COMP815 Nature Inspired Computing

Evolutionary Algorithms –
Genetic Algorithm

Timetable and Assessment

- Assessment in *Assignment* Section

  Part 1: Coding + analysis [*Student ID updated*]

  Part 2: Project report + Portfolio

- Course Format: 1 hour Lecture + 1 hour Workshop
- Timetable and Deadlines in *Announcements*

- Q&A: preferably *Teams*

Course Objectives

- Focus on
  - Evolutionary and genetic algorithms → Biologically inspired
  - Swarm algorithms → Socially inspired
  - Neural Networks → Brain inspired

Mathematical Description

Objective function

Minimize $f(\boldsymbol{x})$

parameters

$\boldsymbol{x} = [x_1, x_2, \dots, x_N]$

Subject to $\boldsymbol{x} \in M$

Feasible set

M is finite = discrete optimization

M is uncountably infinite = continuous optimization

Example of Practical Optimization Problem

A butter production company wants to optimize the use of the machineries in its daily production of butter. Two types of butter are made – sweet and raw. One kilogram of sweet butter gives the company a profit of $10 and one of raw a profit of $15. Two machines are used in the production: a pasteurization machine and a whipping machine. The daily use time of the pasteurization machine is 3.5 hours and 6 hours for the whipping machine. The processing times (in minutes) are given below:

Machine	Sweet butter	Raw butter
Pasteurization	3	3
Whipping	3	6

Any Question so far?

Background

- Inspired by Neo-Darwinsim
  - Darwin's hypothesis of Natural Selection – *survival of the fittest* (most adaptable)
  - Discovery of the genetic code in DNA

- Evolutionary Algorithms
  - Started in the 1960's
  - Simple, general-purpose, global optimization method

Evolutionary Algorithms

Evolution	Problem-solving
Environment ⟷	Problem
Individual ⟷	Candidate Solution
Fitness ⟷	Quality

Algorithm Structure

Initialise a population of candidate solutions

Repeat

Select individuals (parents) for breeding from current population

Generate new individuals (offsprings) from these parents

Replace some or all of the current population with new individuals

Until some terminating criteria are satisfied

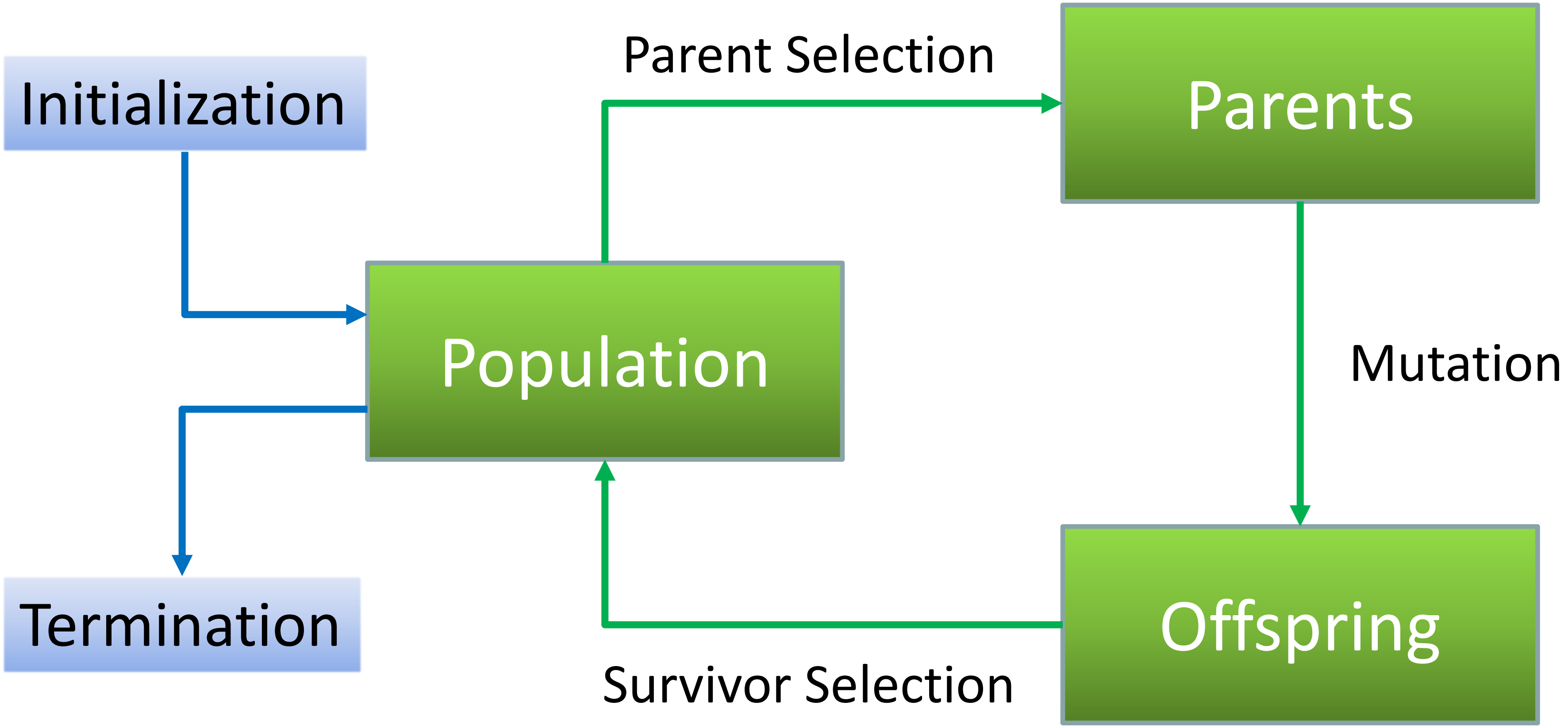
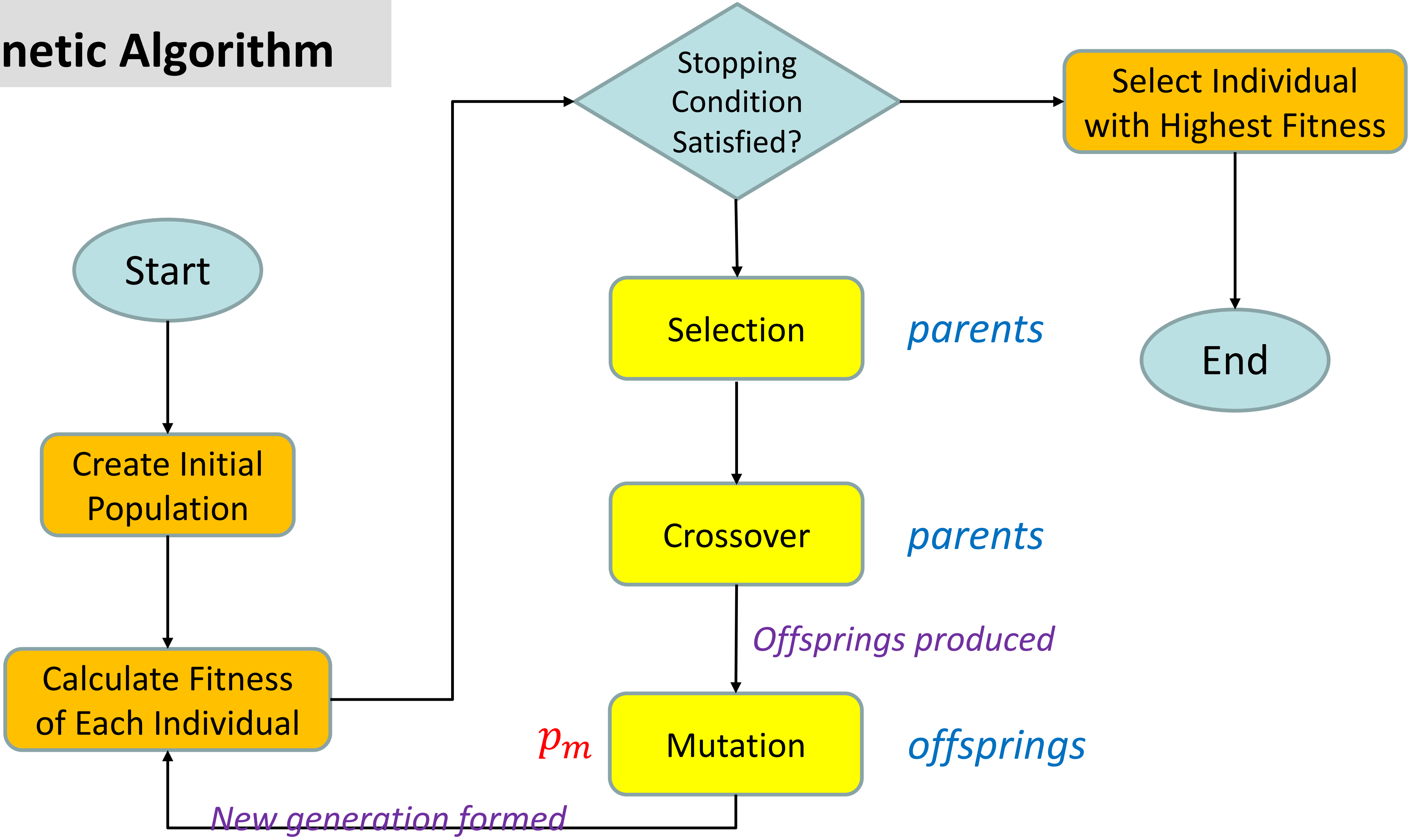


Fig. 3.2 of Introduction to Evolutionary Computation

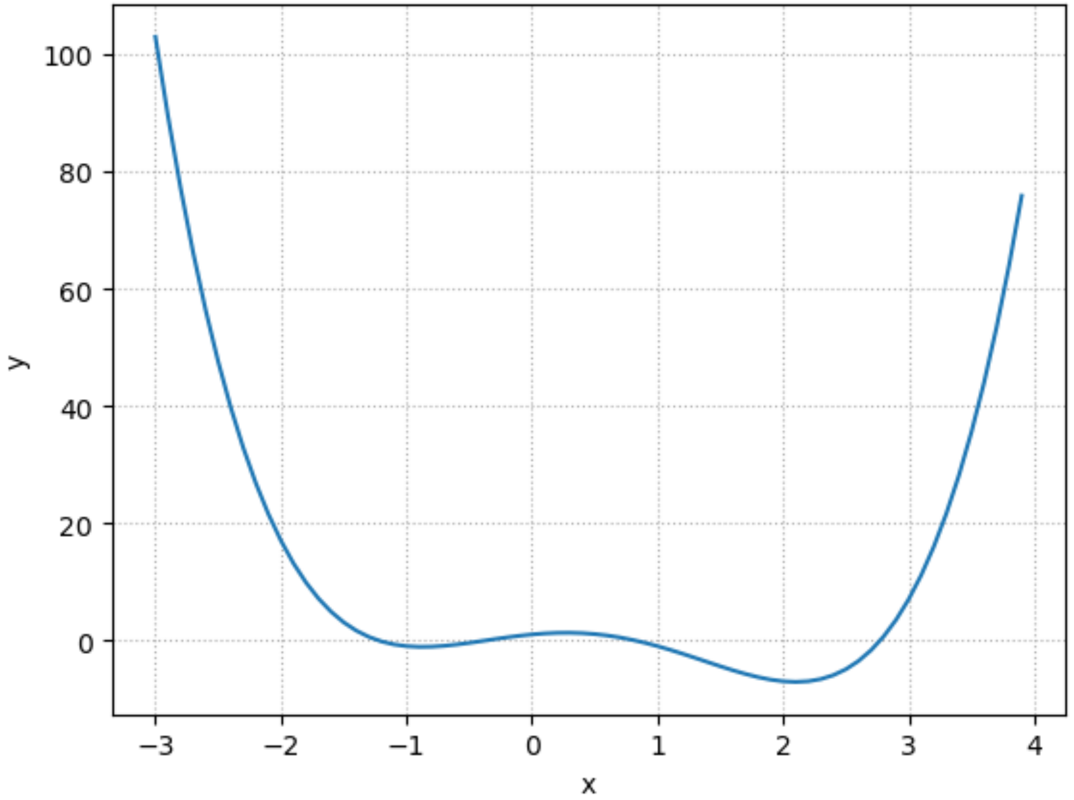
Basic Genetic Algorithm



Example

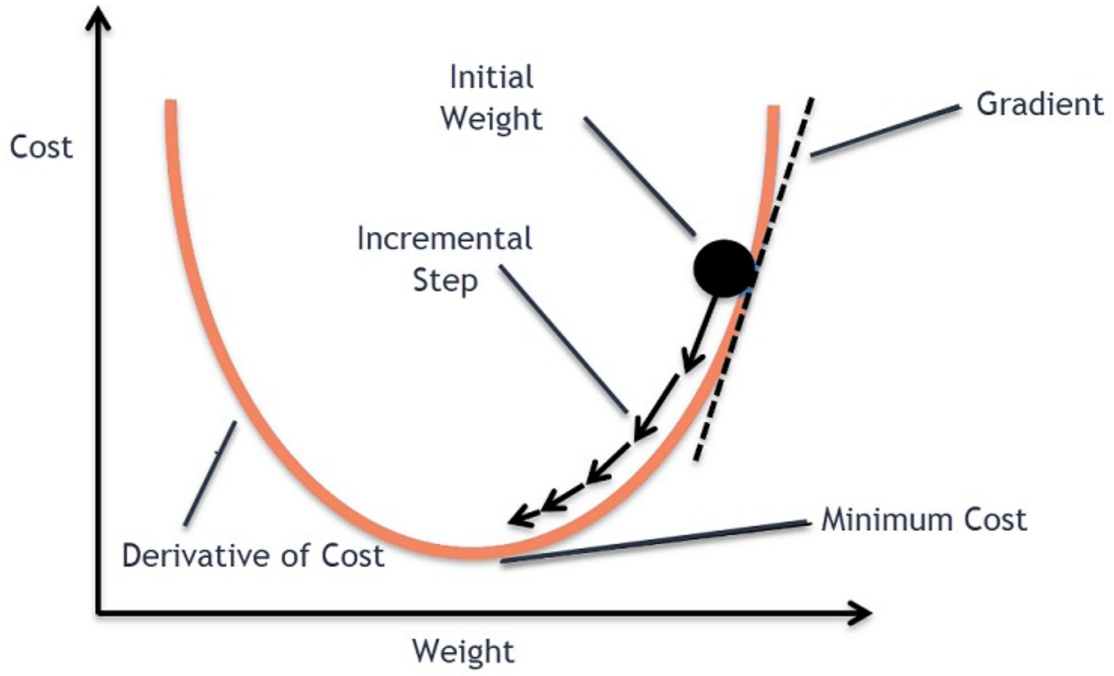
$$\min \quad f(x) = x^4 - 2x^3 - 3x^2 + 2x + 1$$

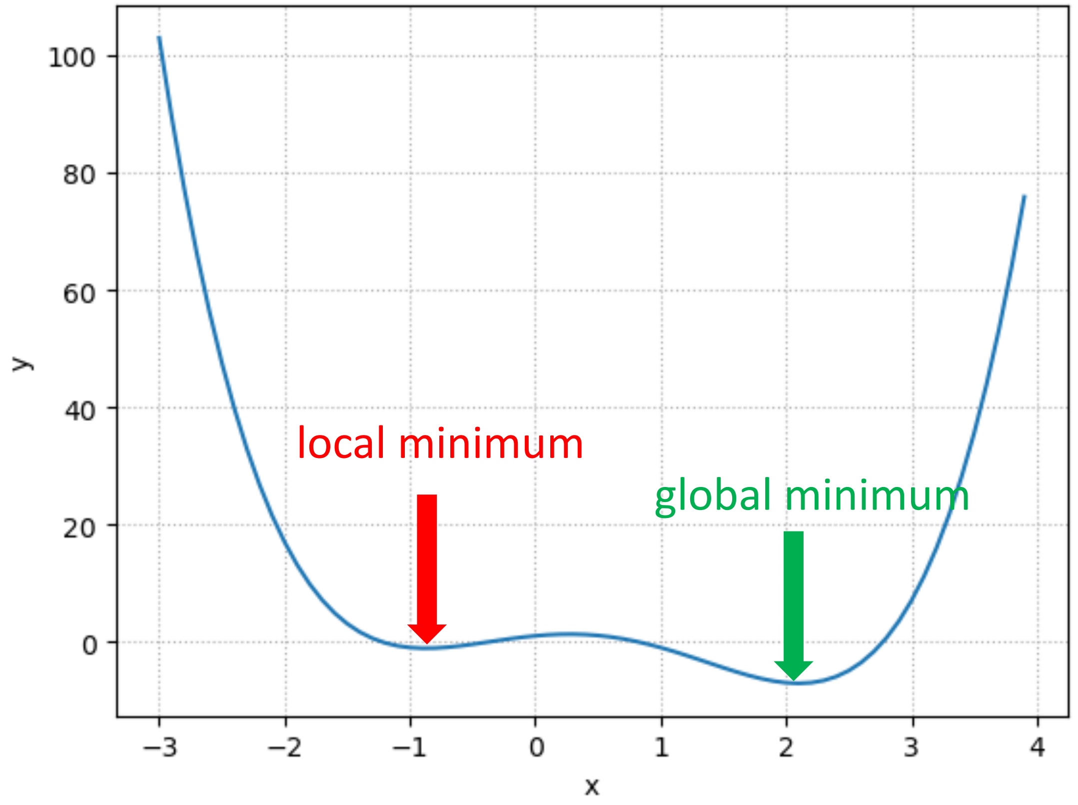
$$\text{for} \quad -3 \leq x \leq 4$$



Example

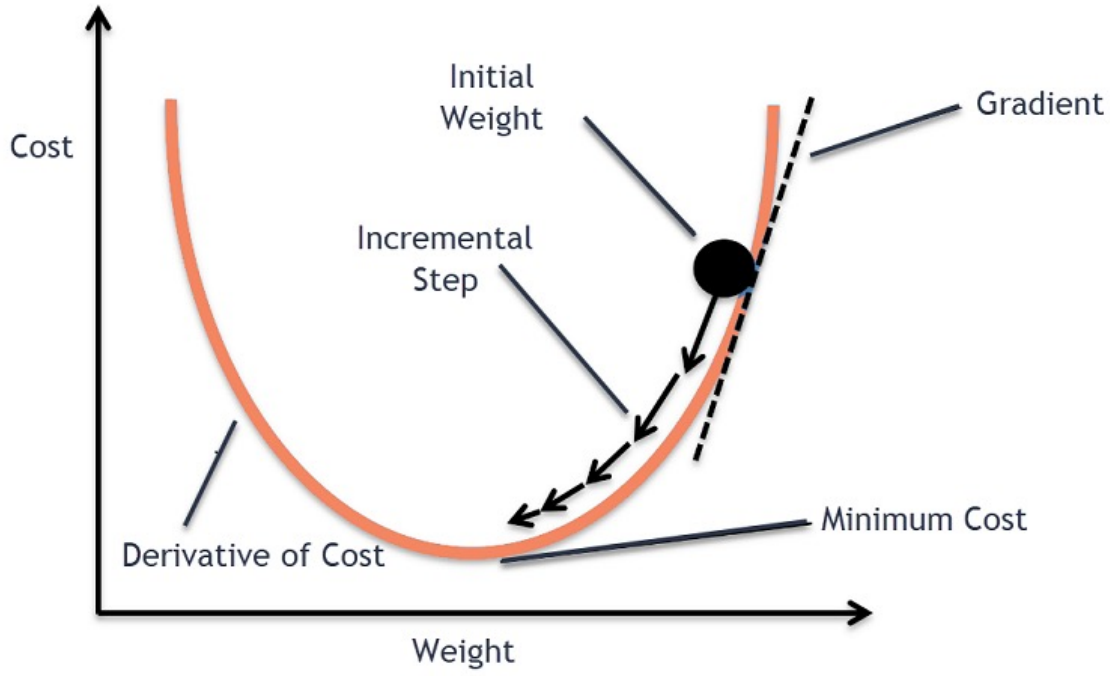
Gradient Descent Algorithm

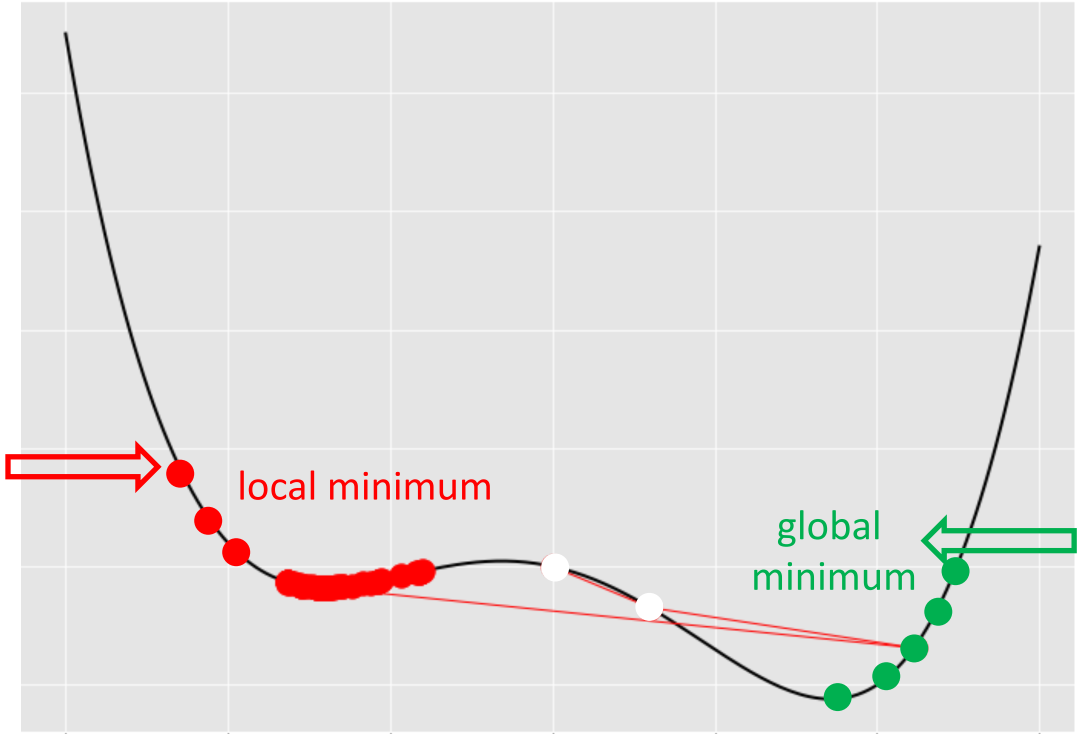




Example

Gradient Descent Algorithm





Chromosome Encoding

Each candidate solution encodes a value of $-3 \leq x \leq 4$

Each candidate solution is represented by a *chromosome*

Chromosomes are strings of *n* binary digits: $\{0,1\}^n$

$$x \rightarrow round\left(\left(2^n - 1\right) \cdot \frac{x-a}{b-a}\right)$$

for $x \in [a,b]$

Let *n* = 10

x	chromosome
-3	00 0000 0000
0	01 1011 0110
4	11 1111 1111

Decoding

For a chromosome *s* (in binary),

Convert to its decimal equivalent value *dec*(*s*)

It represents a value of

$$x = a + dec(s) \cdot \frac{b-a}{2^n - 1}$$

n=10 *a* = -3 *b* = 4

chromosome	x
00 0000 0000	-3
01 1011 0110	- 0.003
11 1111 1111	4

Any Question so far?

Initial Population

1. Decide on the size of initial population
2. Generate random bits for each individual

0	0	1	0	1	0	1	0	1	0

1	0	1	0	0	0	1	0	1	1

0	1	0	0	0	1	1	1	1	0

1	1	1	0	1	1	1	0	0	0

Compute Fitness of Each Candidate

										x	$f(x)$
0	0	1	0	1	0	1	0	1	0	-1.837	10.980
1	0	1	0	0	0	1	0	1	1	1.455	-4.117
0	1	0	0	0	1	1	1	1	0	-1.043	-0.896
1	1	1	0	1	1	1	0	0	0	3.514	36.692

Selection of Parents

Individuals with higher fitness should have a higher probability of being selected – “survival of the fittest”

Rank-based Selection Mechanism:

Individual ***i*** has a probability of being chosen is $p_i = \frac{R(i)}{\sum_j R(j)}$

where $R(i)$ is the rank of ***i***

Lower fitness = lower rank

Rank of Each Individual

	x	$f(x)$	*Rank*	*Prob*
0 0 1 0 1 0 1 0 1 0	-1.837	10.980	2	0.2 $= \frac{2}{1+2+3+4}$
1 0 1 0 0 0 1 0 1 1	1.455	-4.117	4	0.4
0 1 0 0 0 1 1 1 1 0	-1.043	-0.896	3	0.3
1 1 1 0 1 1 1 0 0 0	3.514	36.692	1	0.1

What other selection methods are there?

Replacement Policy

Number of parents chosen = Number of offsprings produced

How many individuals from parent generation will be retained?

Our Example:

Replace the entire population by offsprings

i.e. produce 4 offsprings

Crossover (Recombination)

Parents chosen

0	1	0	0	0	1	1	1	1	0

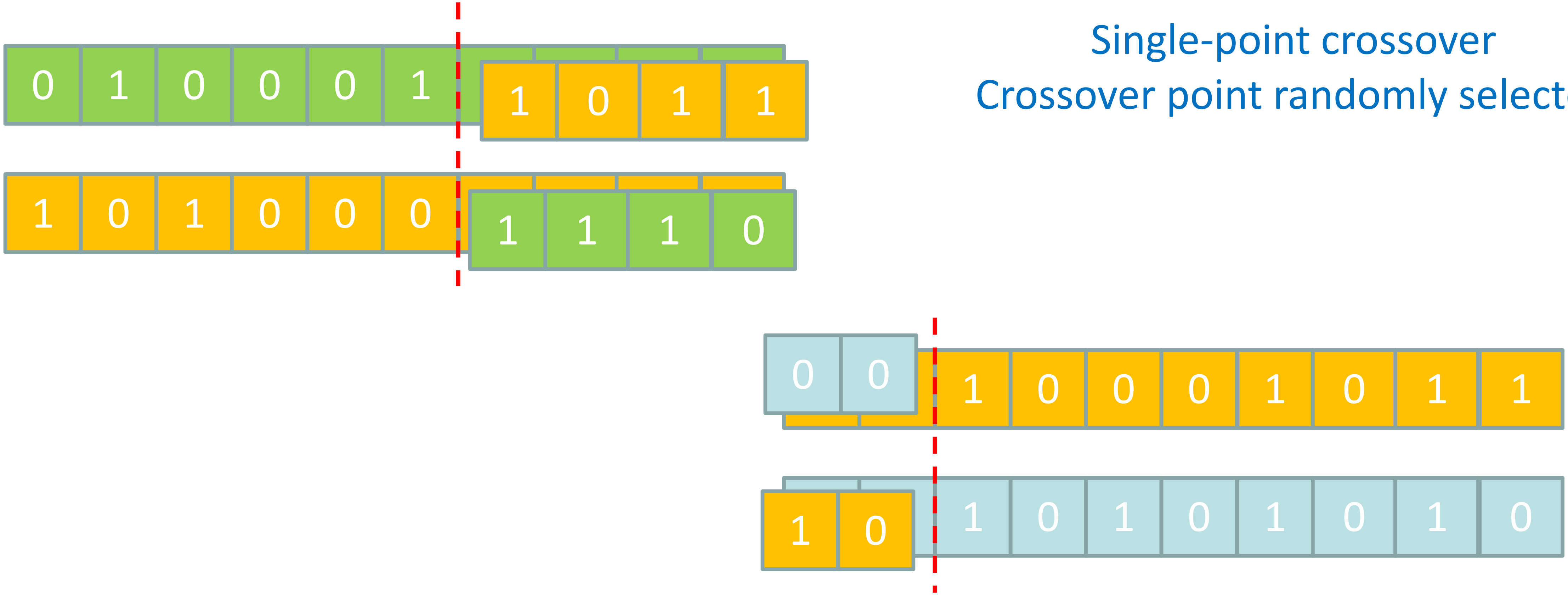
1	0	1	0	0	0	1	0	1	1

1	0	1	0	0	0	1	0	1	1

0	0	1	0	1	0	1	0	1	0

Crossover (Recombination)

Parents chosen



Single-point crossover
Crossover point randomly selected

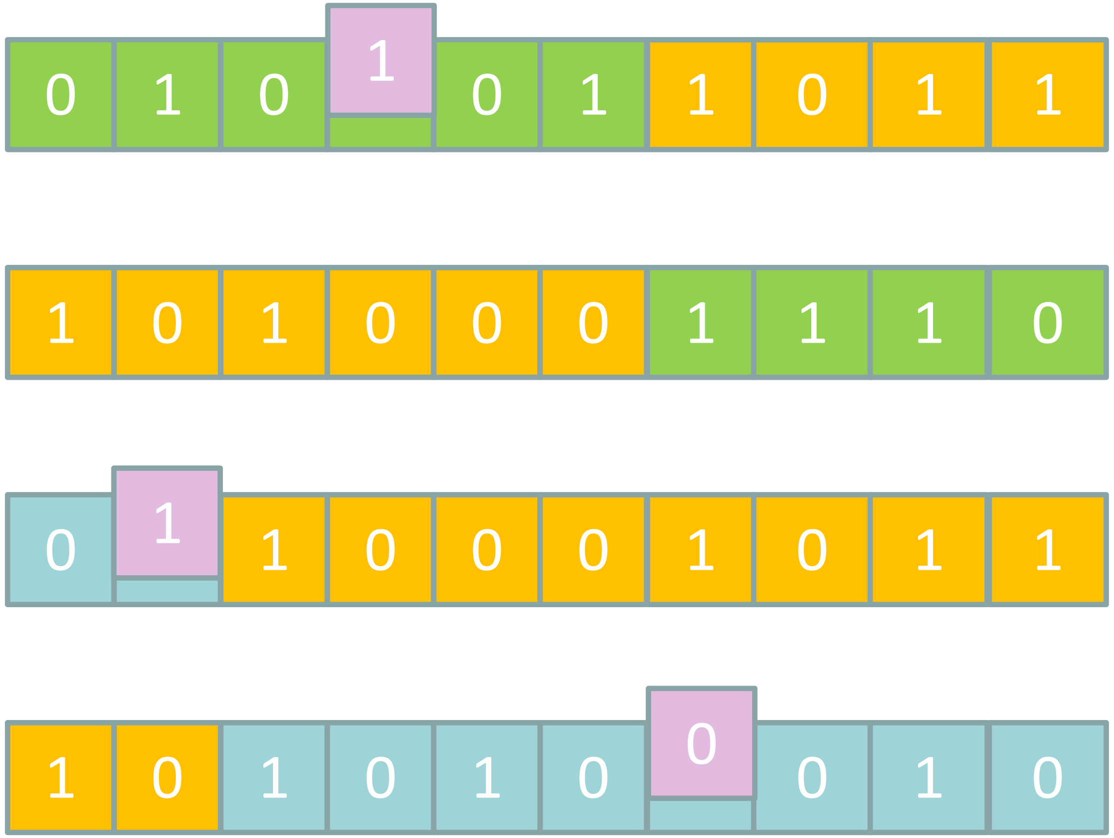
Mutation

Probability of mutation (mutation rate) is usually a small value

Common mutations:

- Single or multiple bit flip
- Swap mutation

Single-bit Mutation



Single-bit Mutation

										x	$f(x)$
0	1	0	1	0	1	1	0	1	1	-0.626	-0.782
1	0	1	0	0	0	1	1	1	0	1.475	-4.262
0	1	1	0	0	0	1	0	1	1	-0.297	0.201
1	0	1	0	1	0	0	0	1	0	1.612	-5.196

Any Question so far?

Exploration vs Exploitation

- Exploration – search for solutions in new, unexplored regions

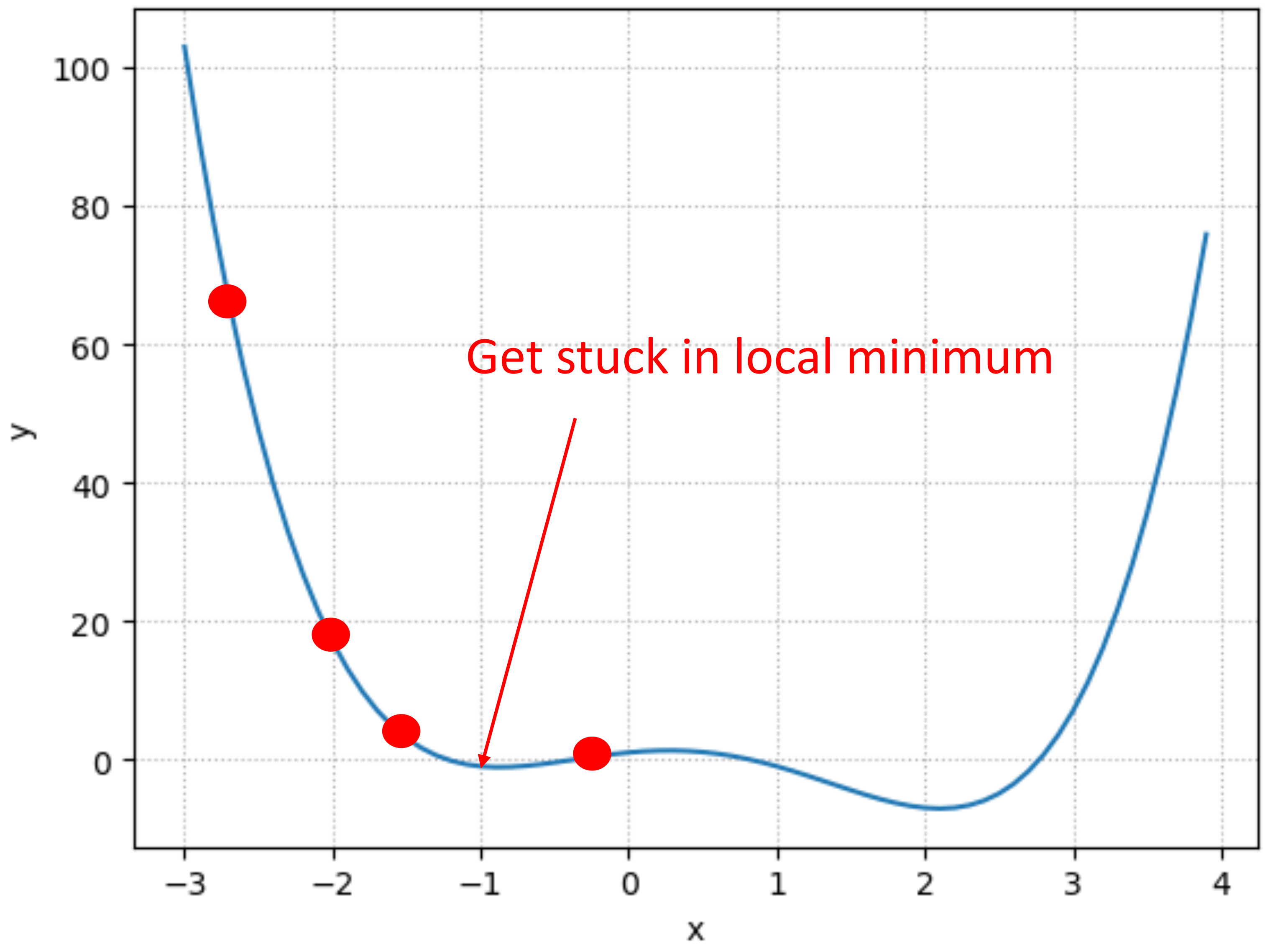
Crossover

Mutation

- Exploitation – Make refinements of existing high fitness solutions

Parent selection

Too much exploitation
Too early



Summary

- Course Overview again
- Evolution Algorithm

- Genetic Example

  Chromosome Encoding

  Parent selection

  Crossover, Mutation