EVOLUTIONARY COMPUTATION

Asst.Prof.Dr.Supakit Nootyaskool
IT-KMITL

บทสุดท้ายแล้ว !!

Study map

- 1.Basic programming
  - R-programming
- 2. Perceptron
  - Activity function
- 3. Feed Forward NN
  - Logistic function
- 4. Feed Forward NN
  - XOR gate
  - Multi-layer perceptron
- 5. Example & Library Feed Forward NN
  - N:N, 1:N model
  - iris dataset

- 6. Writing NN Code
  - Data scaling, Confusion matrix
  - Writing NN code
- 7. Recurrent Neural Network
- 8. Apply RNN & Library
- 9. GRU LSTM
- 10. CNN
- 11 Apply GA to NN

Learning Outcome

- Express about concept of the evolutionary computation helping the escape out from the local solution.
- Call and write genetic algorithm in R
- Apply genetic algorithm to solve a small problem.

OVERVIEW GENETIC ALGORITHM

GA related to NN in the current researches

Wind power prediction based on Elman neural network model optimized by improved genetic algorithm

Zihan Sun; Yanlong Liu; Mingyu Xu; Wanlin Guan

2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)

Year: 2021 | Conference Paper | Publisher: IEEE

Welding quality inspection method based on genetic algorithm to optimize BP neural network

Daoqu Geng; Xingchuan Lan; Hanwen He; Chan Liu

2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)

Year: 2021 | Conference Paper | Publisher: IEEE

Deep Learning Using Genetic Algorithm Optimization for Short Term Solar Irradiance Forecasting

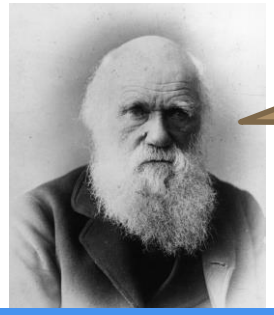
Wadie Bendali; Ikram Saber; Bensalem Bourachdi; Mohammed Boussetta; Youssef Mourad

2020 Fourth International Conference On Intelligent Computing in Data Sciences (ICDS)

What is Genetic algorithm?

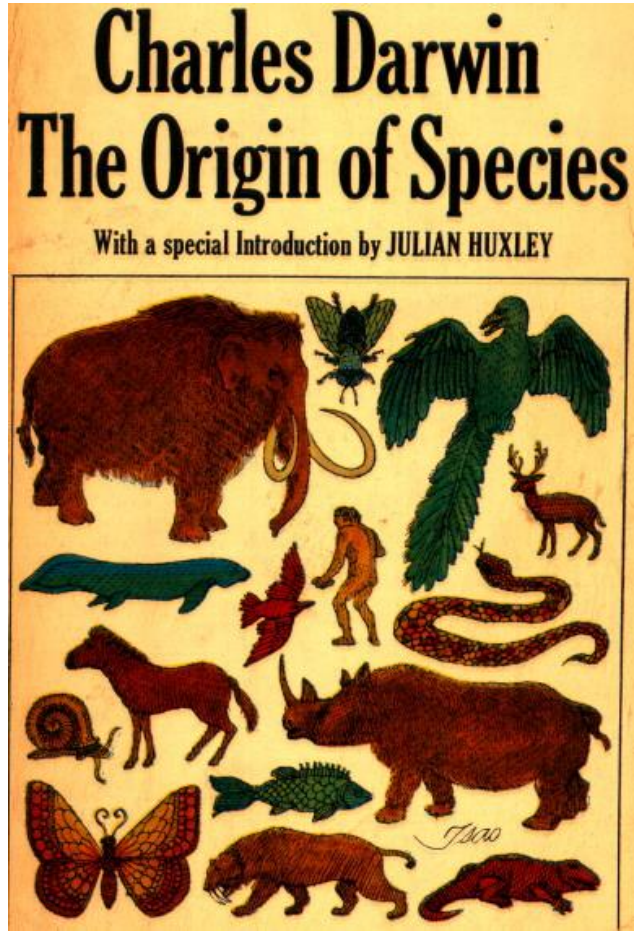
- Genetic algorithms are a technique to solve problems by using concept of life-evolution applying to find an optimum solution.
- GA are a member in Evolutionary Computation.
- GA are based on Charles Darwin's theory, 1859.

ไม่มีไฟฟ้า



Evolution is caused by natural selection.

วิวัฒนาการเกิดจากการคัดเลือกโดยธรรมชาติ



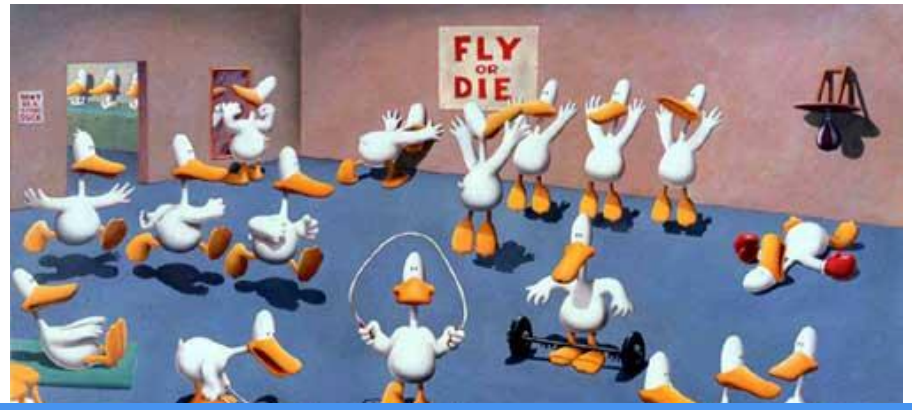
- Ref: https://www.youtube.com/watch?v=hOidIF0_FBc

Introduction to Genetic algorithm

- The origin of species:

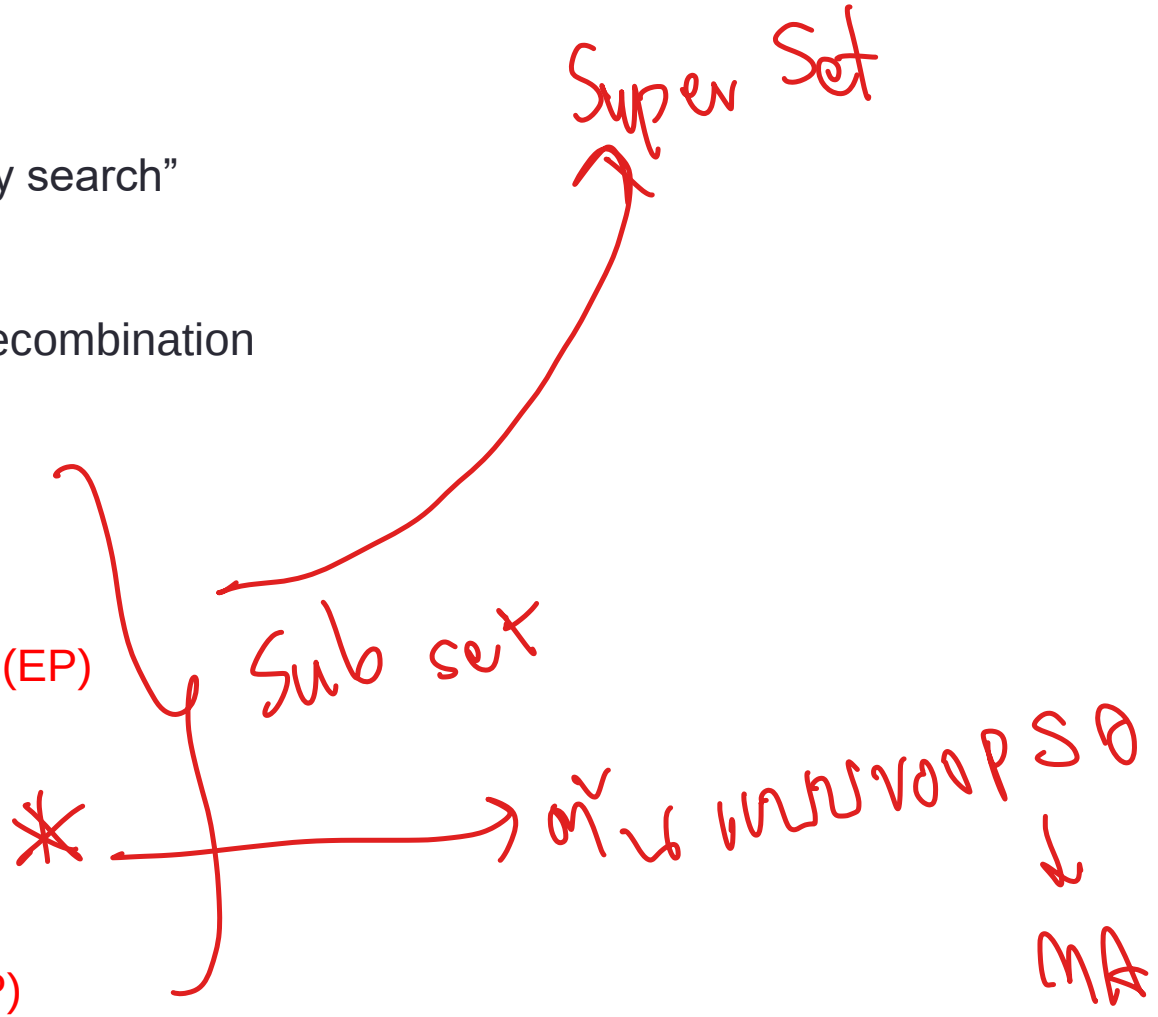
  “Preservation of favorable variations and rejection of unfavorable variations.”

- There are more individuals born than can survive by self-adapt to a different environment.
- Environment changing -> Adaptation -> Survival



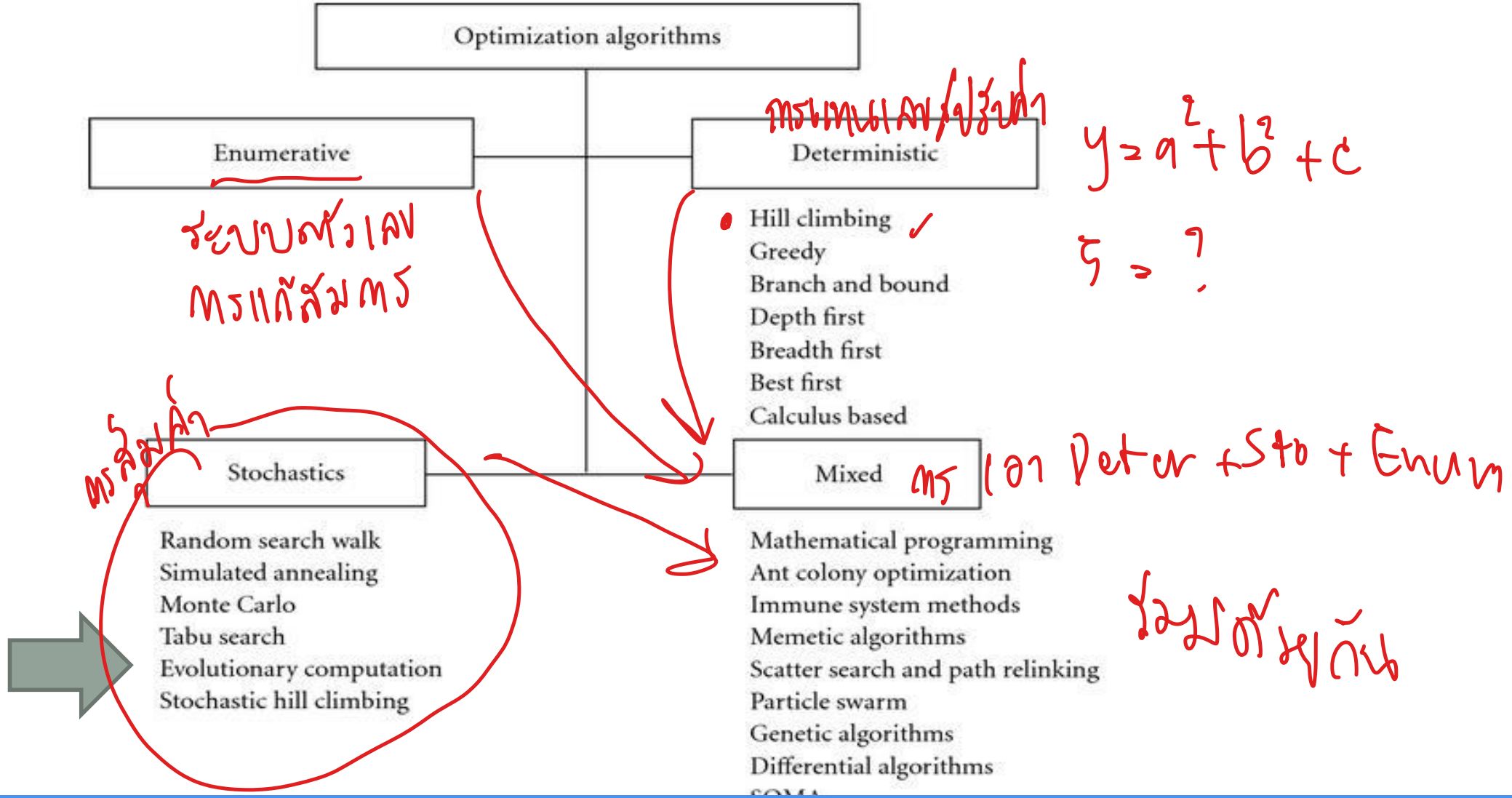
History of Evolutionary Computations

- 1948, Turing:
  proposes “Genetically or evolutionary search”
- 1962, Bremermann
  optimization through evolution and recombination
- 1964, Rechenberg
  introduces evolution strategies (ES)
- 1965, L. Fogel, Owens and Walsh
  introduce evolutionary programming (EP)
- 1975, John Holland
  introduces genetic algorithms (GA)
- 1992, Koza
  introduces genetic programming (GP)



Optimization Algorithms

หาผลลัพธ์ที่ดีที่สุด

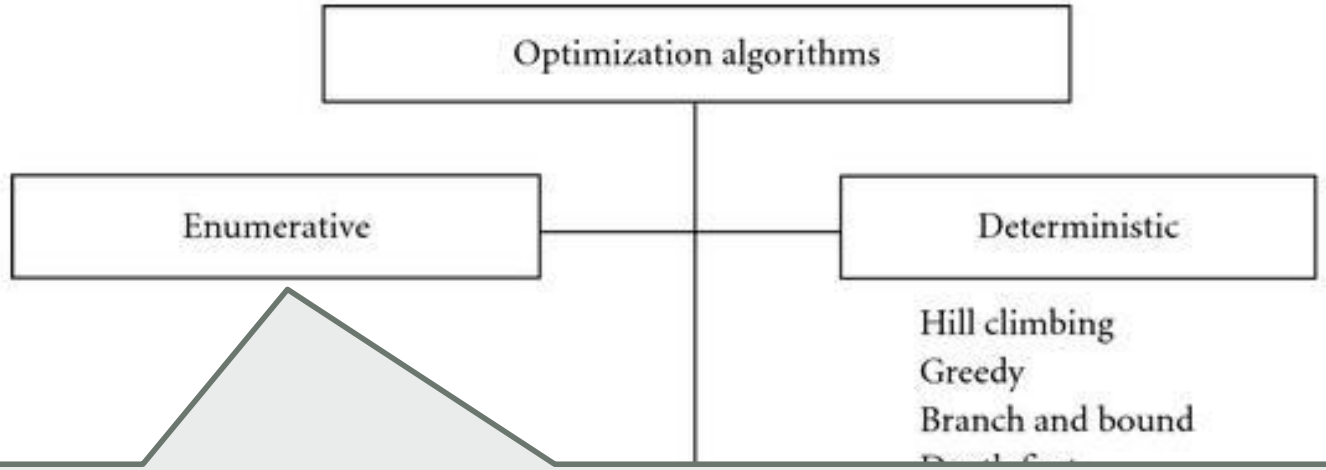


ระบบสุ่มเลข

$y = a^2 + b^2 + c$

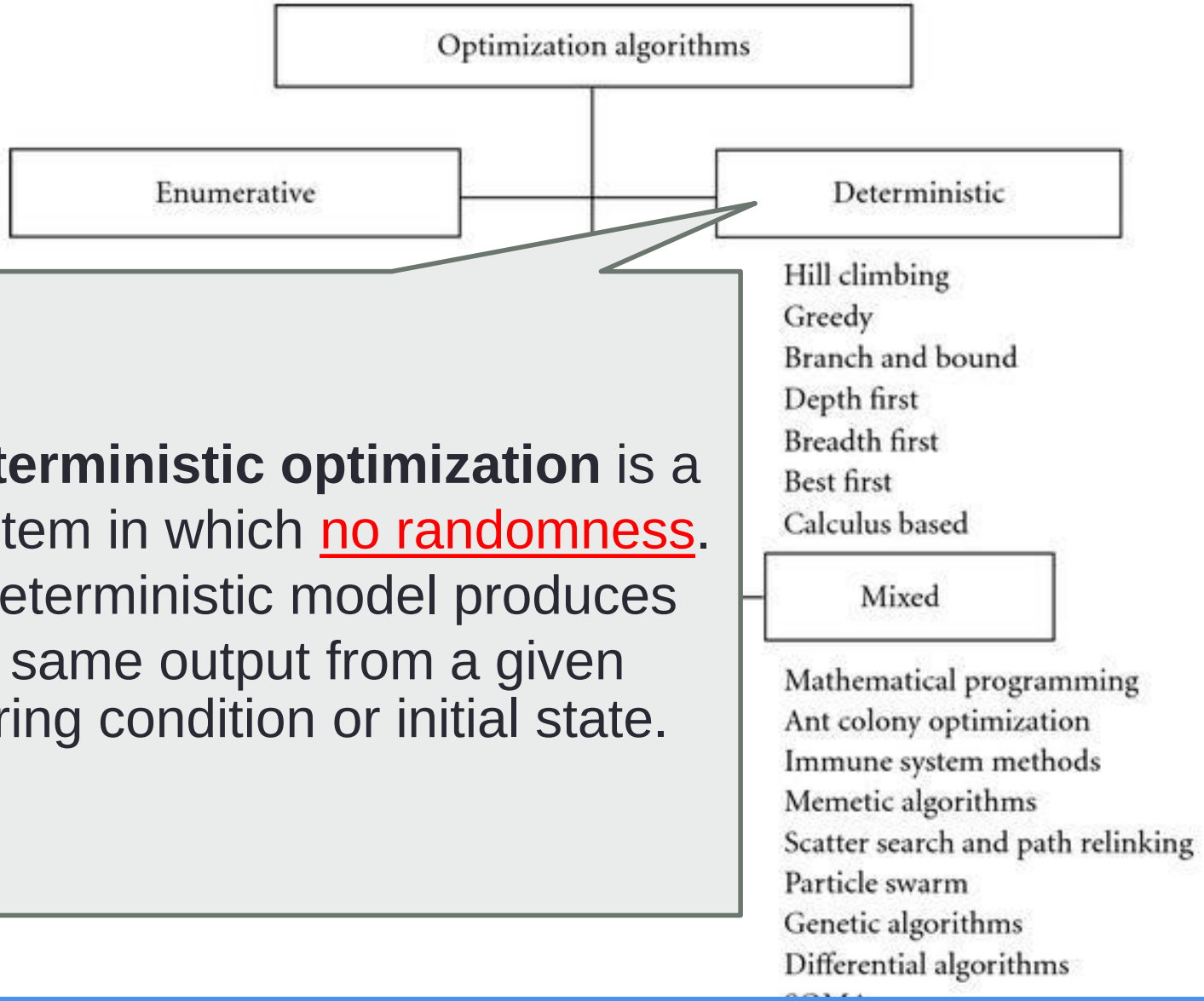
$5 = ?$

Optimization Algorithm



Enumeration:
The collection of items is a complete, ordered listing of all of the items in that collection. The term is commonly used in mathematics and theoretical computer science to refer to a listing of all the elements of a set.

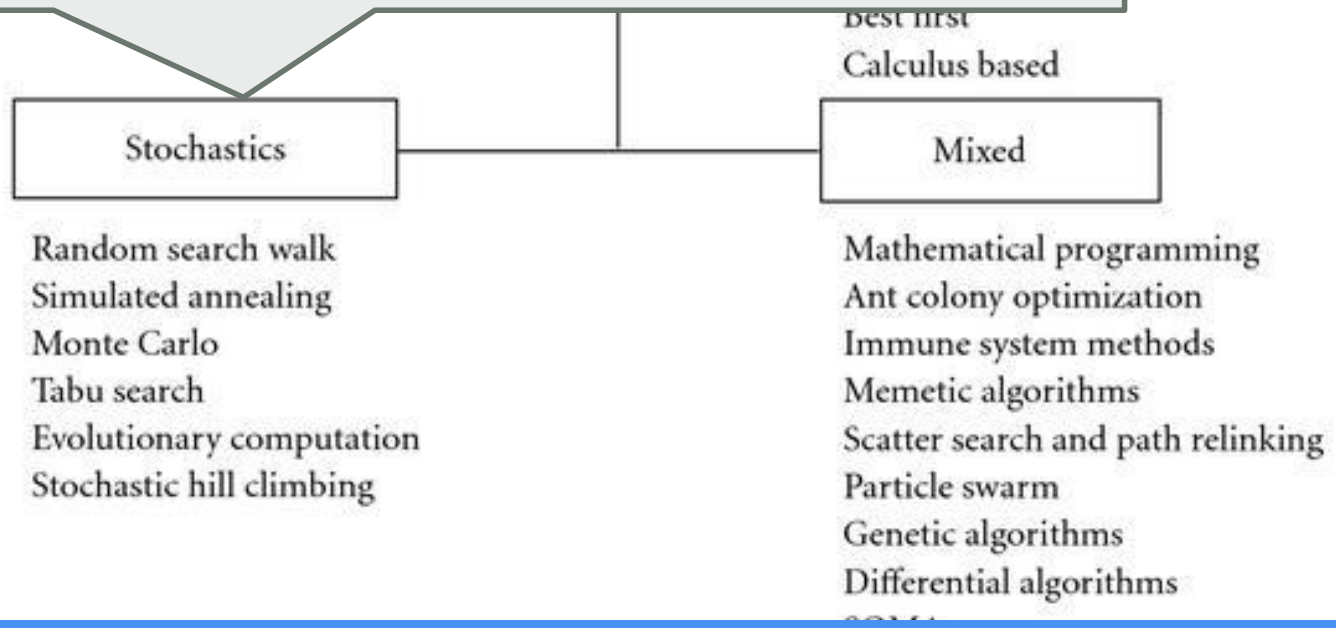
Understand Optimization Algorithm



Deterministic optimization is a system in which no randomness. A deterministic model produces the same output from a given staring condition or initial state.

Understand Optimization Algorithm

Stochastic optimization are optimization methods that generate and use random variables.

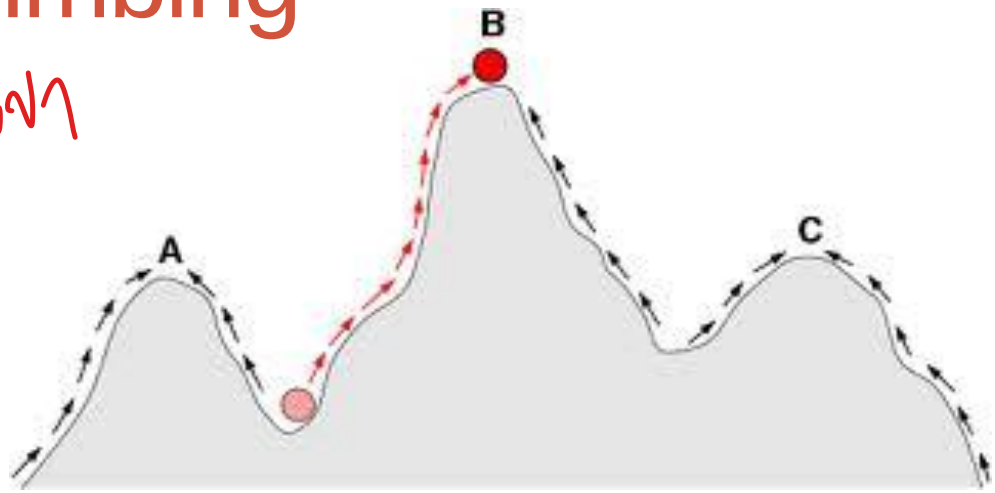


Hill Climbing

วิธีการหาผลลัพธ์ที่ดี

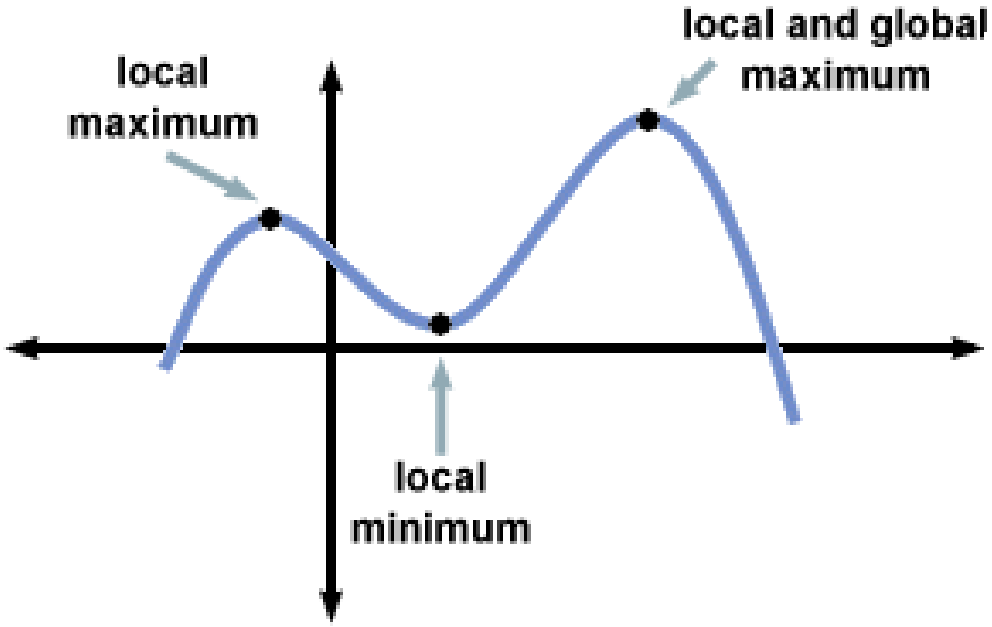
But ไม่ดีที่สุด

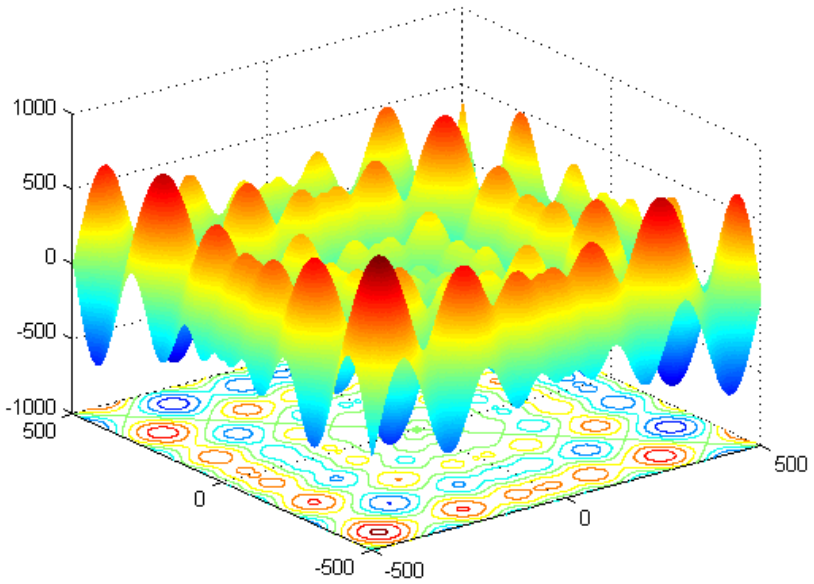
หรือแย่



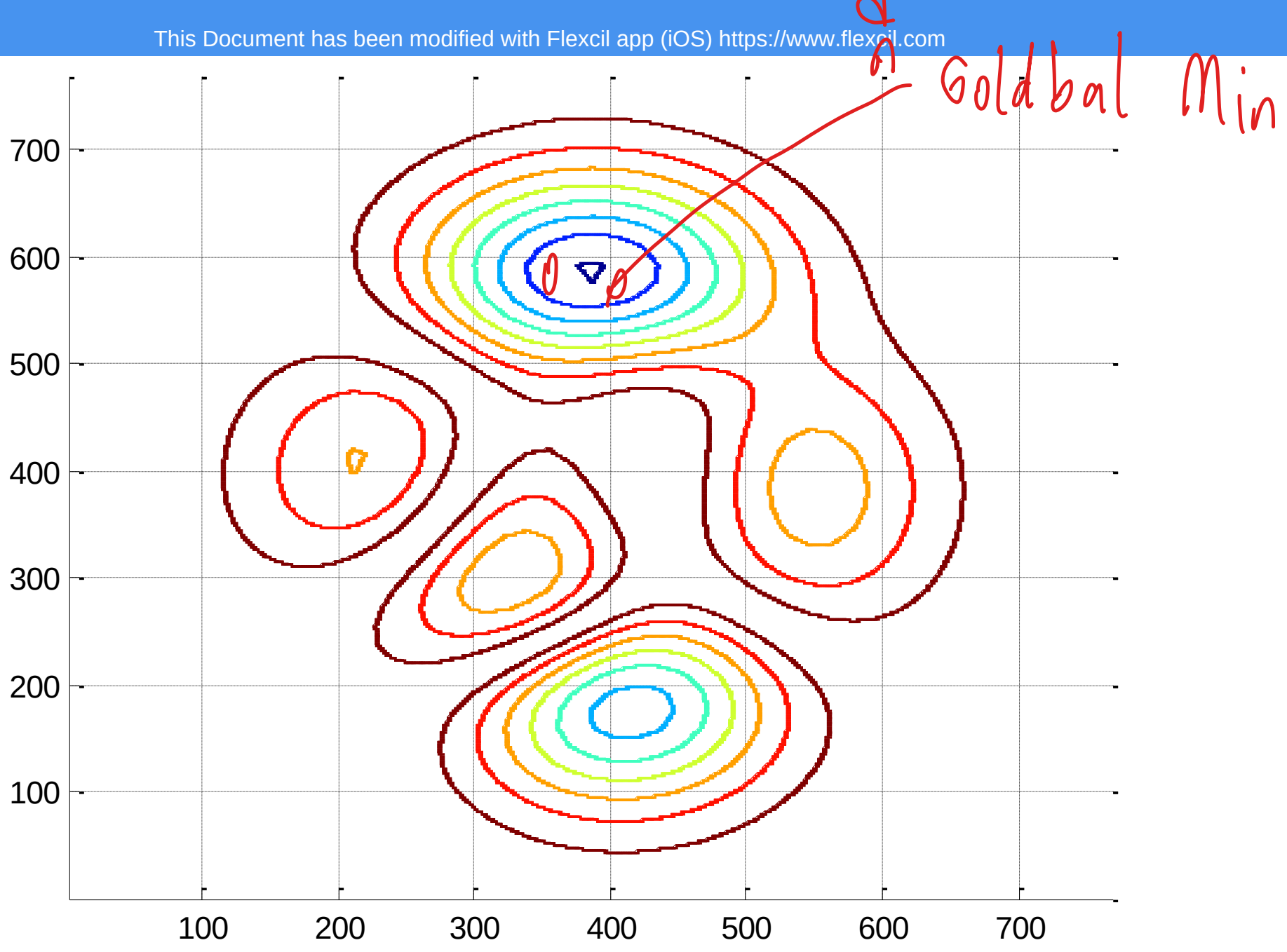
- The strategy of the Hill Climbing is iterating the process of select a neighbor for a candidate solution and only accept it if it results in an improvement. The strategy was proposed to address the limitations of deterministic hill climbing techniques that were likely to **get stuck in local optima** due to their greedy acceptance of neighboring moves.

- Local minima / Local maxima (Local solution)
- Global minimum / Global maximum (Optimum solution)

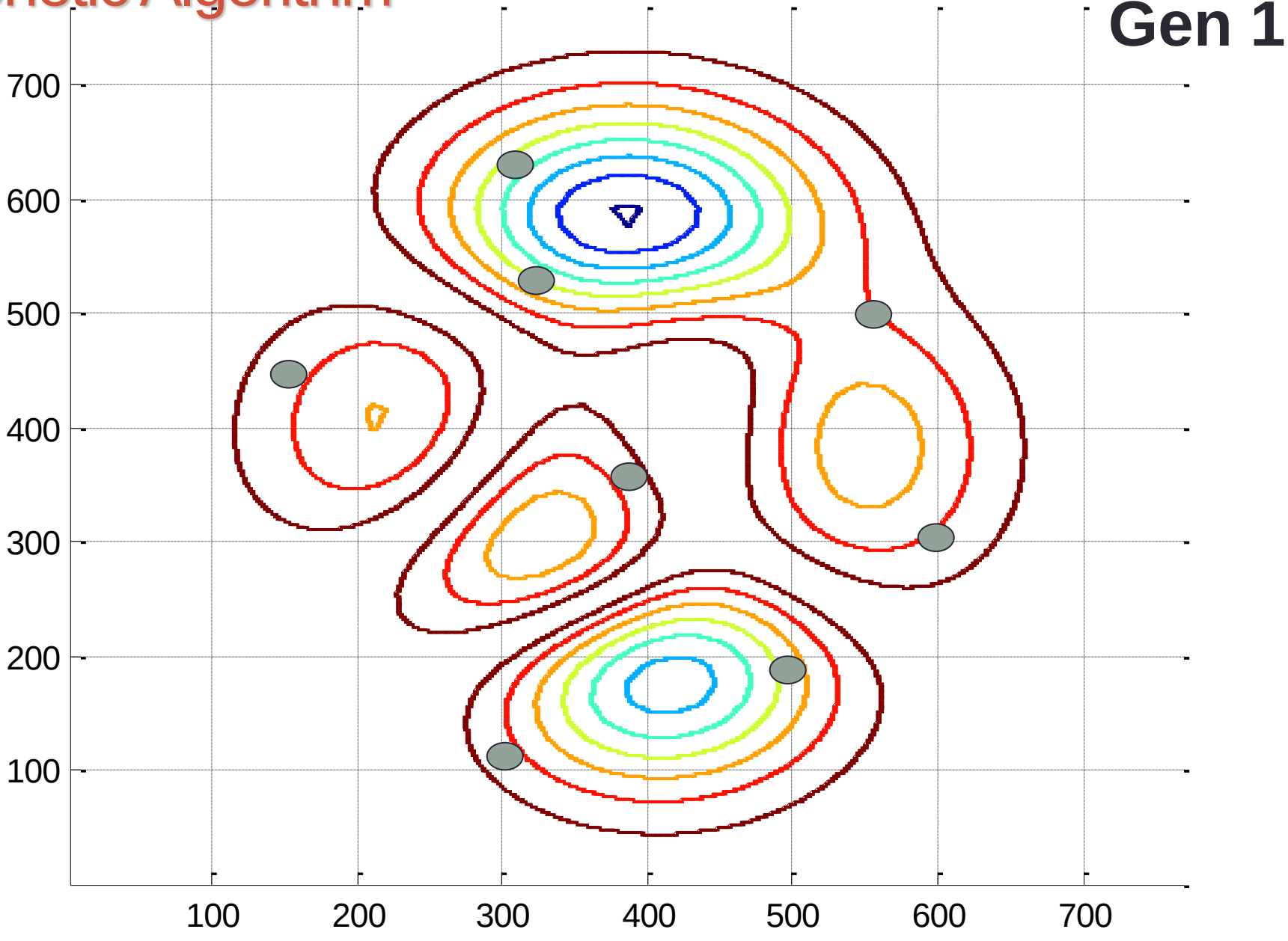




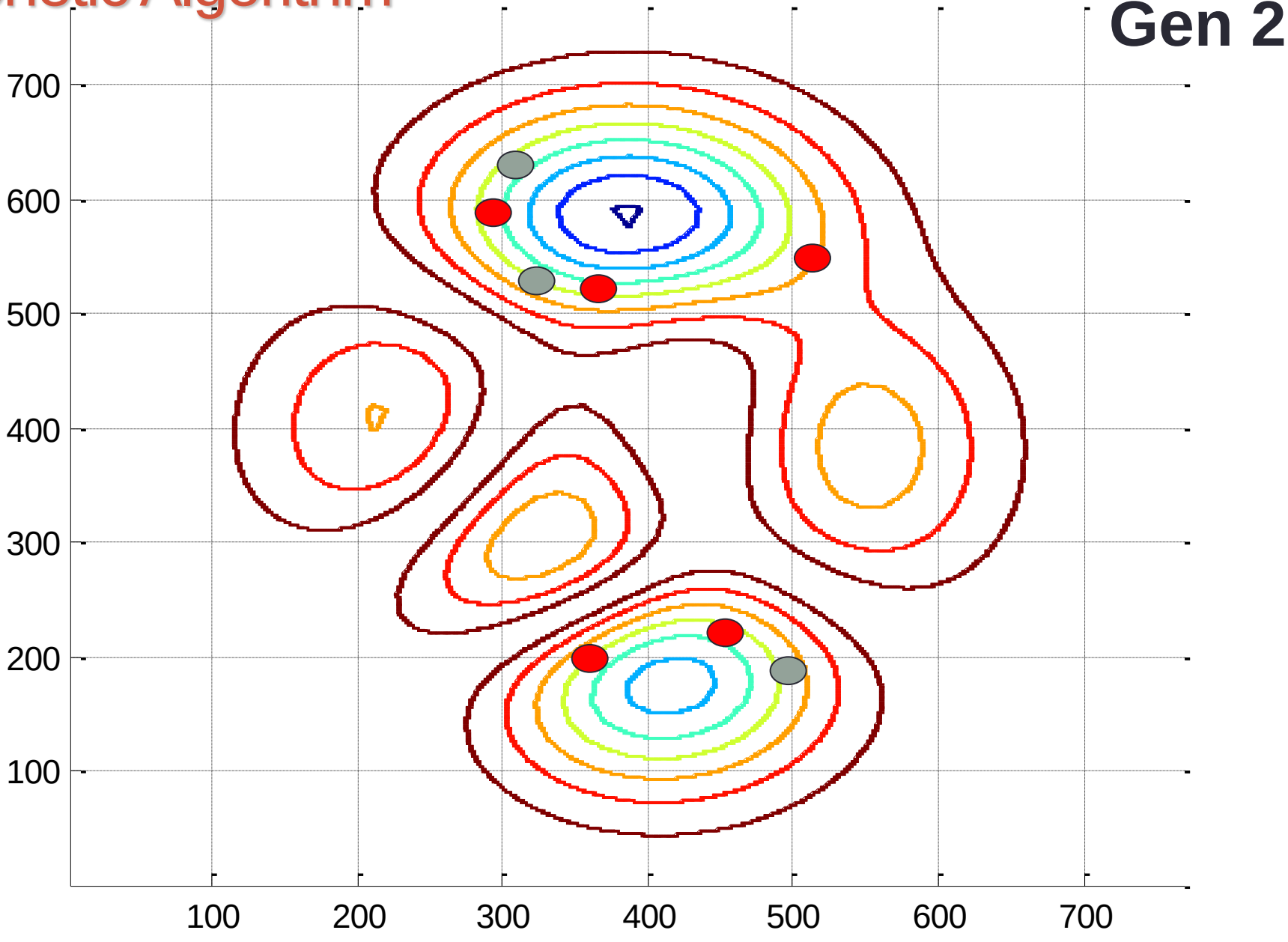
EVOLUTIONARY COMPUTATION CONCEPT



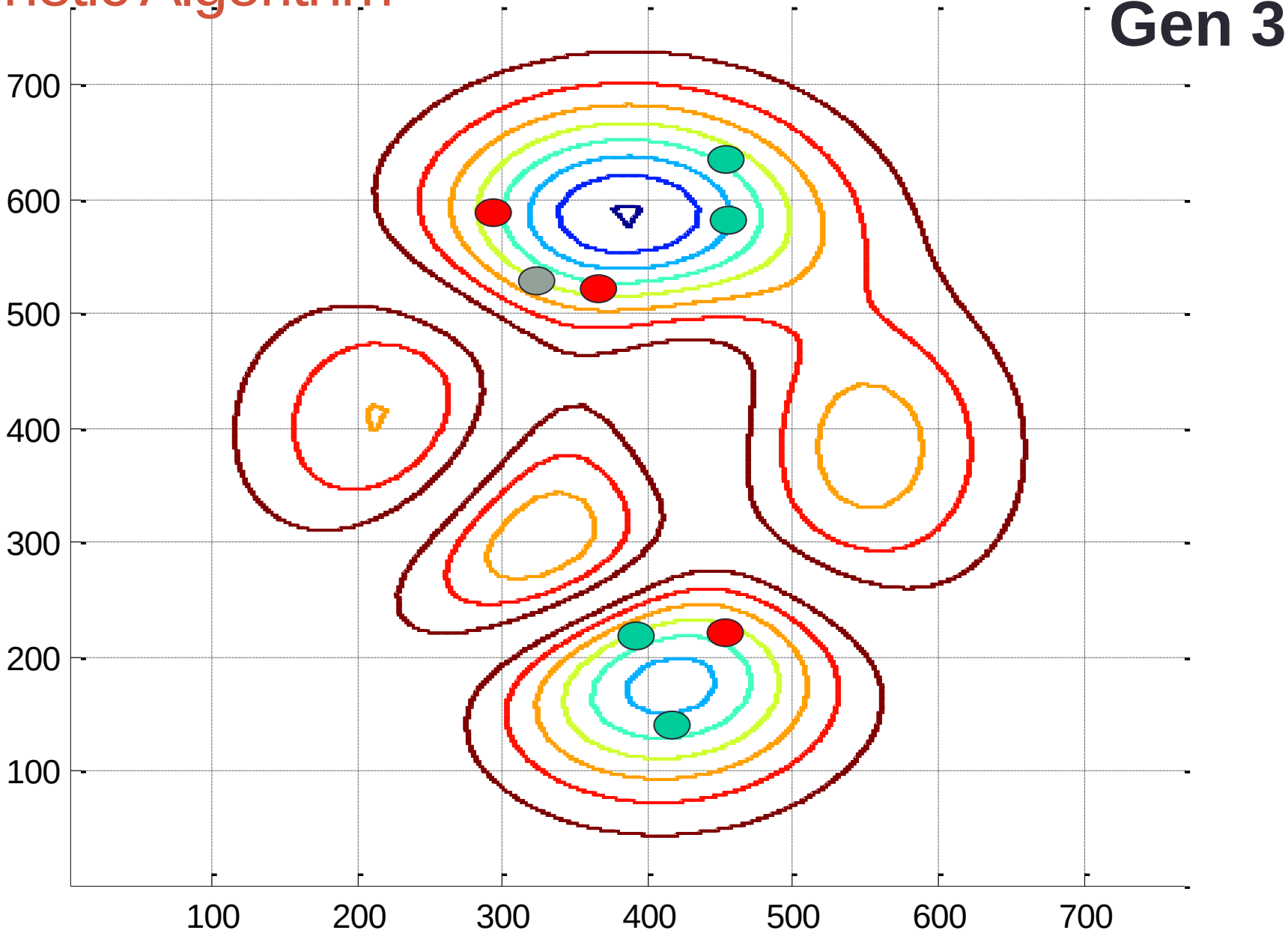
Genetic Algorithm



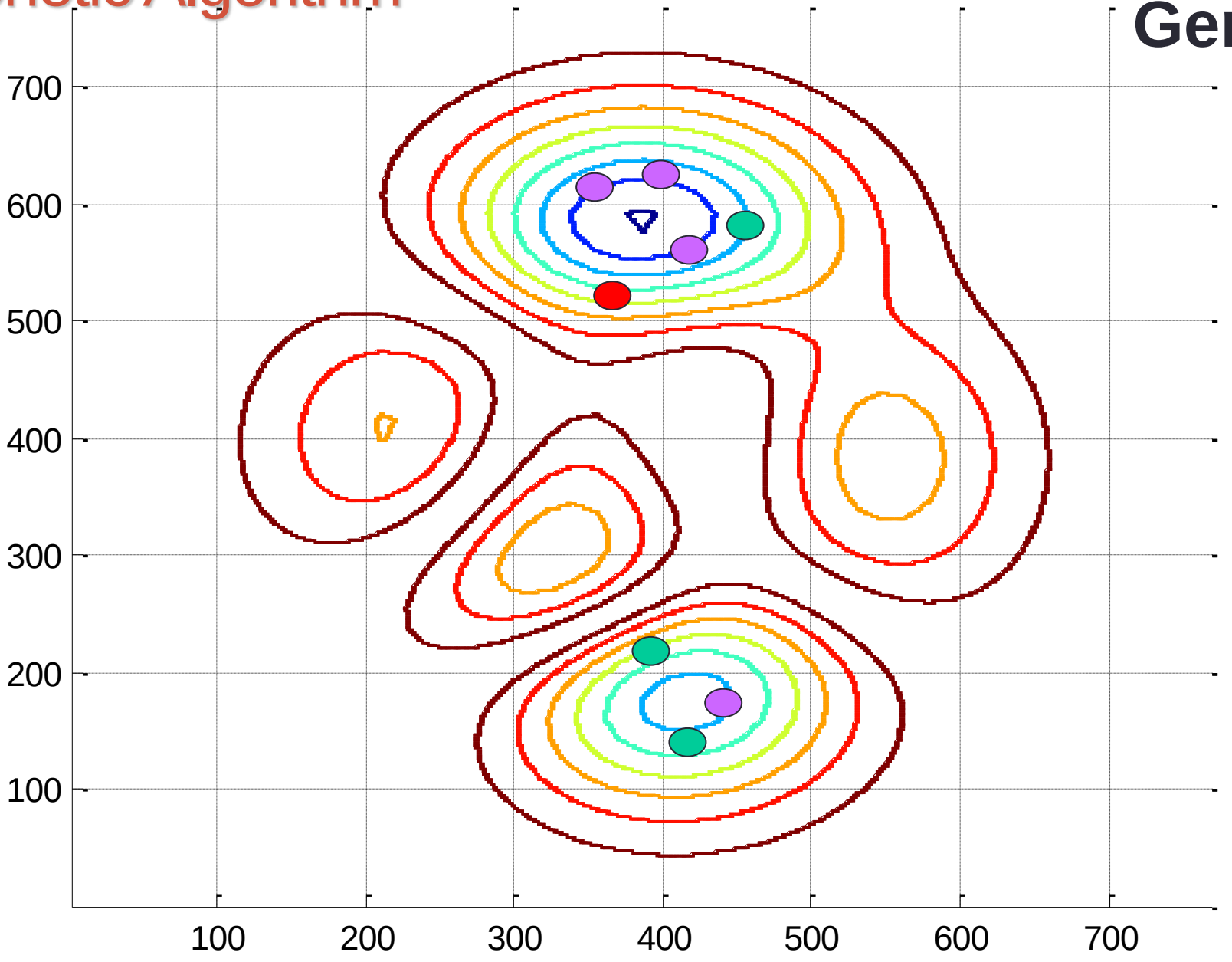
Genetic Algorithm



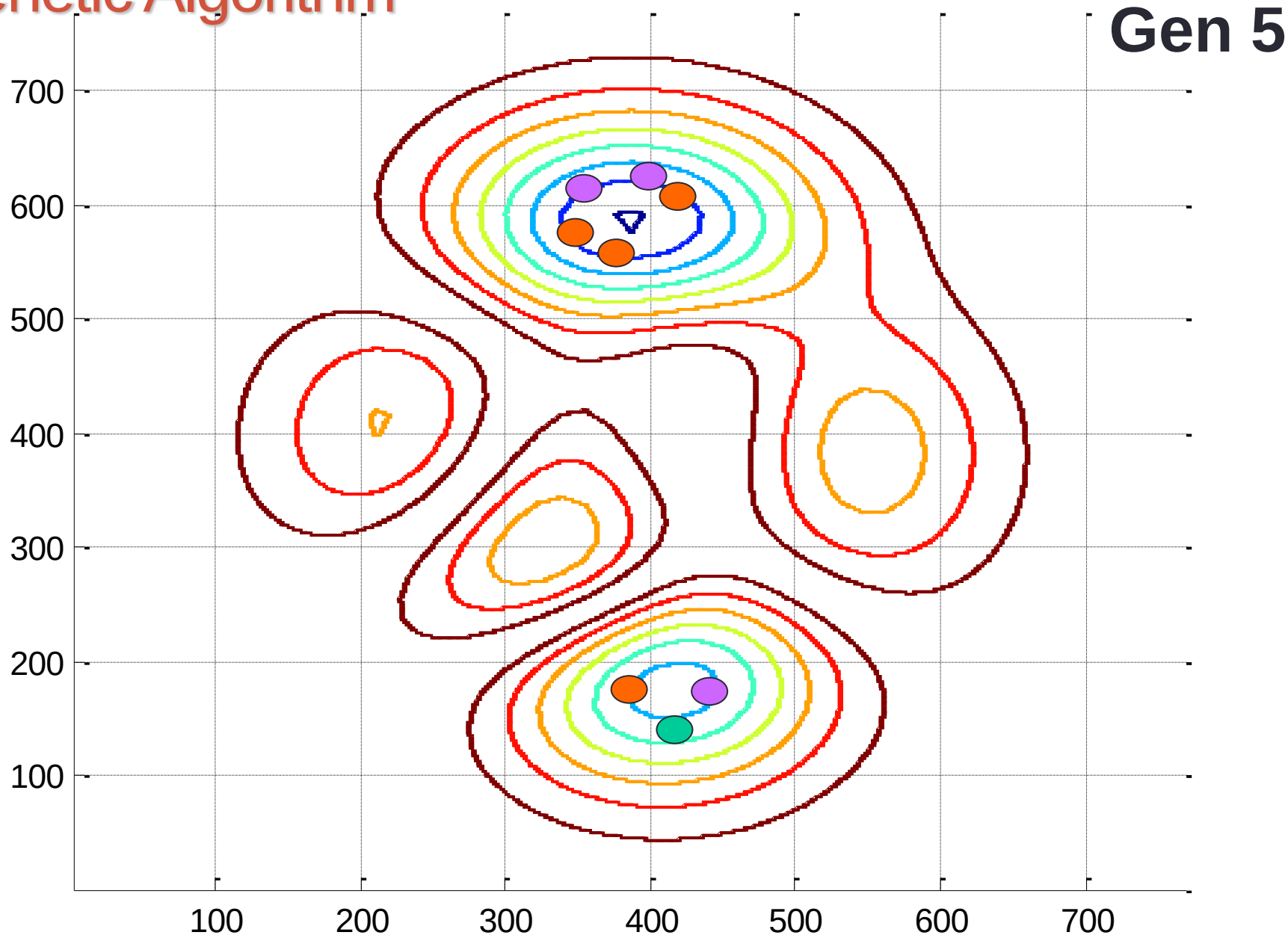
Genetic Algorithm



Genetic Algorithm



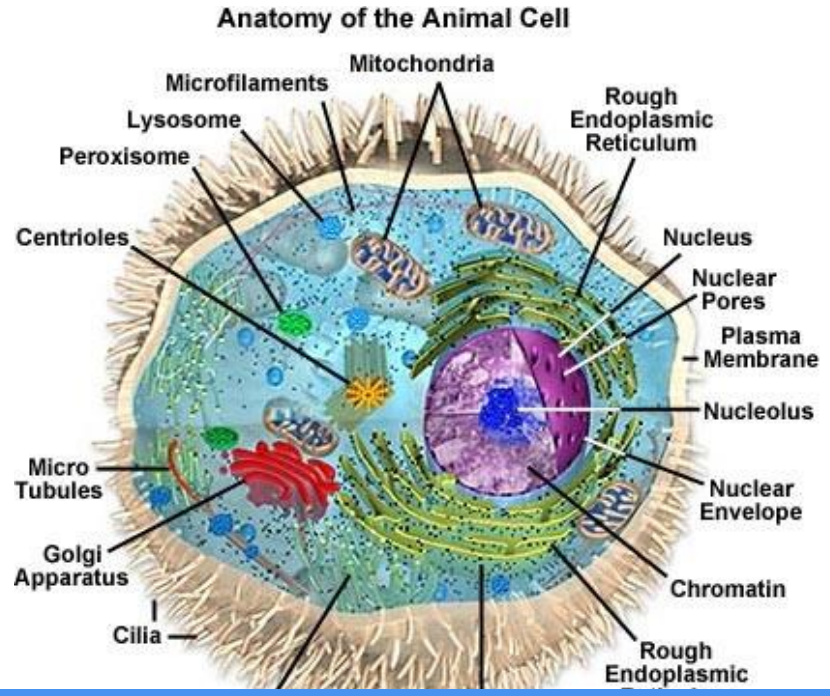
Genetic Algorithm

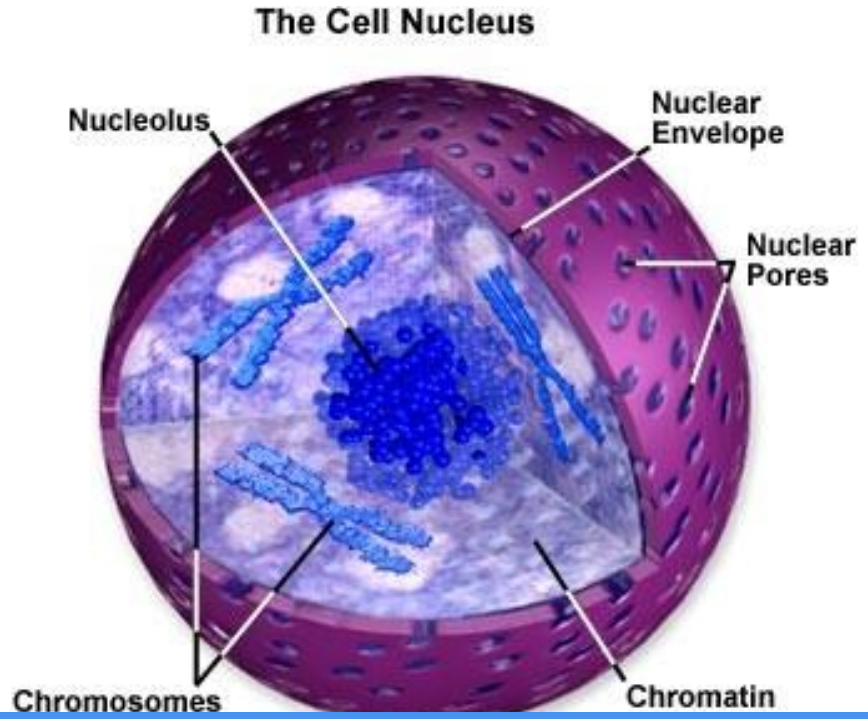


GENETIC ALGORITHM THEORY

Biological background (1)

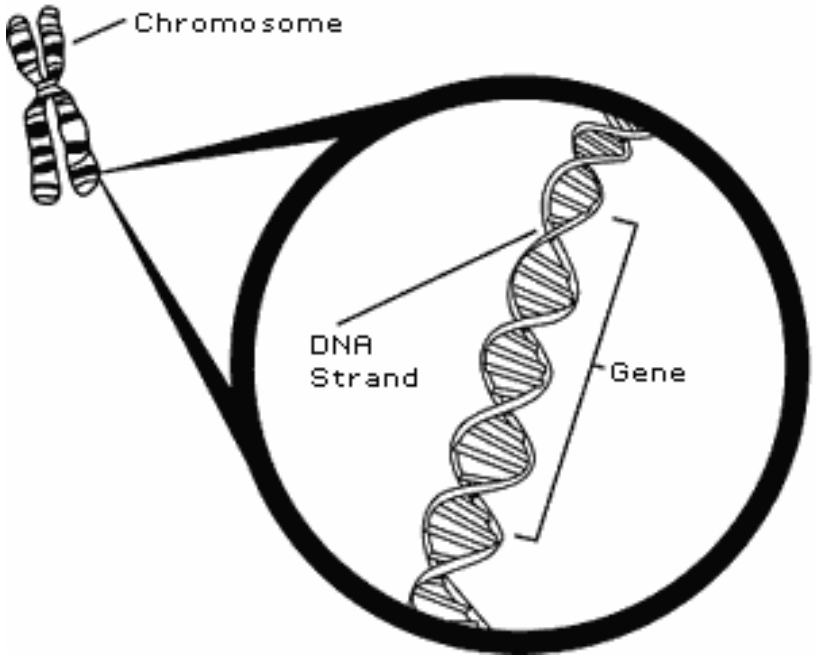
- Every animal cell is a complex of many small "factories" working together
- The center of this all is the cell nucleus
- The nucleus contains the genetic information

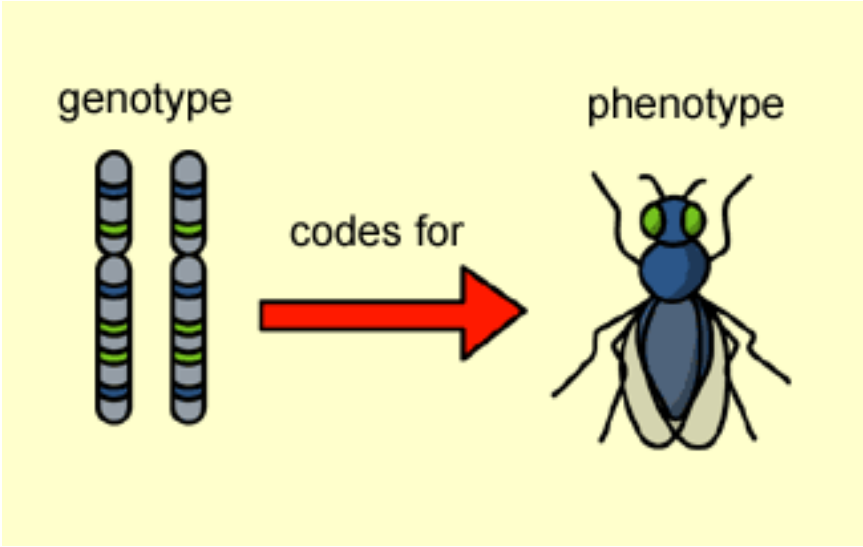




Biological background (2)

- Genetic information is stored in the chromosomes
- Each chromosome is build of DNA
- Chromosomes in humans form pairs
- There are 23 pairs
- The chromosome is divided in parts: genes
- Genes code for properties
- The posibilities of the genes for one property is called: allele
- Every gene has an unique position on the chromosome: locus





- DNA → Chromosome → Gene

ภายใน

- Genotype, DNA inside is the observable in level of gene or chromosome inside individual.

ภายนอก

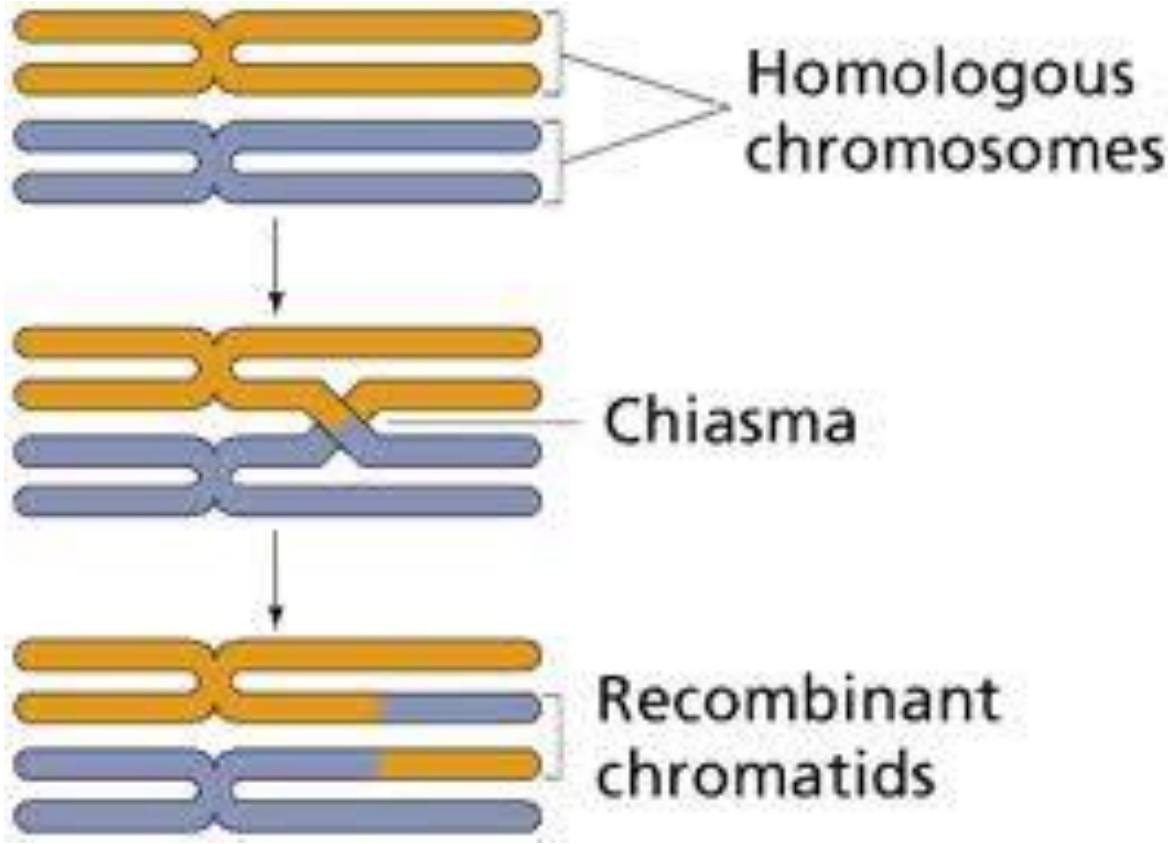
- Phenotype is the observable characteristics of an individual.

Sexual Recombination

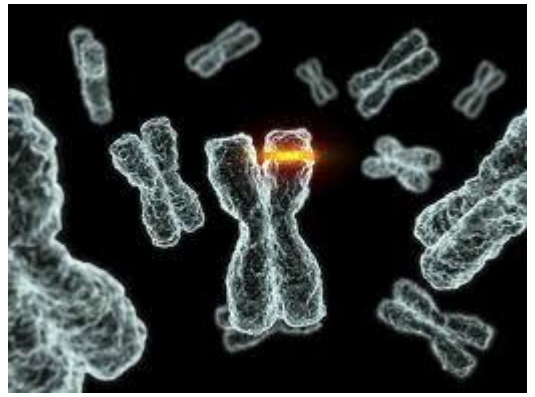
- GA called Crossover

การสลับขา แลกยีน เช่น

- Crossover is the primary operator to introduce diversity in population



Asexual



- GA called Mutation การกลายพันธุ์
- Crossover is the second operator to introduce diversity in population

การรวม + เปลี่ยนพันธุ์ ๆ ดูว่าอันไหนดีกว่ากัน

Genetic Selection techniques

เลือกคู่แต่งงาน

↳ การเลือก

- Uniform selection
- Roulette wheel selection
- Tournament selection
- Random selection

Genetic Selection techniques

- Uniform selection

เหมือนพ่อแม่เลือกให้

- Roule l selection

มีแบบที่กำหนดมาแล้วว่าต้องเลือกตัวไหน

Uniform selection: is the selection by using same pattern.

Example

sequence 1 2 3 4 5 6 7 8 9 10

select every counting 3

1 4 7 10

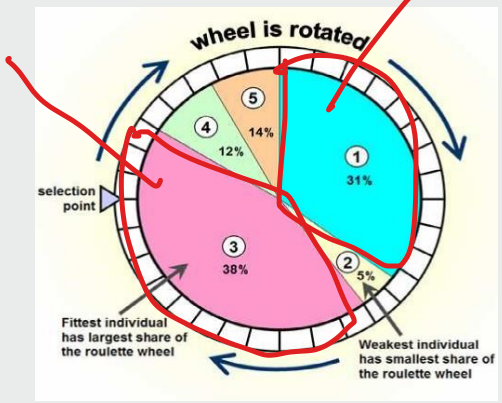
การเลือกแบบมี pattern

Genetic Selection techniques

- Uniform selection
- Roulette wheel selection

คนที่มีพื้นที่มาก เลือกมาก
น้อย

Roulette selection: is random selection applying concept of roulette board.



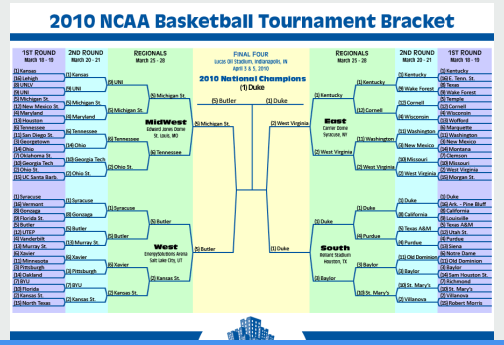

American style double zero roulette wheel.
Photo courtesy www.abbiati.it

การใช้วงล้อ Roulette

Genetic Selection techniques

- Uniform selection
- Roulette wheel selection
- Tournament selection ทรงแข่งขัน
- Rando[illegible]

Tournament selection: is the selection by comparing each pair of population. In each pair, the selection may select a population by randomness.



Genetic Selection techniques

- Uniform selection
- Roulette wheel selection
- Tournament selection
- Random selection สุ่ม

Random selection is the randomness selecting.

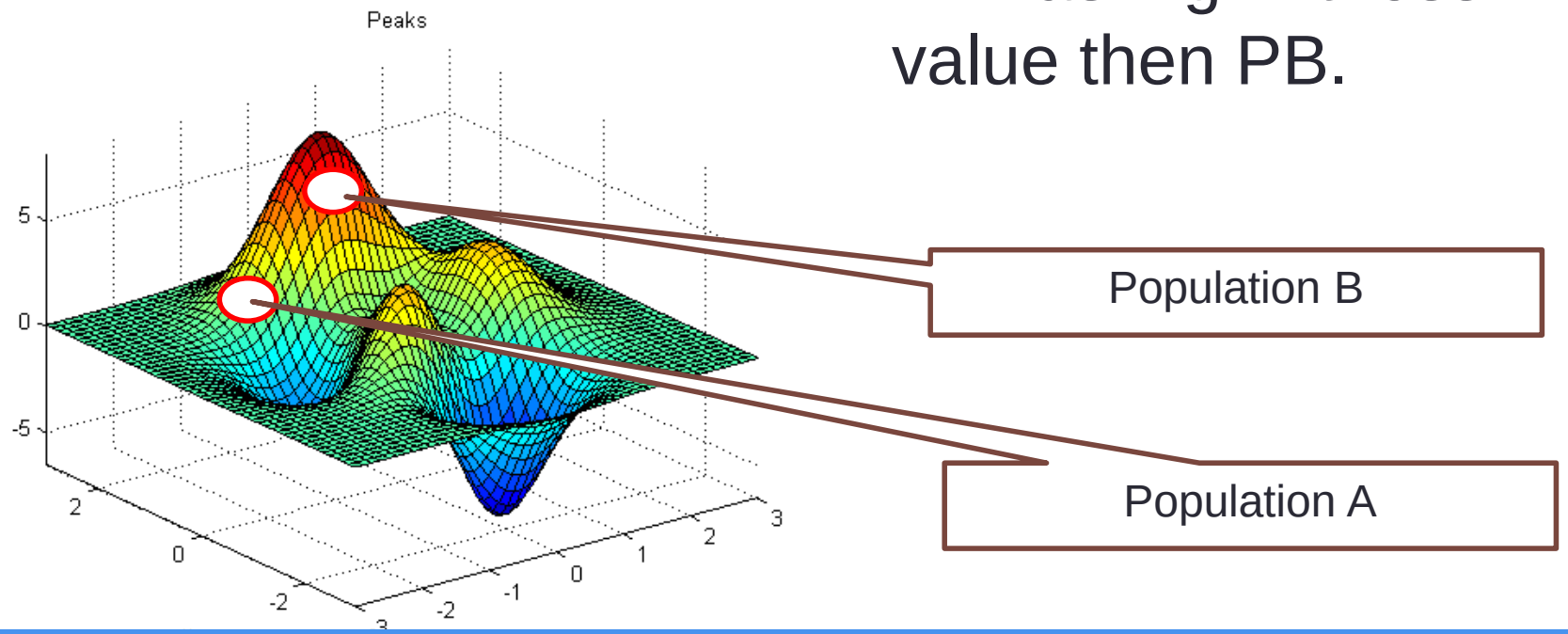
Example:	C-code	srand(), rand()
	Excel	random
	R	sample, runif()

Fitness value

การคำนวณ function ที่ใช้วัด Objective function

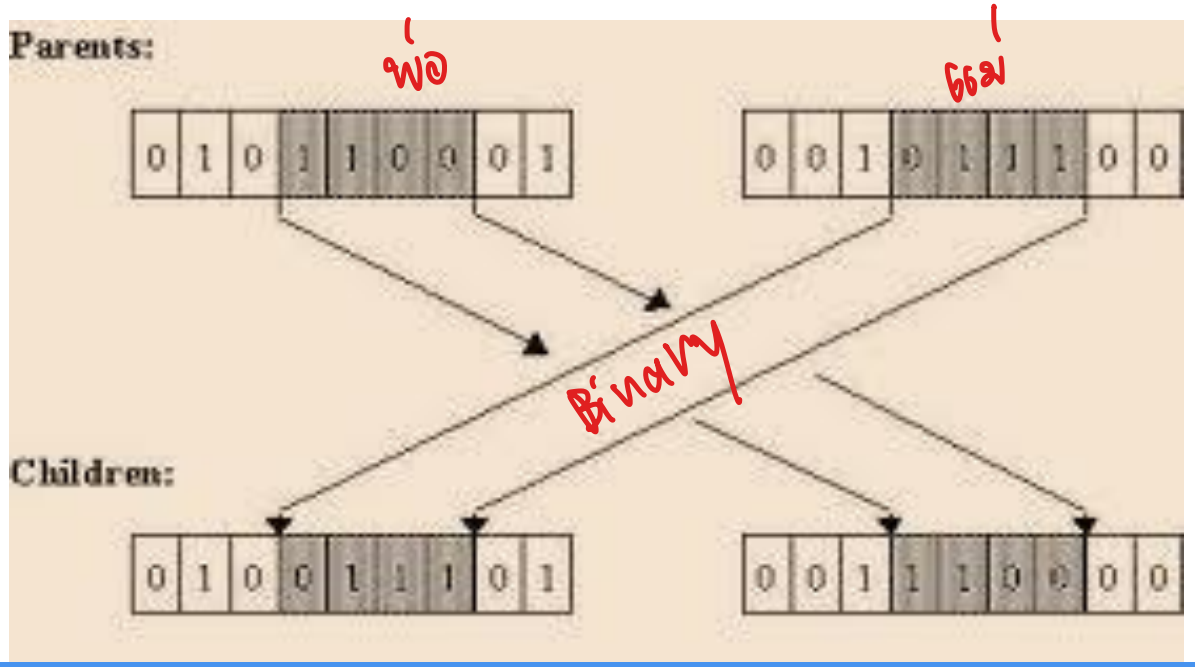
- Fitness value is a value of quality to measure result of solving a problem.

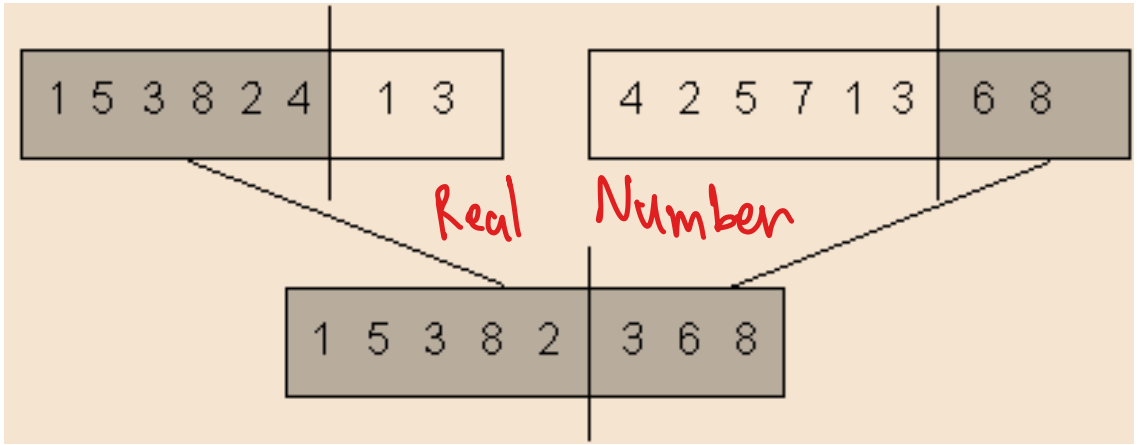
PA has high fitness value then PB.



Chromosome Representation

- Problem representation can be encoding two techniques.
  - Binary chromosome
  - Real number chromosome



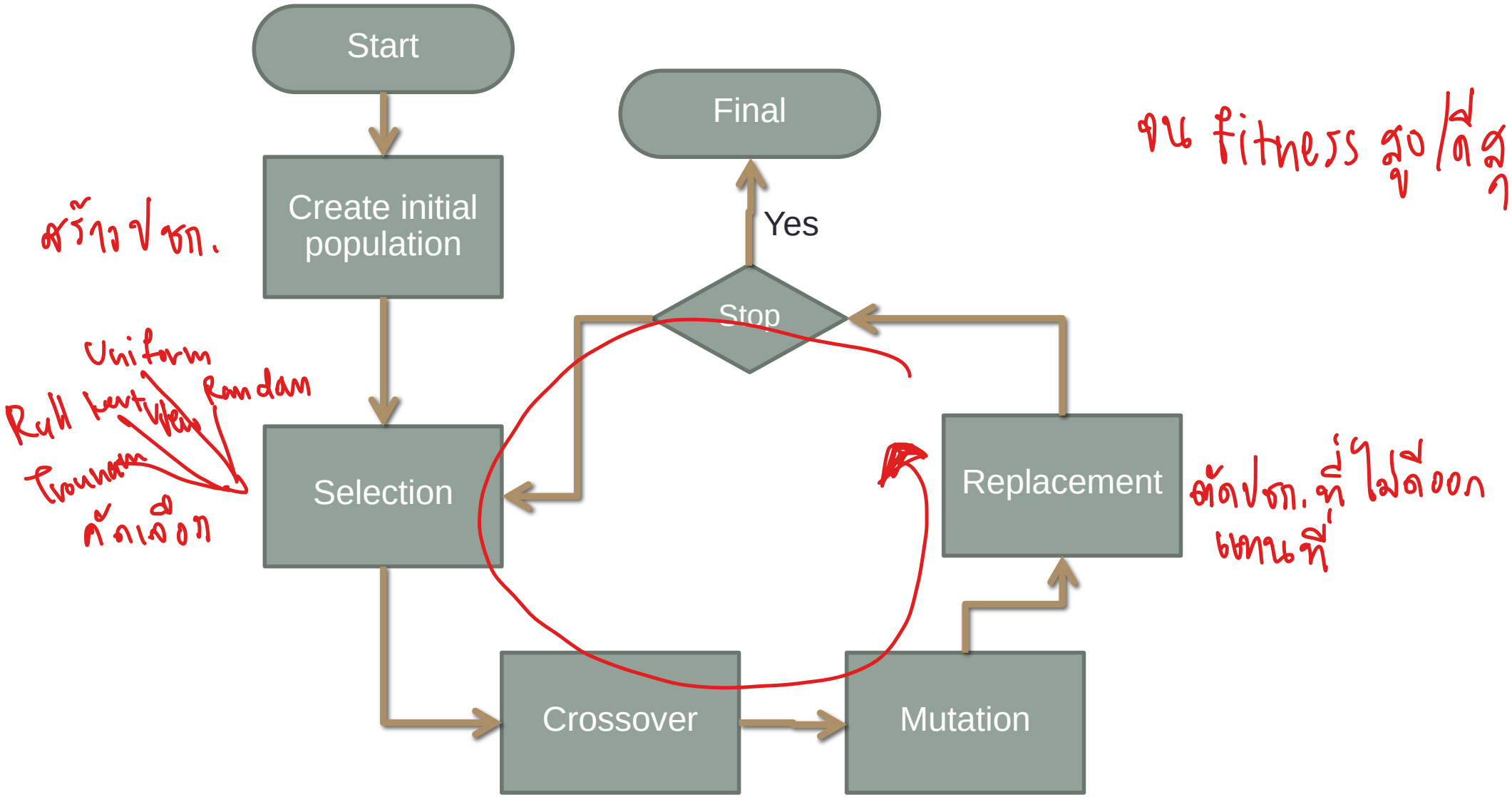


Glossary--GA

- Evolutionary Computation
  - GA
  - EC
  - EA
  - GP
  - EP
- Selection
  - Roulette wheel
  - Uniform
  - Tournament
  - Random
- Crossover (Recombination)
- Mutation
- Chromosome
  - Gene
- Phenotype ←Genotype
- Fitness
- Populations
  - A population = Individual
  - Old population (Parent)
  - New population (Offspring)
- Encode Chromosome
  - Binary
  - Real
- Generation

PROCESS OF GA

GA process



Represent problem

- Problem
  - $z = x^2+y^2$ → Objective function
- What is the value x, y at z =0?
- Step1
  - Encode problem (Example use Binary Chromosome)

เปลี่ยนเป็น Binary

Example

X = 9 = 1001

Y = 3 = 0011

ถ้าข้อมูลยาวใช้เป็น Real Number ได้

Then, a population uses two gene 1001|0011

Create population

Chromosome

$z = x^2 + y^2$

- Parents

P1: 1001|0011 x=9 y=3 z=? 81+9=90

P2: 0101|0110 x=5 y=6 z=? 25+36=61

P3: 0011|0010 x=3 y=2 z=? 9+4=13 2→น้อย/สูง

P4: 1001|0011 x=9 y=3 z=? 81+9=90

P5: 1101|0010 x=13 y=2 z=? 169+4=173

Create population

- Parents

P1: 1001|0011 x=9 y=3 z=90

P2: 0101|0110 x=5 y=6 z=61

P3: 0011|0010 x=3 y=2 z=13

P4: 1001|0011 x=9 y=3 z=90

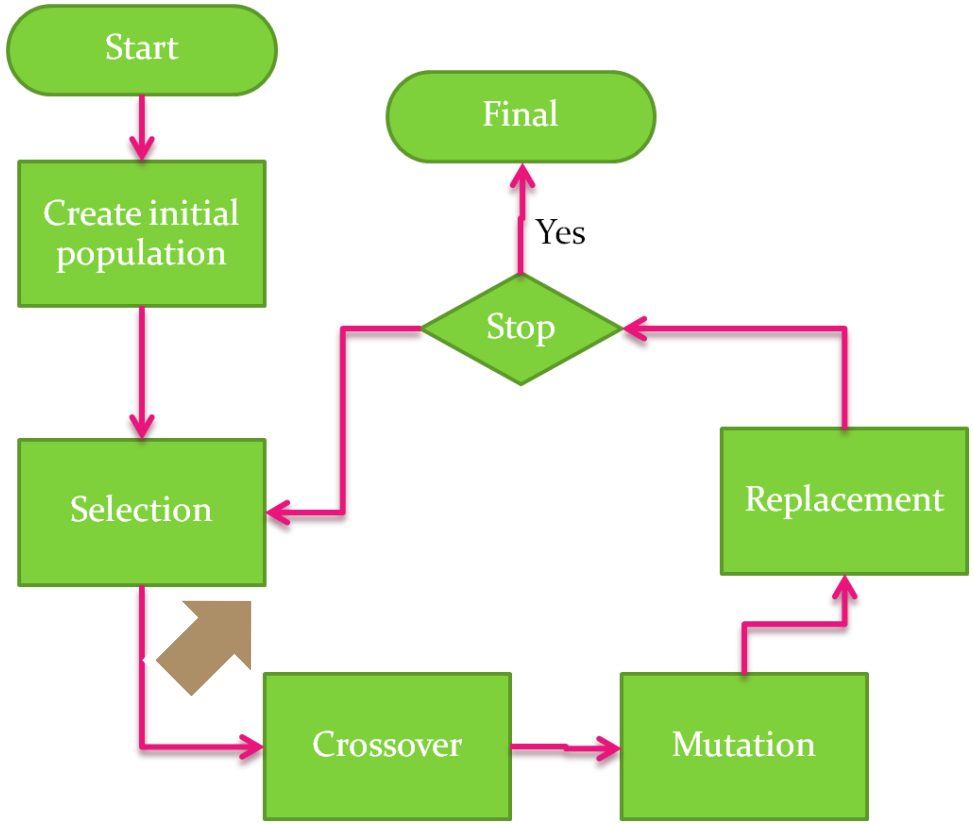
P5: 1101|0010 x=13 y=2 z=139

ดีสุด fitness น้อยสุด

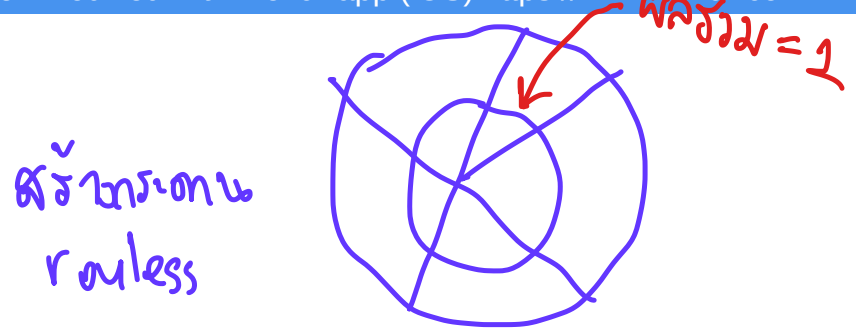
Which population has the best fitness value?

Selection

- Use roulette wheel generating 5 offspring



Selection



- Parents

P1: 1001|0011	x=9 y=3	z=90
P2: 0101|0110	x=5 y=6	z=61
P3: 0011|0010	x=3 y=2	z=13
P4: 1001|0011	x=9 y=3	z=90
P5: 1101|0010	x=13 y=2	z=139

Sum z = 393

Which population has best fitness?

Selection

- Parents

P1: 1001|0011	x=9 y=3	z=90**/393**
P2: 0101|0110	x=5 y=6	z=61**/393**
P3: 0011|0010	x=3 y=2	z=13**/393**
P4: 1001|0011	x=9 y=3	z=90**/393**
P5: 1101|0010	x=13 y=2	z=139**/393**

Sum z = 393

Which population has best fitness?

Selection

- Parents

P1: 1001|0011 x=9 y=3 z=0.229

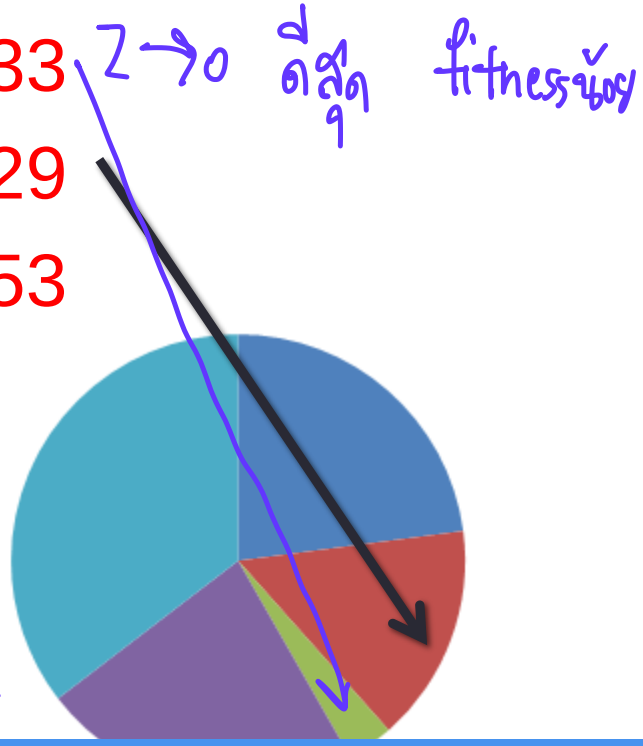
P2: 0101|0110 x=5 y=6 z=0.1552

P3: 0011|0010 x=3 y=2 z=0.033

P4: 1001|0011 x=9 y=3 z=0.229

P5: 1101|0010 x=13 y=2 z=0.353

Sum z = 393



แก้โจทย์ 1 ไปลบ ๆ Z=0.033 = 9.867

Selection

- Parents

P1: 1001|0011 x=9 y=3 z=0.229

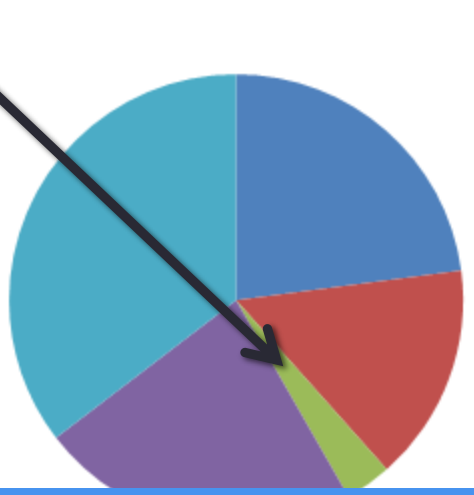
P2: 0101|0110 x=5 y=6 z=0.1552

P3: 0011|0010 x=3 y=2 z=0.033

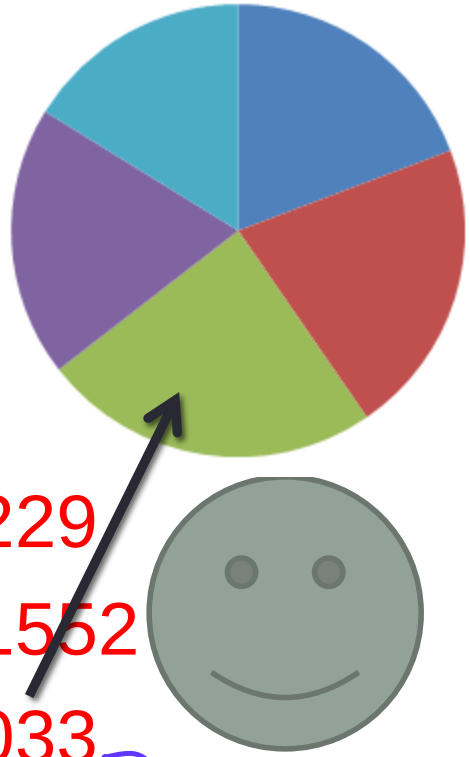
P4: 1001|0011 x=9 y=3 z=0.229

P5: 1101|0010 x=13 y=2 z=0.353

Sum z = 0.393



Selection



- Parents

P1: 1001|0011 x=9 y=3 z=1-0.229

P2: 0101|0110 x=5 y=6 z=1-0.1562

P3: 0011|0010 x=3 y=2 z=1-0.033

P4: 1001|0011 x=9 y=3 z=1-0.229

P5: 1101|0010 x=13 y=2 z=1-0.353

Sum z = 393

Selection

- Offspring

P6: 0011|0010 x=3 y=2 z=1-0.033 ซ้ำกัน

P7: 0101|0110 x=5 y=6 z=1-0.1552

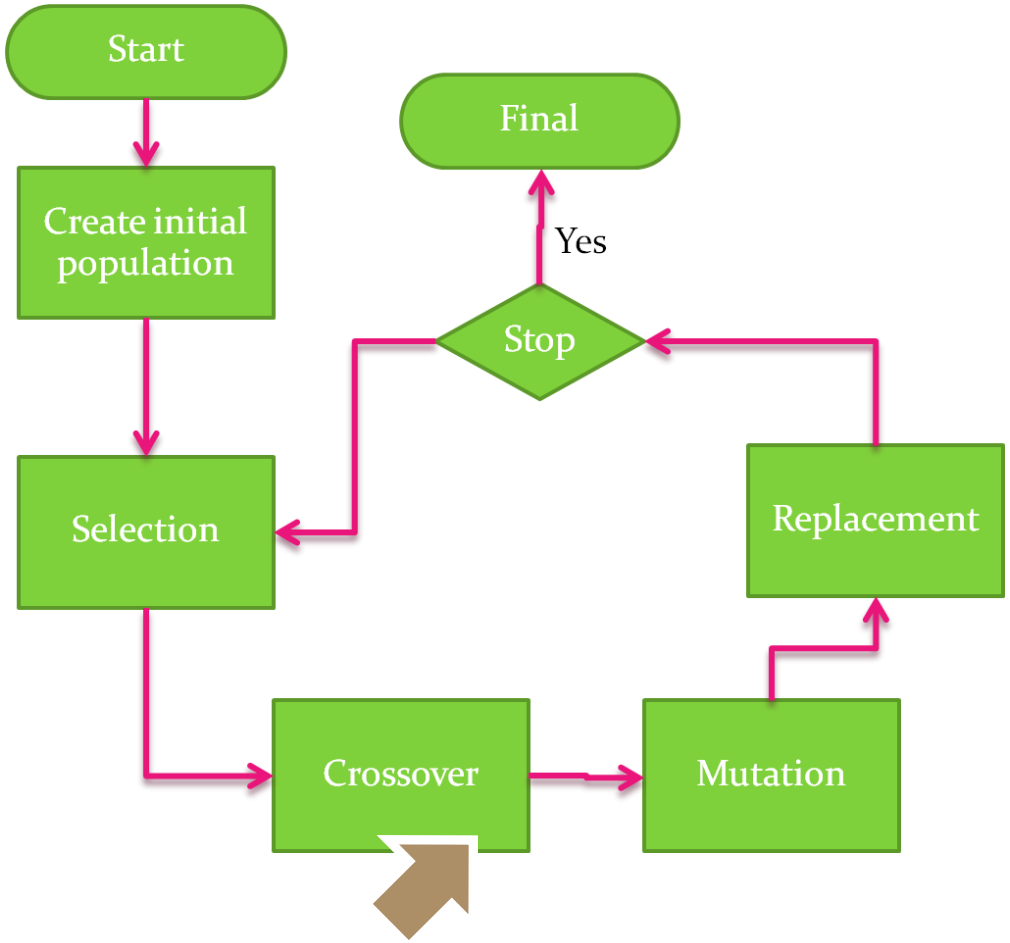
P8: 0011|0010 x=3 y=2 z=1-0.033 0.967 ✓ fitness ยิ่งมาก

P9: 1001|0011 x=9 y=3 z=1-0.229

P10: 0101|0110 x=5 y=6 z=1-0.1552

Crossover

- Crossover techniques
  - One point
  - Two point
  - N-point



Crossover

สุ่มจับคู่

- Select P6 and P10 by random control with Crossover Probability

P6: 0011|0010 x=3 y=2

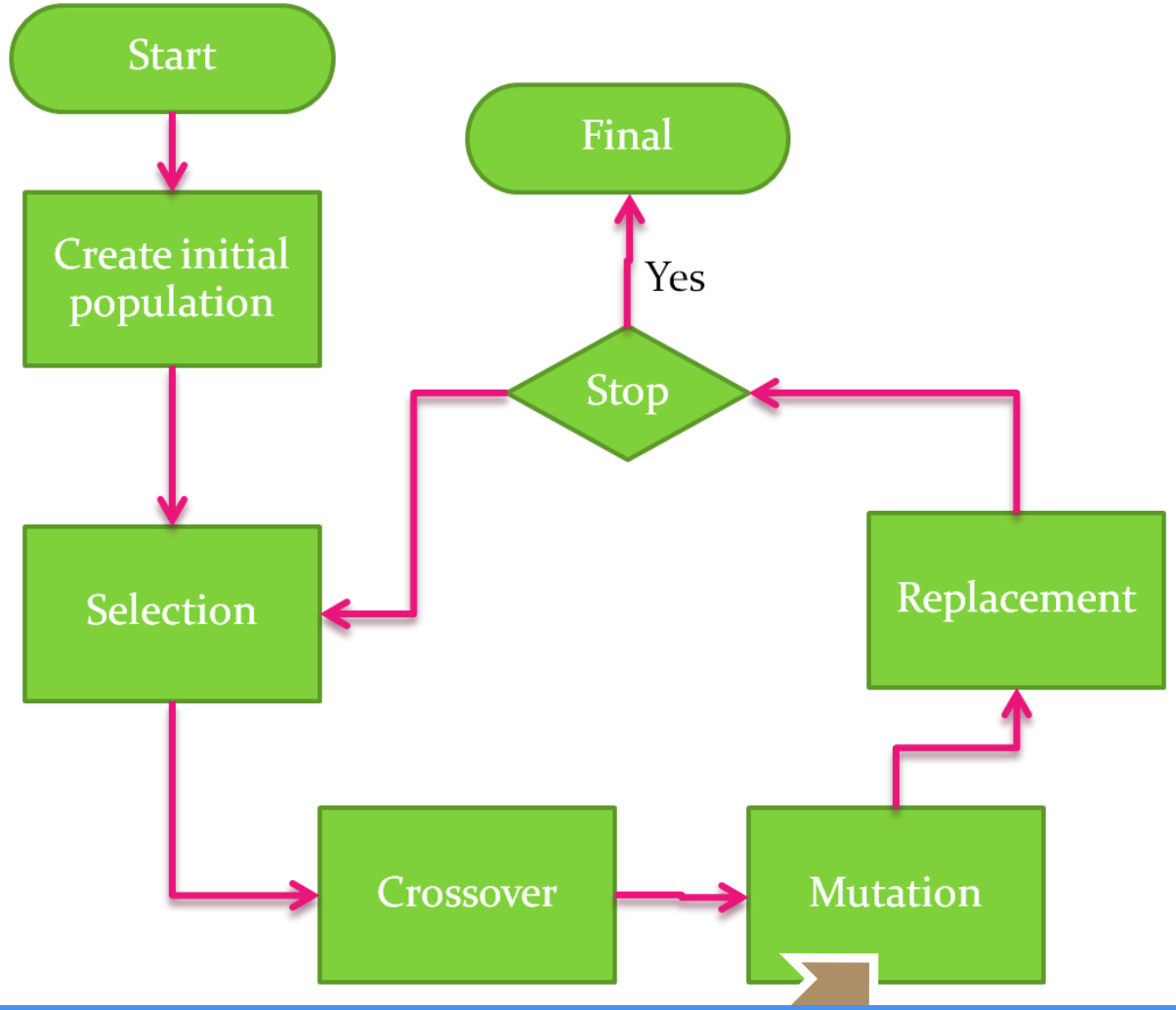
P10: 0101|0110 x=5 y=6

- Random select gene to swap

P6: 0111|0110 x=? y=? z=?

P10: 0001|0010 x=? y=? z=?

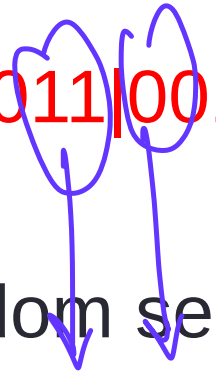
Mutation



Crossover Mutation เลือก คน โดย สุ่ม เปลี่ยน

- Select P6 by random control with mutation probability (pm)

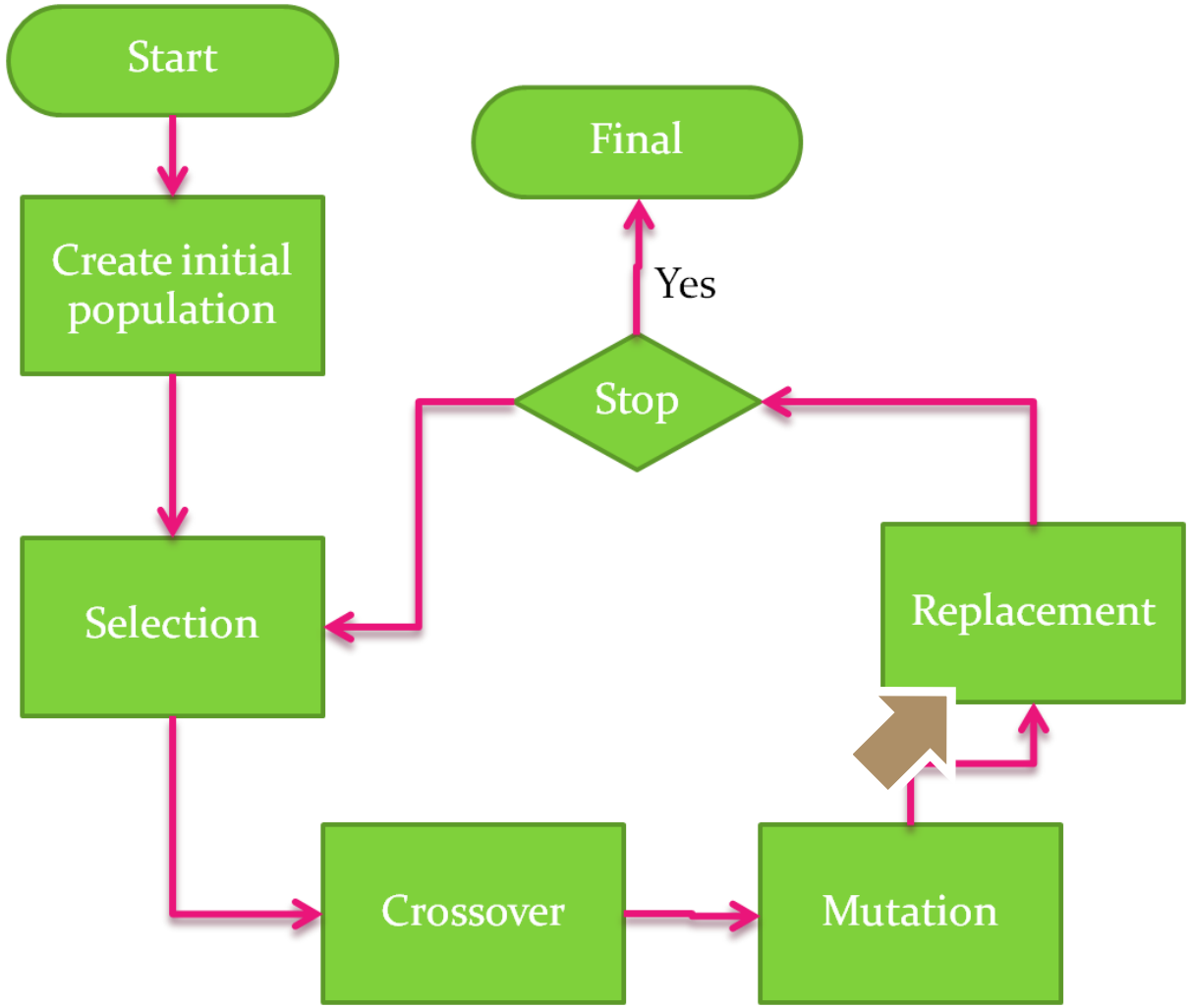
P6: 0011|0010 x=3 y=2



- Random select gene to give new value

P6: 0001|1010 x=? y=?

Replacement



Selection

- Offspring

P6: 0011|0010 x=3 y=2z=?

P7: 0111|0100 x=5 y=6z=?

P8: 1011|0010 x=3 y=2z=?

P9: 1001|0011 x=9 y=3z=?

P10: 0111|0110 x=5 y=6z=?

หา fitness ใหม่

- Parents

P1: 1001|0011 x=9 y=3

P2: 0101|0110 x=5 y=6

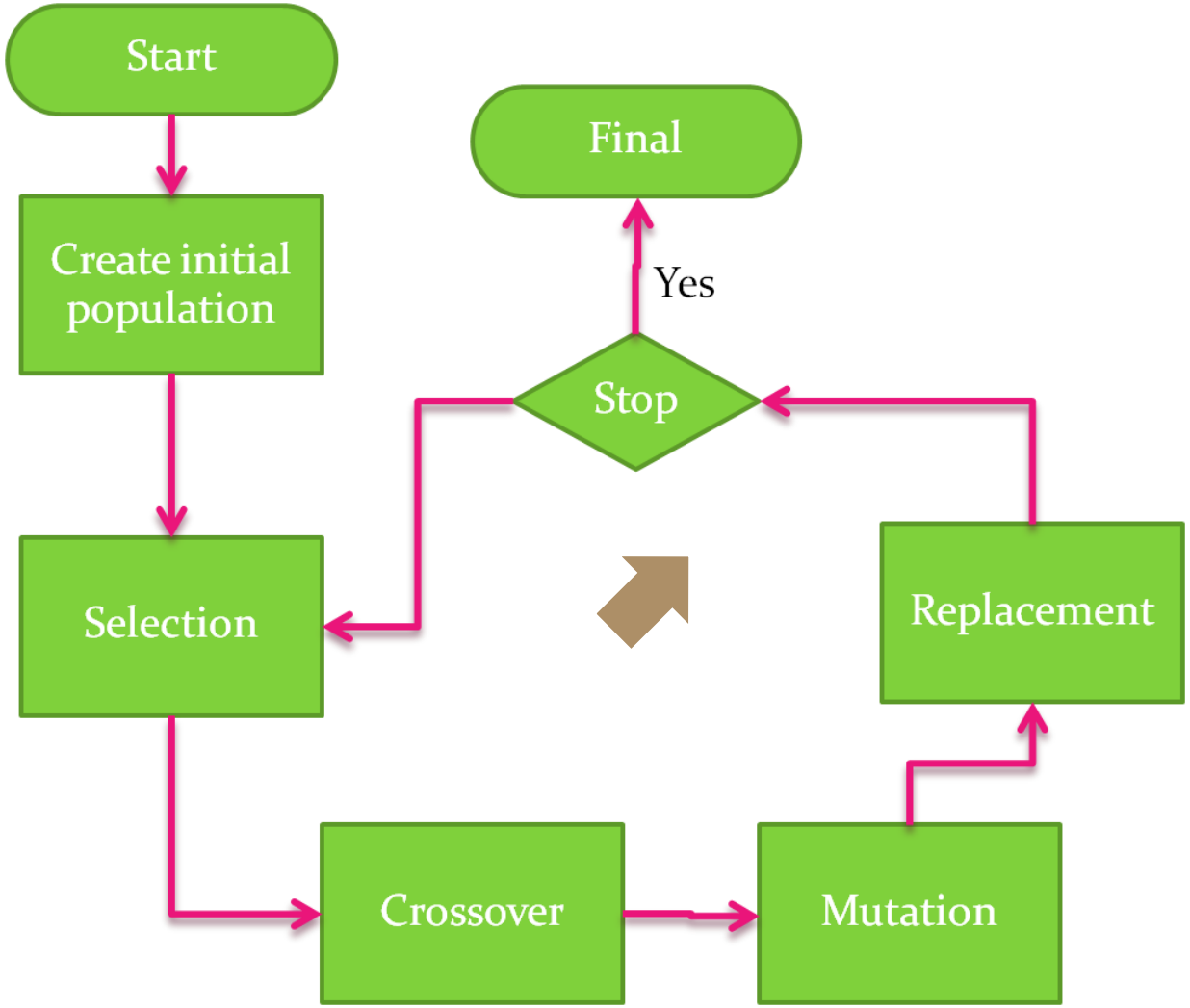
P3: 0011|0010 x=3 y=2

P4: 1001|0011 x=9 y=3

P5: 1101|0010 x=13 y=2

ตัวไหนดี เอาไปแทนที่พ่อแม่

Termination?



Example GA

```
1  //Example Simple-GA
2  //Programmer: Dr.Supakit Nootyaskool
3  //Email: supakitnootyaskool@gmail.com
4  //Work: IT-KMITL
5  //Date: 2012 July 02
6  //Code status: development
7  //(c)2012
8
9
10 #include <stdio.h>
11 #include <stdlib.h>
12 #include <conio.h>
13
14 typedef struct population
15 {
16     double a,b,c;
17     double fit;
18 };
19
20 void mathfunc(population &p)
21 {
22     double fitness;
23     fitness = p.a*p.a + p.b*p.b + p.c;
24     p.fit = fitness;
25 }
```

ภาษา C

ล้อกับหลักการ

ลักษณะ weight wx+f(x)

fitness function

```
27  void showdata(char name[10],population p)
28  {
29      printf("%s [%f,%f,%f] fit= %f\n",name,p.a,p.b,p.c,p.fit);
30  }
31
32  double random()
33  {
34      return (double) rand()/1000;
35  }
36
37  double random_0to1()
38  {
39      return (double) rand()/INT_MAX;
40  }
41
42  int main(void)
43  {
44      //parameters
45      const int maxpop = 10;
46      double pc = 0.1;
47      double pm = 0.8;
48      int maxgen = 3000;
49      int gen = 1;
50
```

```
51      //create initial population
52      population oldp[maxpop];
53      population newp[maxpop];
54      for(int i=0;i<maxpop;i++)
55      {
56          oldp[i].a = random();
57          oldp[i].b = random();
58          oldp[i].c = random();
59          mathfunc(oldp[i]);
60          showdata("oldpop",oldp[i]);
61      }
62
63      //generation cycle
64      while(1)
65      {
66          //Selection uses uniform random
67          printf("\n--SELECTION--\n");
68          for(int i=0;i<maxpop;i++)
69          {
70              newp[i] = oldp[rand()%maxpop];
71              showdata("newpop",newp[i]);
72          }
73
```

ฟังก์ชันเริ่มต้น

```
74          //crossover
75          printf("\n--CROSSOVER--\n");
76          for(int i=0;i<maxpop;i++)
77          {
78              int z1,z2;
79              z1 = rand()%maxpop;
80              z2 = rand()%maxpop;
81
82              if(random_0to1()<=pc)
83              {
84                  double t;
85                  t = newp[z1].a;
86                  newp[z1].a = newp[z2].a;
87                  newp[z2].a = t;
88              }
89
90              if(random_0to1()<=pc)
91              {
92                  double t;
93                  t = newp[z1].b;
94                  newp[z1].b = newp[z2].b;
95                  newp[z2].b = t;
96              }
```

```
99          {
100             double t;
101             t = newp[z1].c;
102             newp[z1].c = newp[z2].c;
103             newp[z2].c = t;
104         }
105         mathfunc(newp[i]);
106         showdata("newpop",newp[i]);
107     }
108
109     //Mutation
110     printf("\n--MUTATION--\n");
111     for(int i=0;i<maxpop;i++)
112     {
113         if(random_0to1()<=pm)
114             newp[i].a = random();
115         if(random_0to1()<=pm)
116             newp[i].b = random();
117         if(random_0to1()<=pm)
118             newp[i].c = random();
119         mathfunc(newp[i]);
120         showdata("newpop",newp[i]);
121     }
122
```

```
126         for(int j=0;j<maxpop;j++)
127             tmp[j] = oldp[j];
128         for(int j=0;j<maxpop;j++)
129             tmp[j+maxpop] = newp[j];
130 
131 
132         for(int j=0;j<maxpop*2;j++)
133         for(int i=0;i<maxpop*2-1;i++)
134         {
135             if(tmp[i].fit > tmp[i+1].fit)
136             {
137                 population t;
138                 t = tmp[i];
139                 tmp[i] = tmp[i+1];
140                 tmp[i+1] = t;
141             }
142         }
143         printf("\n--REPLACEMENT--\n");
144         for(int i=0;i<maxpop-1;i++)
145         {
146             showdata("tmp",tmp[i]);
147             oldp[i] = tmp[i];        //copy
148         }
149 
150         //pause
151         printf("\n--PAUSE Gen[%d]--",gen++);
152         getch();
153     }
154     getch();
```

```
oldpop [11.478000,29.358000,26.962000] fit= 1020.598648
oldpop [24.464000,5.705000,28.145000] fit= 659.179321
oldpop [23.281000,16.827000,9.961000] fit= 835.113890
oldpop [0.491000,2.995000,11.942000] fit= 21.153106
oldpop [4.827000,5.436000,32.391000] fit= 85.241025
oldpop [14.604000,3.902000,0.153000] fit= 228.655420
oldpop [0.292000,12.382000,17.421000] fit= 170.820188
oldpop [18.716000,19.718000,19.895000] fit= 758.983180

--SELECTION--
newpop [14.604000,3.902000,0.153000] fit= 228.655420
newpop [4.827000,5.436000,32.391000] fit= 85.241025
newpop [26.500000,19.169000,15.724000] fit= 1085.424561
newpop [0.292000,12.382000,17.421000] fit= 170.820188
newpop [18.716000,19.718000,19.895000] fit= 758.983180
newpop [11.478000,29.358000,26.962000] fit= 1020.598648
newpop [14.604000,3.902000,0.153000] fit= 228.655420
newpop [18.716000,19.718000,19.895000] fit= 758.983180
newpop [0.491000,2.995000,11.942000] fit= 21.153106
newpop [23.281000,16.827000,9.961000] fit= 835.113890

--CROSSOVER--
newpop [14.604000,3.902000,0.153000] fit= 228.655420
newpop [18.716000,19.718000,19.895000] fit= 758.983180
newpop [26.500000,19.169000,15.724000] fit= 1085.424561
newpop [23.281000,16.827000,9.961000] fit= 835.113890
newpop [18.716000,19.718000,19.895000] fit= 758.983180
newpop [14.604000,3.902000,0.153000] fit= 228.655420
newpop [14.604000,3.902000,0.153000] fit= 228.655420
newpop [0.292000,12.382000,17.421000] fit= 170.820188
newpop [0.491000,2.995000,11.942000] fit= 21.153106
newpop [23.281000,16.827000,9.961000] fit= 835.113890

--MUTATION--
newpop [32.439000,11.323000,21.538000] fit= 1202.037050
newpop [2.082000,16.541000,31.115000] fit= 309.054405
newpop [29.658000,9.930000,2.306000] fit= 980.507864
newpop [22.386000,28.745000,19.072000] fit= 1346.480021
newpop [5.829000,15.573000,16.512000] fit= 293.007570
newpop [13.290000,18.636000,24.767000] fit= 548.691596
newpop [15.574000,12.052000,1.150000] fit= 388.950180
newpop [21.724000,3.430000,30.191000] fit= 513.888076
newpop [11.337000,12.287000,10.383000] fit= 289.880938
newpop [8.909000,9.758000,18.588000] fit= 193.176845

--REPLACEMENT--
tmp [0.491000,2.995000,11.942000] fit= 21.153106
tmp [4.827000,5.436000,32.391000] fit= 85.241025
tmp [0.292000,12.382000,17.421000] fit= 170.820188
tmp [8.909000,9.758000,18.588000] fit= 193.176845
tmp [14.604000,3.902000,0.153000] fit= 228.655420
tmp [11.337000,12.287000,10.383000] fit= 289.880938
tmp [5.829000,15.573000,16.512000] fit= 293.007570
tmp [2.082000,16.541000,31.115000] fit= 309.054405
```

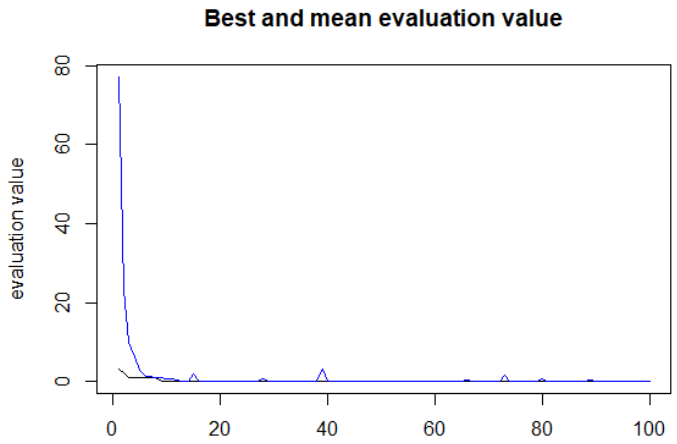
```
tmp [1.054000,0.213000,0.475000] fit= 1.631285
tmp [1.369000,0.641000,4.820000] fit= 7.105042
tmp [1.888000,1.887000,0.648000] fit= 7.773313
tmp [1.453000,2.141000,1.807000] fit= 8.502090
tmp [1.831000,1.668000,5.099000] fit= 11.233785
tmp [0.821000,2.175000,6.360000] fit= 11.764666
tmp [0.632000,1.950000,7.763000] fit= 11.964924
tmp [0.533000,2.767000,4.415000] fit= 12.355378
tmp [2.890000,1.869000,0.948000] fit= 12.793261

--PAUSE Gen[599]--
--SELECTION--
newpop [1.054000,0.213000,0.475000] fit= 1.631285
newpop [1.369000,0.641000,4.820000] fit= 7.105042
newpop [1.831000,1.668000,5.099000] fit= 11.233785
newpop [1.369000,0.641000,4.820000] fit= 7.105042
newpop [1.888000,1.887000,0.648000] fit= 7.773313
newpop [0.533000,2.767000,4.415000] fit= 12.355378
newpop [1.888000,1.887000,0.648000] fit= 7.773313
newpop [0.533000,2.767000,4.415000] fit= 12.355378
newpop [1.453000,2.141000,1.807000] fit= 8.502090
newpop [2.890000,1.869000,0.948000] fit= 12.793261

--CROSSOVER--
newpop [1.054000,0.213000,0.475000] fit= 1.631285
newpop [1.369000,0.641000,4.820000] fit= 7.105042
newpop [1.888000,1.887000,0.648000] fit= 7.773313
newpop [1.369000,0.641000,4.820000] fit= 7.105042
newpop [1.888000,1.887000,0.648000] fit= 7.773313
newpop [1.054000,0.213000,0.475000] fit= 1.631285
newpop [1.831000,1.668000,5.099000] fit= 11.233785
newpop [0.533000,2.767000,4.415000] fit= 12.355378
newpop [0.533000,2.767000,4.415000] fit= 12.355378
newpop [1.369000,0.641000,4.820000] fit= 7.105042

--MUTATION--
newpop [21.360000,25.005000,16.756000] fit= 1098.255625
newpop [6.344000,28.462000,25.972000] fit= 876.303780
newpop [2.175000,32.700000,7.145000] fit= 1081.165625
newpop [12.622000,16.545000,19.500000] fit= 452.551909
newpop [25.318000,31.141000,10.898000] fit= 1621.661005
newpop [30.084000,12.271000,2.439000] fit= 1058.063497
newpop [29.387000,23.826000,18.383000] fit= 1449.657045
newpop [27.941000,12.300000,16.524000] fit= 948.513481
newpop [20.183000,11.349000,29.143000] fit= 565.296290
newpop [15.495000,24.412000,5.393000] fit= 841.433769

--REPLACEMENT--
tmp [1.054000,0.213000,0.475000] fit= 1.631285
tmp [1.369000,0.641000,4.820000] fit= 7.105042
tmp [1.888000,1.887000,0.648000] fit= 7.773313
tmp [1.453000,2.141000,1.807000] fit= 8.502090
tmp [1.831000,1.668000,5.099000] fit= 11.233785
tmp [0.821000,2.175000,6.360000] fit= 11.764666
tmp [0.632000,1.950000,7.763000] fit= 11.964924
tmp [0.533000,2.767000,4.415000] fit= 12.355378
```

GA IN R

Activity 11.1 Real Value GA in R

```
##############################################
#Activity 11  Real Value  GA in R
##############################################

rm(list=ls())

#install.packages("genalg")  #Genetic algorithm
library(genalg)

chromosome = c(10,10,10) # x and y

evalFunc = function(X = c())
{
    r = abs(X[1]*X[1] + X[2]*X[2]+ X[3])
    #print(r)
    return(r)
}
##############################################
```

```
iter = 100
GAmodel <- rbga(
    stringMin = c(-10,-10,-10),
    stringMax = c(10,10,10),
    popSize = 30,
    iters = iter,
    mutationChance = 0.001,
    elitism = T,                #Crossover
    evalFunc = evalFunc )
##############################################
GAmodel
plot(GAmodel)
```



ถ้าสนใจ ติดต่อ + email สอบถามได้

Research

พวก RNN clock selection

ไม่ค่อยทำ ด NN

NN & DL
CONCLUSION

Asst.Prof.Dr.Supakit Nootyaskool

Study map

- 1.Basic programming
  - R-programming
- 2. Perceptron
  - Activity function
- 3. Feed Forward NN
  - Logistic function
- 4. Feed Forward NN
  - XOR gate
  - Multi-layer perceptron
- 5. Example & Library Feed Forward NN
  - N:N, 1:N model
  - iris dataset

- 6. Writing NN Code
  - Data scaling, Confusion matrix
  - Writing NN code
- 7. Recurrent Neural Network
- 8. Apply RNN & Library
- 9. GRU LSTM
- 10. CNN
- 11 Apply GA to NN

ไม่รู้