D$

Selection

- Given a *minimization* problem with objective function *f*(*x*)
- The next generation is selected by

$$x_{i,G+1} = \begin{cases} u_{i,G+1} & \text{if } f\left(u_{i,G+1}\right) \leq f\left(x_{i,G}\right) \\ x_{i,G} & \text{otherwise} \end{cases} \qquad i = 1,2,\ldots,N$$

Performance

- It has been shown to be effective on a large range of classic optimization problems

- Compared with GA in general
  - More efficient
  - Better solutions

Summary

- Evolution Strategy VS. Genetic Algorithm

- Recombination and Mutation
  Gaussian $N(0, \sigma)$

- Differentiable Evolution

Explanation of the Workshop Code

- Jupyter Notebook
- https://deap.readthedocs.io/en/master/api/tools.html