Lecture Note

Practice

Evolutionary Computation

Akira Imada
Brest State Technical University

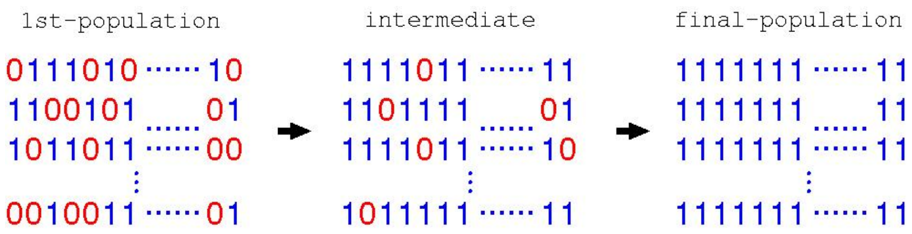
Last modified on 21 April 2016

I. All One Problem

What is All one Problem?

Assume "1" is a good gene and "0" is a bad gene.

Then try to evolve a population of random binary chromosomes, with fitness being the number of gene "1" - the less the better.



Exercise 1 *1. Create 100 random binary-chromosomes each with 1000 genes.*

2. Fitness is the number of "1" in one chromosome – the more the better.

3. Select 2 chromosomes at random from the better half of the population.

4. Create a child chromosome by a onr-point-crossover.

5. Give the child a mutation with a probability of 1/1000 = 0.001.

6. Repeat from 2. to 5. 100 times and create the next generation.

7. Repeat 6. until the fitness value does not change any more.

8. Show the result:

(1) Desplay the best chromosome in the 1st, an intermediate & final generation.

(2) Desplay the best and average fitness vs. generation.

Exercise 2 *1. Try to improve the performance by:*

(1) changing mutation rate, say, 0.005, 0.010 $\cdots$.

(2) increasing the number of population, say, 200, 300 $\cdots$.

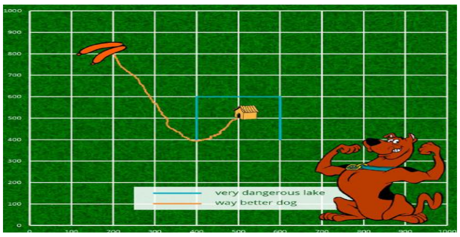
(3) changing the initial rate of "1" and "0:", say, 600 vs 400, 700 vs 300 $\cdots$.

2. Show all the successful results.

II. Lucky Dog

What is Lucky Dog Problem?

Simulate one dog by a chromosome with 1000 genes of 1, 2, 3 or 4.



From the work by Dmitry Verenich (2015)

Exercise 3 *1. Simulate 100 dogs in the gridworld (0,0)-(1000,1000) with sausage at (800,800) where all dogs start from (500,500) looking for the sausage by:*

(1) assuming all dog has a chromosome whose each of 1000 genes are 1, 2, 3 or 4,

(2) moving step by step according to his/her chromosome,

(3) with 1, 2, 3, 4 meaning a movement toward up, down, right, left, respectively.

2. Show the result with:

(1) the best and average fitness vs. generation of the 5 best dogs in the 1st, an intermediate, and final generation.

(2) the route of the 5 best dogs in the 1st, an intermediate, and final generation.

Exercise 4 *Observe what will happen when two sausages exist in the gridworld (0,0)-(1000,1000) at (200,800) & (800,800).*

III. Minimization of Test Functions

High-dimensional Test Function

Problem is minimize or maximize a n-dimensinal function

$$y = f(x_1, x_2, x_3, \cdots, x_n)$$

We may use chromosomes with n genes of real value, such as $(0.32, -0.51, 0.48, \cdots, -0.93)$

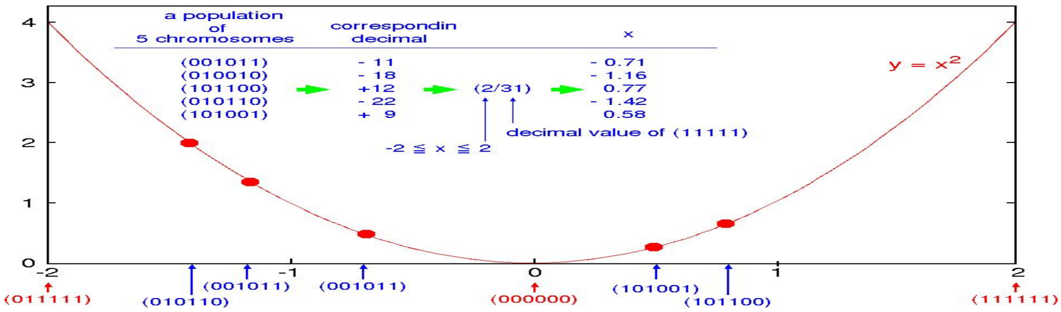
For example, minimize

$$y = x_1^2 + x_1^2 + x_1^2 + \cdots + x_{20}^2$$

What then about 2-D cases?

E.g. $y = x^2$. Then, the way used in 20-D (real-value 20 genes) cannot be used, because chromosome with only one real-value gene doesn't make sense any more.

So let's represent each x with binary chromosome as below!



(1) 20-Dimensional Schwefel's function

Exercise 5 1. *Minimize y in the following way!*

$$y = x_1 \sin(|x_1|) + x_2 \sin(|x_2|) + \cdots + x_{20} \sin(|x_{20}|)$$

(1) Represent each of $x_i (i = 1, \cdots, 20)$ by a chromosome with 20 genes.

(2) Create a population of 20 chromosomes at random, with fitness being y.

(3) Evolve this population till fitness dosen't change.

2. *Show*

(1) the graph of fitness vs generation.

(2) Create a population of 20 chromosomes at random, with fitness being y.

(3) Evolve this population till fitness dosen't change.

Hints

Minimize y

$$y = f(x_1, x_2, x_3, \cdots, x_{20}) \quad -1 \leqq x_i \leqq 1$$

$i = (1,2, \ldots, 20)$

with chromosomes such as

(0.54, -0.72, -0.59, ⋯ , 3.86)

↑ x_1 ↑ x_2 ↑ x_3 ↑ x_{20} → calculate y = fitness

- Create a population of 20 such chromosomes at random
  (Smaller the fitness, the better the chromosome.)
- Evolve the population till fitness no change

(2) 2-D version of Schwefel's function

Exercise 6 *1. Minimize*

$$y = x \sin(|x|)$$

in the following way!

(1) Represent value of x by a 10-bit binary chromosome.

(2) Create a population of 20 chromosomes at random, with fitness being y.

(3) Evolve this population till fitness dosen't change.

2. Show

(1) the graph of fitness vs generation.

(2) all 20 points (x, y) in the 1st, an intermediate, and final generation.

Hints

Minimize y

$y = f(x) \quad -1 \leqq x \leqq 1$

with binary chromosomes such as

(1 1 0 0 0 1 0 1 1 1 0) → x = 558/1023 = 0.55

↑ + or -

or

(0 0 1 0 0 0 1 0 0 1 1) → x = -275/1023 = -0.27

↑ + or -

calculate y = fitness

- Create a population of 20 such chromosomes at random ➡ (means) 20 points
  (Smaller the fitness, the better the chromosome.)
- Evolve the population till fitness no change
- Plot 20 points of (x, y) in the first, an intermidiate and final generation

(3) 3-D version of Schwefel's function

Exercise 7 1. *Minimize*

$$y = x_1 \sin(|x_1|) + x_2 \sin(|x_2|)$$

in the following way!

(1) Represent value of x by a 10-bit binary chromosome.

(2) Create a population of 20 chromosomes at random, with fitness being y.

(3) Evolve this population till fitness dosen't change.

2. *Show*

(1) the graph of fitness vs generation.

(2) all 20 points (x, y) in the 1st, an intermediate, and final generation.

Hints

Minimize y

$$y = f(x_1, x_2, x_3, \cdots, x_{20}) \quad -1 \leqq x_i \leqq 1$$

$i = (1,2, \ldots, 20)$

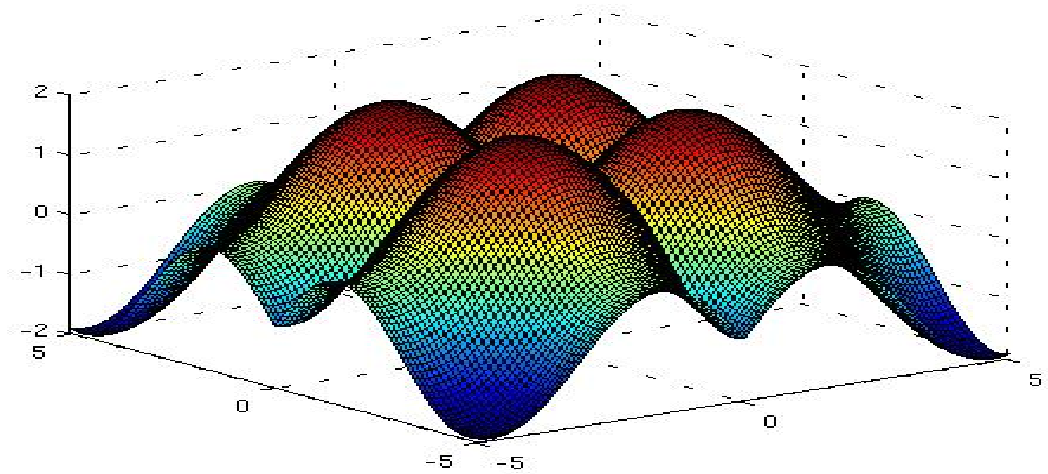
with chromosomes such as

(0.54, -0.72, -0.59, ··· , 3.86)

↑ x_1 ↑ x_2 ↑ x_3 ... ↑ x_{20} → calculate y = fitness

- Create a population of 20 such chromosomes at random (Smaller the fitness, the better the chromosome.)
- Evolve the population till fitness no change

3-D graph of Schwefel function as a hint



Yet the other test functions

Rastrigin's Function

Exercise 8 *1. Minimize the following y in (i) 20-D, (ii) 3-D and (iii) 2-D cases.*

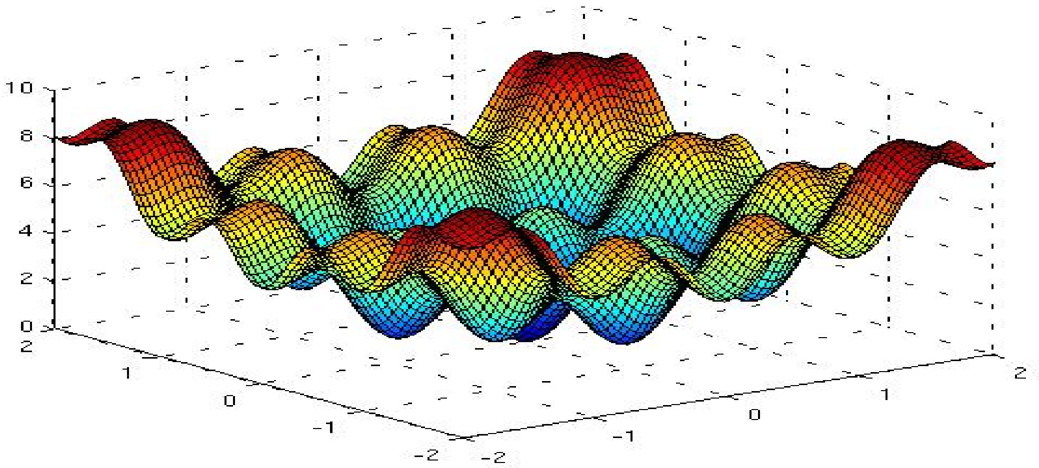
$$y = nA + \sum_{i=1}^{n} (x_i^2 - A\cos(2\pi x_i))$$

2. Show the following graphics in each of 3 cases (i), (ii) and (iii).

(1) the graph of fitness vs generation.

(2) Create a population of 20 chromosomes at random, with fitness being y.

3-D graph of Rastrigin's function as a hint



Griewangk's functio

Exercise 9 *1. Minimize the following y in (i) 20-D, (ii) 3-D and (iii) 2-D cases.*

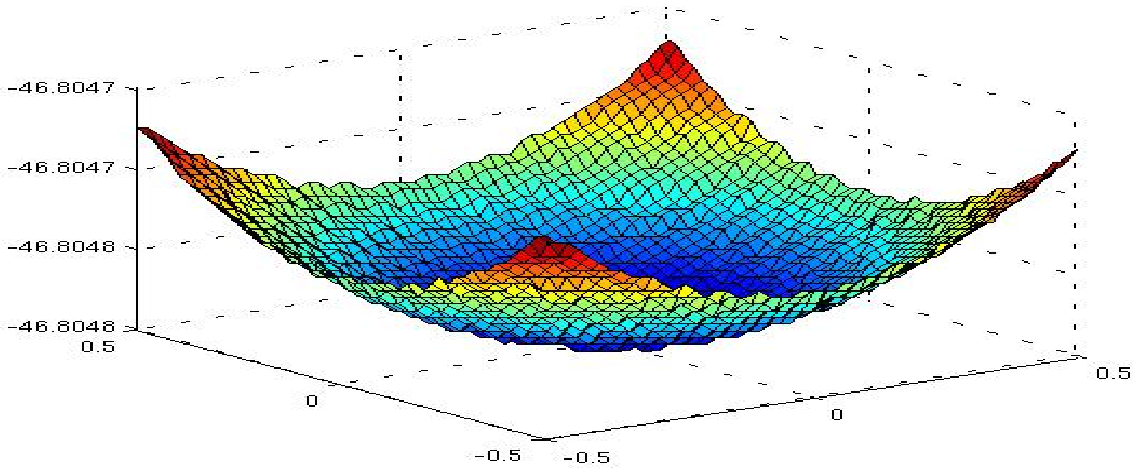
$$y = \sum_{i=1}^{n} x_i^2/4000 - \prod_{i=1}^{n} \cos(x_i/\sqrt{i}) + 1$$

2. Show the following graphics in each of 3 cases (i), (ii) and (iii).

(1) the graph of fitness vs generation.

(2) Create a population of 20 chromosomes at random, with fitness being y.

3-D graph of Griewangk's function as a hint



Ackley's function

Exercise 10 *1. Minimize the following y in (i) 20-D, (ii) 3-D and (iii) 2-D cases.*

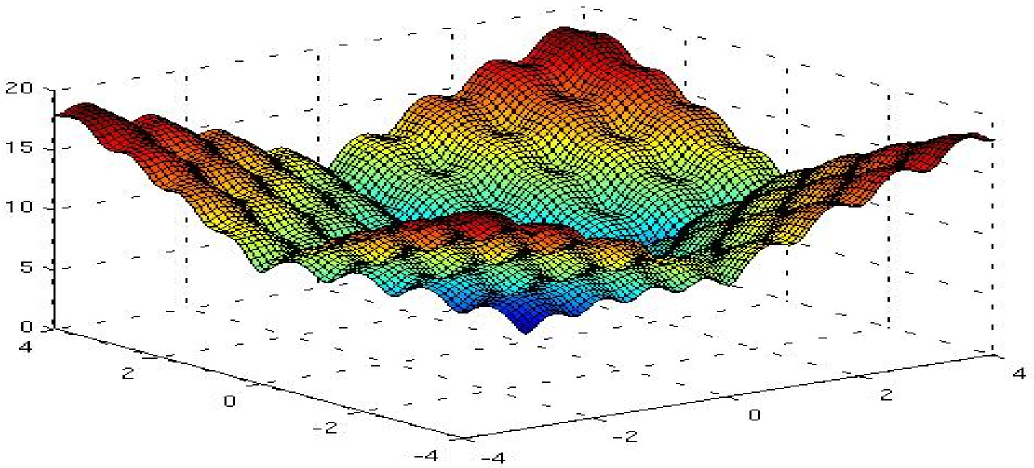
$$y = -20 \sum_{i=1}^{n} \exp\left(-0.2\sqrt{x_i^2/n}\right) - \exp\left(\left(\sum_{i=1}^{n} \cos 2\pi x_i\right)/n\right) + 20 + e$$

2. Show the following graphics in each of 3 cases (i), (ii) and (iii).

(1) the graph of fitness vs generation.

(2) Create a population of 20 chromosomes at random, with fitness being y.

3-D graph of Ackley's function as a hint

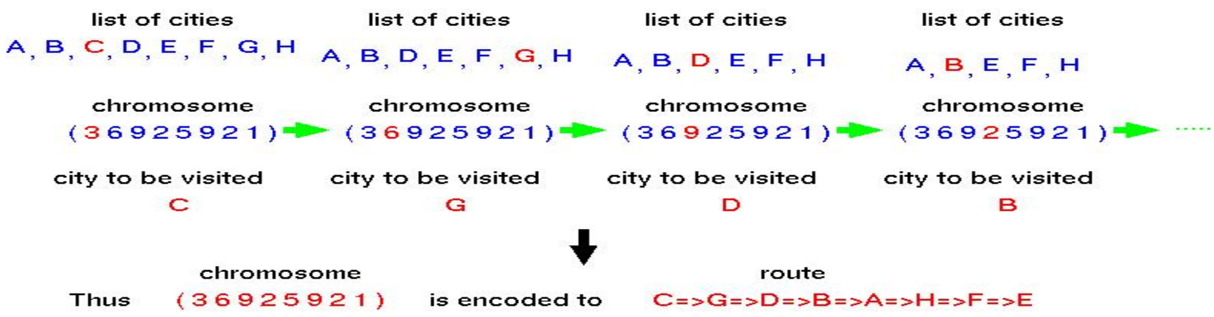


IV. Traveling Salesperson Problem (TSP)

What is TSP?

Assume a salesperson, starting from his/her city, should visit N cities, all of them but only once, and then return to the city he/she started. The problem is to look for the minimum length of such tour. Real examples of cities are as follows.

How to encode chromosome representing one tour?



(1) TSP with 5 cities of random location

Exercise 11 *1. Create 5 cities Z, A, B, C, D at random in the x-y cordinate from (0,0) to (10,10). All A,B,C and D must be visited only once, starting from Z and return to Z.*

2. Calcurate distance matrix (5×15).

3. Find all the possible tours and calculate the length of each tour.

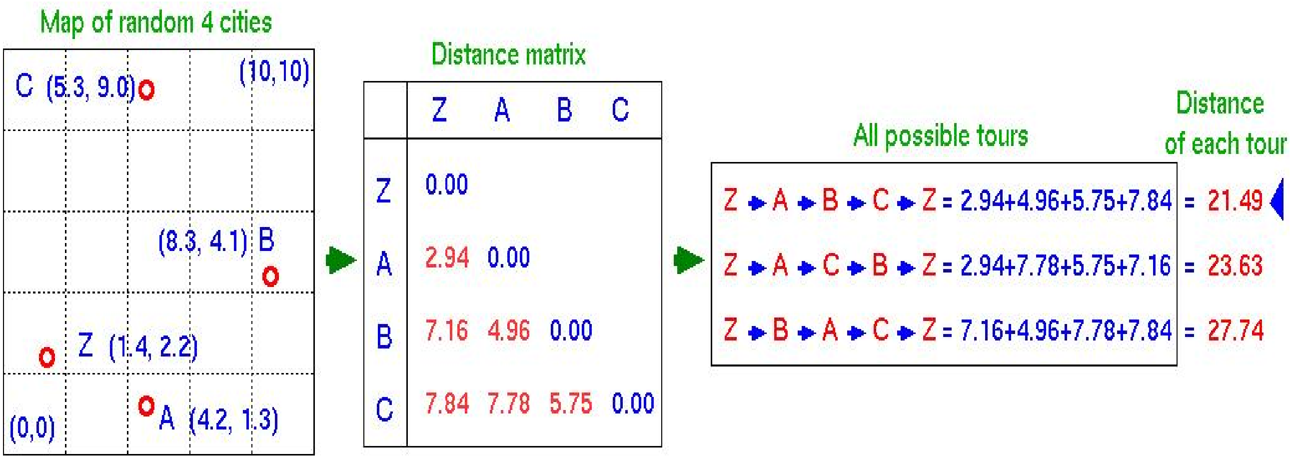
4. Apply GA and evolve chromosomes to be the tours of minimum length.

5. Show (1) map of 5cities, (2) distance matrix, (3) the list of all tours with its length. *(See examples in the next page!)*

6. Also show

(4) the graph of fitness vs generation.

(5) The minimum tour in the 1st, two intermediate, and the final generation.


Map of random 4 cities
C (5.3, 9.0)
(10,10)
(8.3, 4.1) B
Z (1.4, 2.2)
(0,0)
A (4.2, 1.3)
Distance matrix
Z A B C
Z 0.00
A 2.94 0.00
B 7.16 4.96 0.00
C 7.84 7.78 5.75 0.00
All possible tours
Distance of each tour
Z → A → B → C → Z = 2.94+4.96+5.75+7.84 = 21.49
Z → A → C → B → Z = 2.94+7.78+5.75+7.16 = 23.63
Z → B → A → C → Z = 7.16+4.96+7.78+7.84 = 27.74

(2) TSP with 25 cities of a fixed location

Exercise 12 *1. Assume 15 cities as shown in the next page (start from Z and return to Z).*

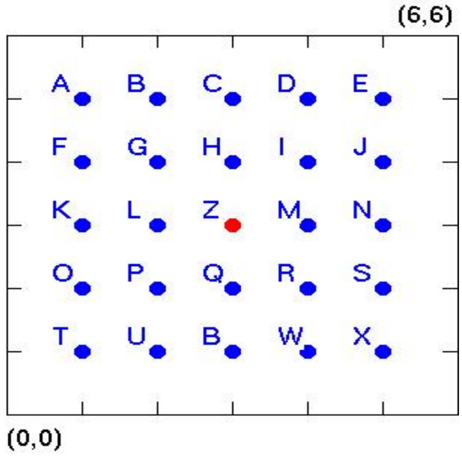
2. Calcurate distance matrix (15×15).

3. Apply GA and evolve chromosomes to be the tours of minimum length.

4. Also show

(5) the graph of fitness vs generation.

(6) The minimum tour in the 1st, two intermediate, and the final generation.


(6,6)
A B C D E
F G H I J
K L Z M N
O P Q R S
T U B W X
(0,0)

(3) TSP with 15 cities on a circle

Exercise 13 *1. Create 25 cities at random on circle* $(x-5)^2+(y-5)^2=25$. *See next page (start from Z and return to Z).*

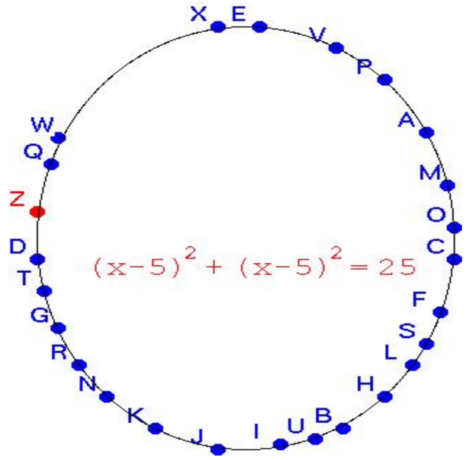
2. Calcurate distance matrix (15×15).

3. Apply GA and evolve chromosomes to be the tours of minimum length.

4. Also show

(5) the graph of fitness vs generation.

(6) The minimum tour in the 1st, two intermediate, and the final generation.


X
E
V
P
A
M
O
C
F
S
L
H
B
U
I
J
K
N
R
G
T
D
Z
Q
W
(x-5)^2 + (x-5)^2 = 25

IV. What if multiple solutions?

(1) Fitness Sharing Algorithm

Maximization of a 2-D function with 2 peaks

Exercise 14 *1. Create a population of 20 binary cromosomes with 10 genes each of which represent a value of* $x \in [0, 2]$

2. Assuming fitness being $y = 1 - \sin^2(2\pi x)$*, apply Fitnessg Sharing Algorithm with* $s(d_{ij}) = 1 - (d_{ij}/\sigma)$ *where* $\sigma = 0.06$

3. Show

(4) the graph of fitness vs generation.

(5) 20 points (x, y) *on the graph in the 1st generations, two intermediate generations, and the final generation.*

(6) Also the table of original fitness and shared fitness for all the 20 chromosomes in the above 5 generations *(See next page).*

generation = N-1

population	x	original fitness	shared fitness
#0			
#1			
#2			
#3			
#4			
#5			
#6			
#7			
#8			
#9			

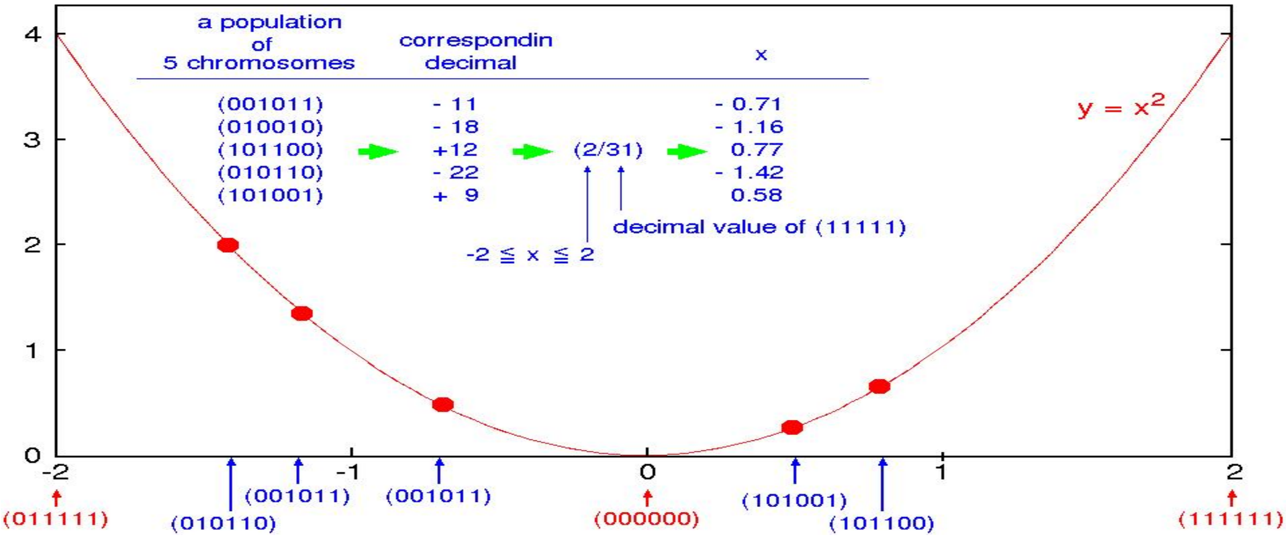
generation = N

population	x	original fitness	shared fitness
e.g.	(101001)=0.29		
#0			
#1			
#2			
#3			
#4			
#5			
#6			
#7			
#8			
#9			

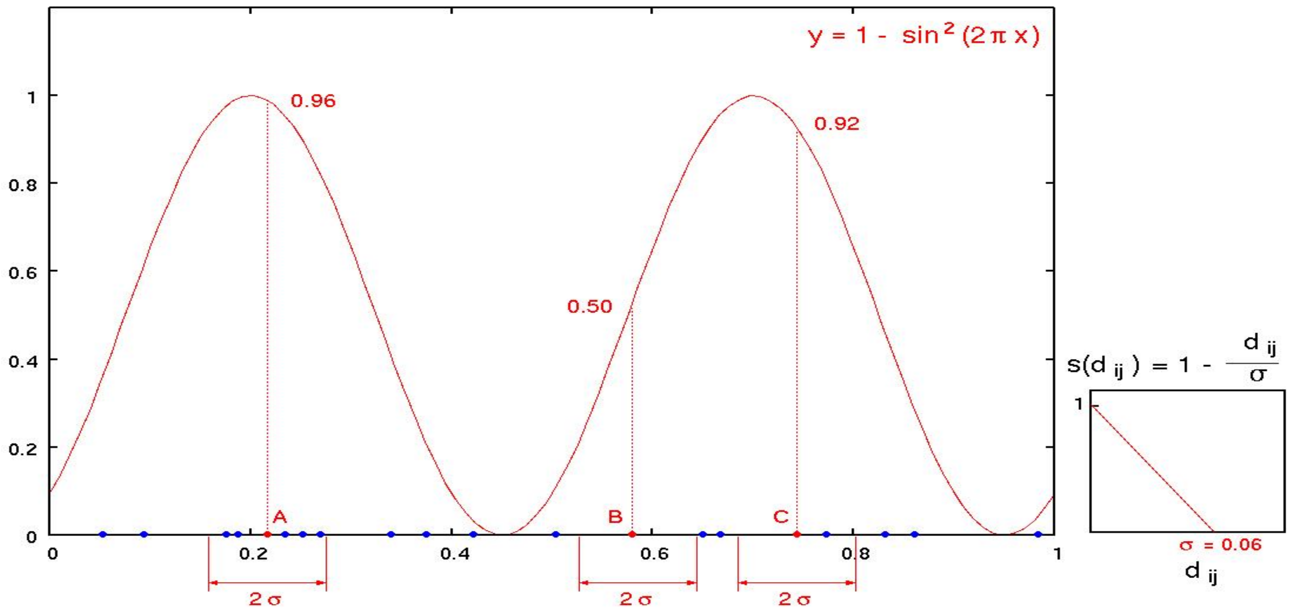
generation = N+1

population	x	original fitness	shared fitness
#0			
#1			
#2			
#3			
#4			
#5			
#6			
#7			
#8			
#9			

Hints


a population of 5 chromosomes
correspondin decimal
x
(001011) - 11 - 0.71
(010010) - 18 - 1.16
(101100) +12 (2/31) 0.77
(010110) - 22 - 1.42
(101001) + 9 0.58
decimal value of (11111)
-2 ≦ x ≦ 2
y = x2
4
3
2
1
0
-2
-1
0
1
2
(011111)
(010110)
(001011)
(001011)
(000000)
(101001)
(101100)
(111111)

Hints



Lucky Dog Problem with Fithess Sharing Algorithm

Exercise 15 *1. Create a population of N cromosomes of 1000 genes each of whose values is 1, 2, 3 or 4. Chromosome represents 1000 steps of one dog.*

2. Assuming fitness being {600 - distance to the nearest sausage}, apply Fitnessg Sharing Algorithm with $\sigma = 5$

3. Show

(4) The graph of fitness vs generation of 8 highest fitness dogs.

(5) The route of 8 highest fithess dogs in the 1st generations, two intermediate generations, and the final generation.

(6) Also the table of original fitness and shared fitness for all the N chromosomes in the above 5 generations *(See next page).*

generation = N-1

population	location	original fitness	shared fitness
#1			
#2			
#3			
#4			
#5			
#6			
#7			
#8			

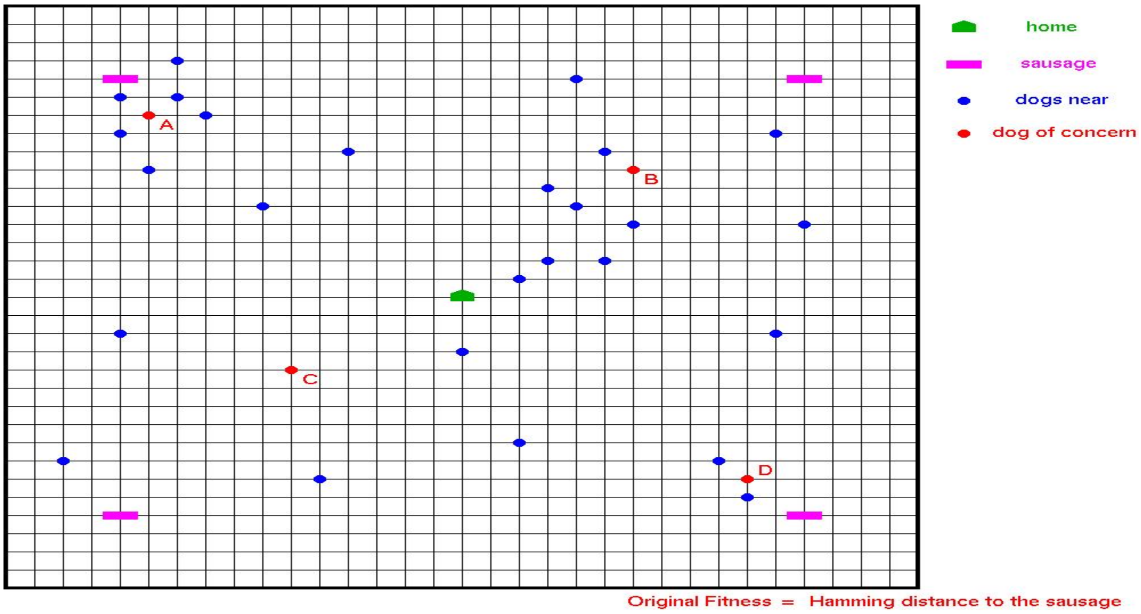
generation = N

population	location	original fitness	shared fitness
e.g.	(650, 723)	600 - (150+77) = 378	153.8
#1			
#2			
#3			
#4			
#5			
#6			
#7			
#8			

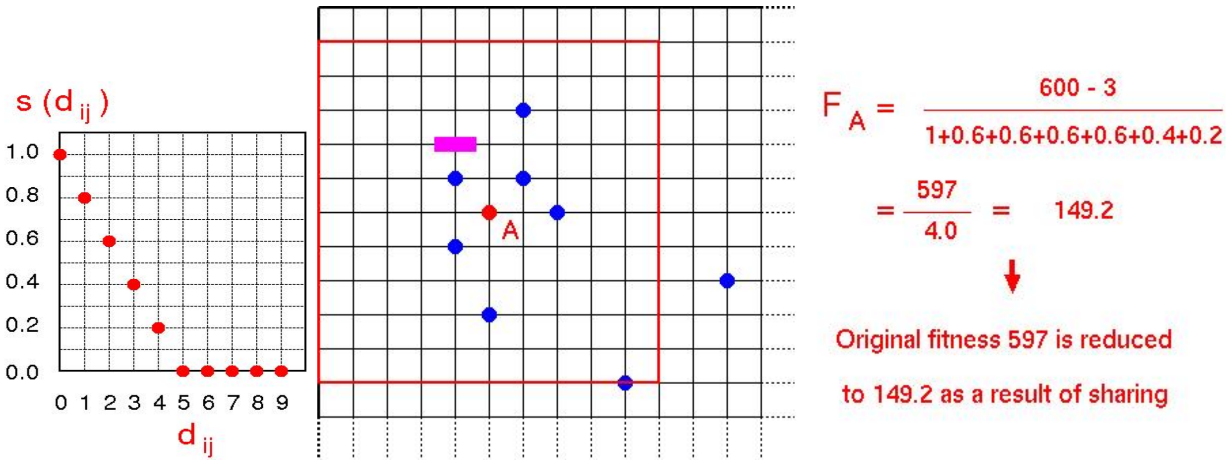
generation = N+1

population	location	original fitness	shared fitness
#1			
#2			
#3			
#4			
#5			
#6			
#7			
#8			

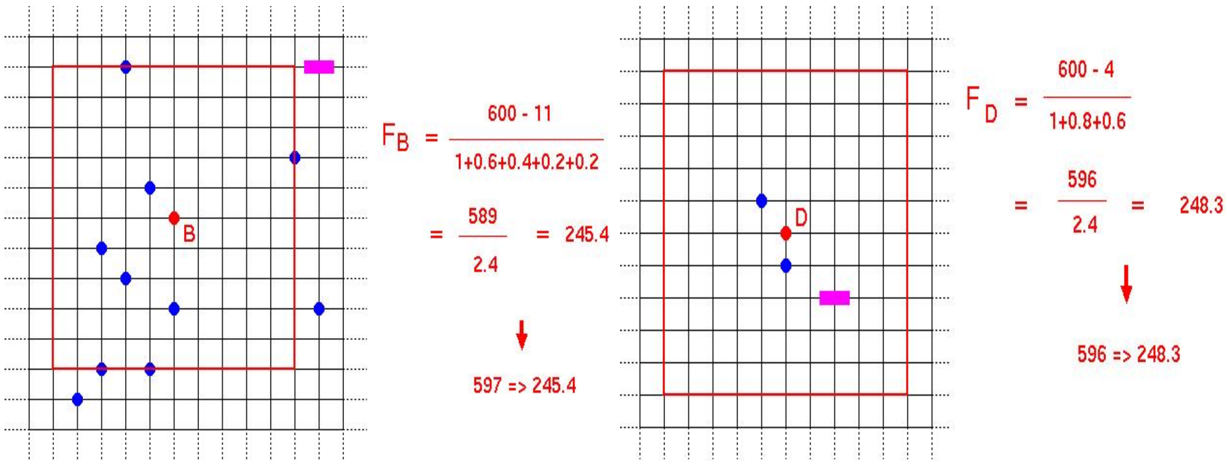
An example of where those dogs are now.



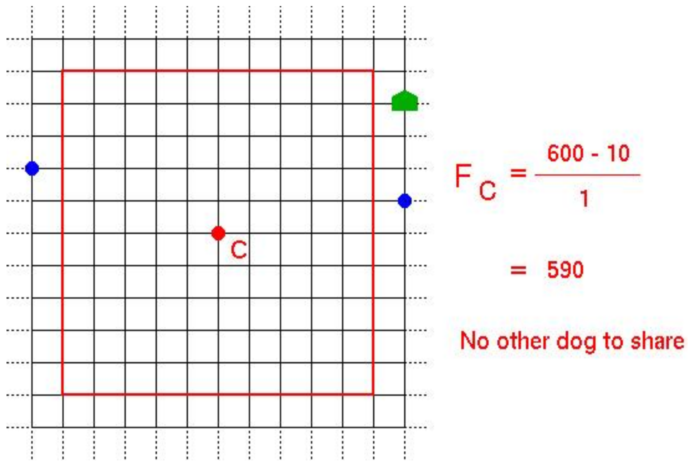
Those dogs whom the dog A should share



How dogs B and D share fitness?



The dog C is really lucky because no need to share



(2) Crowding Algorithm

Crowding Algorithm

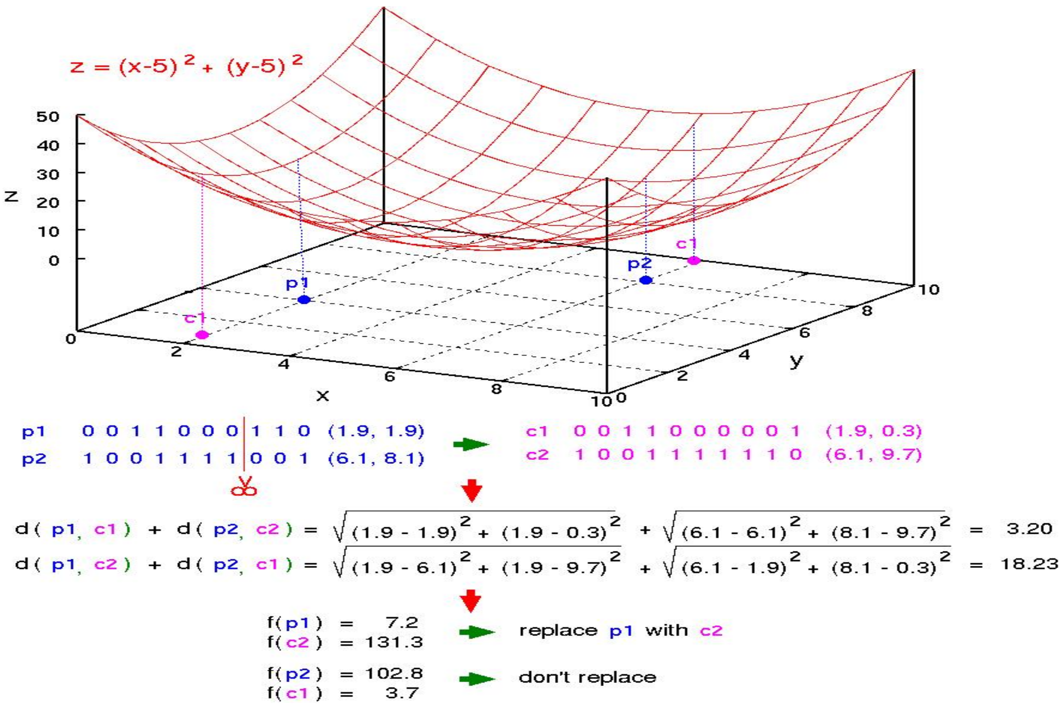
Algorithm 1 *1. Choose two parents, p_1 and p_2, at random.*

2. Produce two children, c'_1 and c'_2.

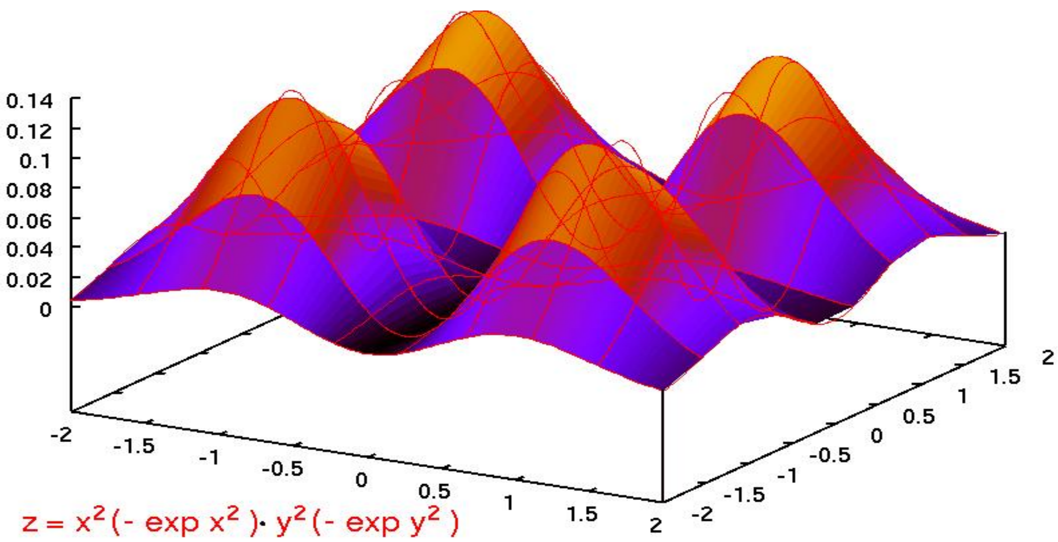
4. Replace parent with child as follows:

- *IF* $d(p_1, c_1) + d(p_2, c_2) > d(p_1, c_2) + d(p_2, c_1)$
  * *IF* $f(c_1) > f(p_1)$ *THEN* *replace p_1 with c_1*
  * *IF* $f(c_2) > f(p_2)$ *THEN* *replace p_2 with c_2*
- *ELSE*
  * *IF* $f(c_2) > f(p_1)$ *THEN* *replace p_1 with c_2*
  * *IF* $f(c_1) > f(p_2)$ *THEN* *replace p_2 with c_1*

See an example in the next page.


z = (x-5)² + (y-5)²
z
x
y
p1 0 0 1 1 0 0 0 1 1 0 (1.9, 1.9)
p2 1 0 0 1 1 1 1 0 0 1 (6.1, 8.1)
c1 0 0 1 1 0 0 0 0 0 1 (1.9, 0.3)
c2 1 0 0 1 1 1 1 1 1 0 (6.1, 9.7)
d(p1, c1) + d(p2, c2) = √((1.9 - 1.9)² + (1.9 - 0.3)²) + √((6.1 - 6.1)² + (8.1 - 9.7)²) = 3.20
d(p1, c2) + d(p2, c1) = √((1.9 - 6.1)² + (1.9 - 9.7)²) + √((6.1 - 1.9)² + (8.1 - 0.3)²) = 18.23
f(p1) = 7.2
f(c2) = 131.3
replace p1 with c2
f(p2) = 102.8
f(c1) = 3.7
don't replace

A 3D with 4 peak-function



A maxiization of 3D function with Crowding Algorithm

Exercise 16 *1. Create a population of 10 binary cromosomes of 22 genes each of which represents one point in $x - y$ plane $-2 \leq x, y \leq 2$.*

2. With fitness being value of z, maximize z applying Crowding Algorithm.

3. Show

(4) The graph of fitness vs generation, repeating evolution untill fitness becomes stable.

(5) The 3D graph including 10 points on the surface in the 1st generations, two intermediate generations, and the final generation.

(6) Also the table such as shown in the next page in the above 5 generations.

generation = N-1

#0	
#1	
#2	
#3	
#4	
#5	
#6	
#7	
#8	
#9	

generation = N

	selected parents & fitness	created children & fitness	d(p1,c1) +d(p2,c2)	d(p1,c2) +d(p2,c1)	new chromosomes
e.g.	p1: (0011000110) = (1.9,1.9) z=8.26 p2: (1001111001) = (1.9,1.9) z=8.26	c1: (0011000001) = (1.9,0.3) z=8.26 c2: (1001111110) = (6.1,9.7) z=8.26	1.6	8.9	(1001111110) (1001111001)
#0					
#1					
#2					
#3					
#4					
#5					
#6					
#7					
#8					
#9					

generation = N+1

#0	
#1	
#2	
#3	
#4	
#5	
#6	
#7	
#8	
#9	

Hints

E.g. To maximize $z = f(x, y)$ $\quad -2 \leqq x, y \leqq 2$

chromosome → coordinate of point

(11000101110 00100010011) → (1.10, -0.54)

$x = 11000101110 \rightarrow +558 \times 2 / 1023 = 1.10$

$y = 00100010011 \rightarrow -275 \times 2 / 1023 = -0.54$

V. What if multiple fitness functions?

Parate Optimum Solution- Dominate & Rank

We now try to evoluve a population of chromosomes each of which has a multiple criteria (fitnesses). So we cannot sort the population according to the fitness like before (because we have mulitiple fitnesses).

In this situation, when chromosome A is better than chromosome B with respect to all the criteria (fitnesses) it is said to be A domonates B.

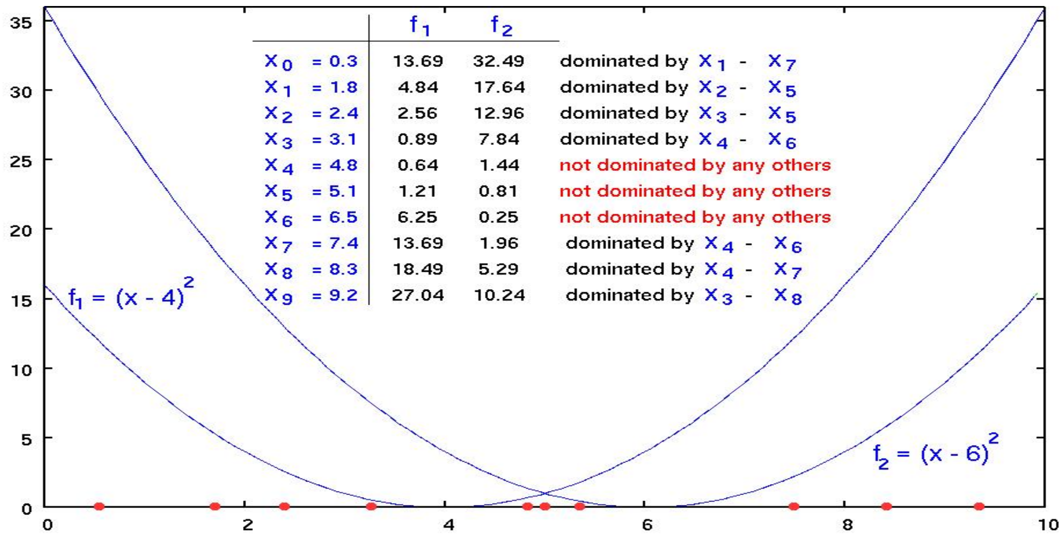
Then how many a chromosome dominates others in the population is called rank .

We sort population according to rank.

If a chromosome is not dominated by any others in the population, then this

chromosome is called Non-dominated solution or Parate Optimum Solution.

An example


f1 f2
x0 = 0.3 13.69 32.49 dominated by x1 - x7
x1 = 1.8 4.84 17.64 dominated by x2 - x5
x2 = 2.4 2.56 12.96 dominated by x3 - x5
x3 = 3.1 0.89 7.84 dominated by x4 - x6
x4 = 4.8 0.64 1.44 not dominated by any others
x5 = 5.1 1.21 0.81 not dominated by any others
x6 = 6.5 6.25 0.25 not dominated by any others
x7 = 7.4 13.69 1.96 dominated by x4 - x6
x8 = 8.3 18.49 5.29 dominated by x4 - x7
x9 = 9.2 27.04 10.24 dominated by x3 - x8
f1 = (x - 4)^2
f2 = (x - 6)^2
35
30
25
20
15
10
5
0
0
2
4
6
8
10

Algorithm

Algorithm 2 *1. Initialize the population.*

2. Select individuals uniformly from population.

3. Perform crossover and mutation to create a child.

4. Calculate the rank of the new child.

5. Find the individual in the entire population that is most similar to the child. Replace that individual with the new child if the child's ranking is better, or if the child dominates it.

[1]

6. Update the ranking of the population if the child has been inserted.

7. Perform steps 2-6 according to the population size.

8. If the stop criterion is not met go to step 2 and start a new generation.

[1] Step 5 implies that the new child is only inserted into the population if it dominates the most similar individual, or if it has a lower ranking, i.e. a lower degree of dominance.

The restricted replacement strategy also constitutes an extreme form of elitism, as the only way of replacing a

An example

Exercise 17 (Parate Optimal Solutions)
Try the algorithm above with two objective functions $y_1 = (x-2)^2$ *and* $y_2 = (x-4)^2$ *as follows:*

1. *Create 20 10-bit binary chromosomes, assuming each chromosome represent* x*-cordinate ranges from 0 to 6 with (0000· · · 00) and (1111· · · 11) being corresponding to 0 and 6, respectively.*
2. *Calculate* y_1 *and* y_2 *for each of 20* x*'s represented by these 20 chromosomes.*
3. *Create a table with 5 columns: (i) chromosome, (ii) its* x *value, (iii) its* y_1 *value, (iv) its* y_2 *value, (v) how many these* (y_1, y_2) *dominates others (rank).*
4. *Plot these 20 points on each of two graphs (e.g., red and blue color).*
5. *Create next generation by applying the algorithm on these 20 chromosomes.*
6. *While 20 points are different from previous generation Do 2-4 Else stop.*

Give me such table and graph at the first generation, at two intermediate generations, and at the final generation.

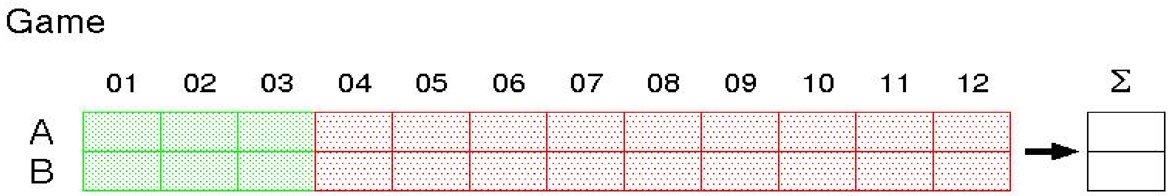
Vi. Iterated Prisonner's Dillenmma

A two player's game like Paper Stone Scissors

Rewards:

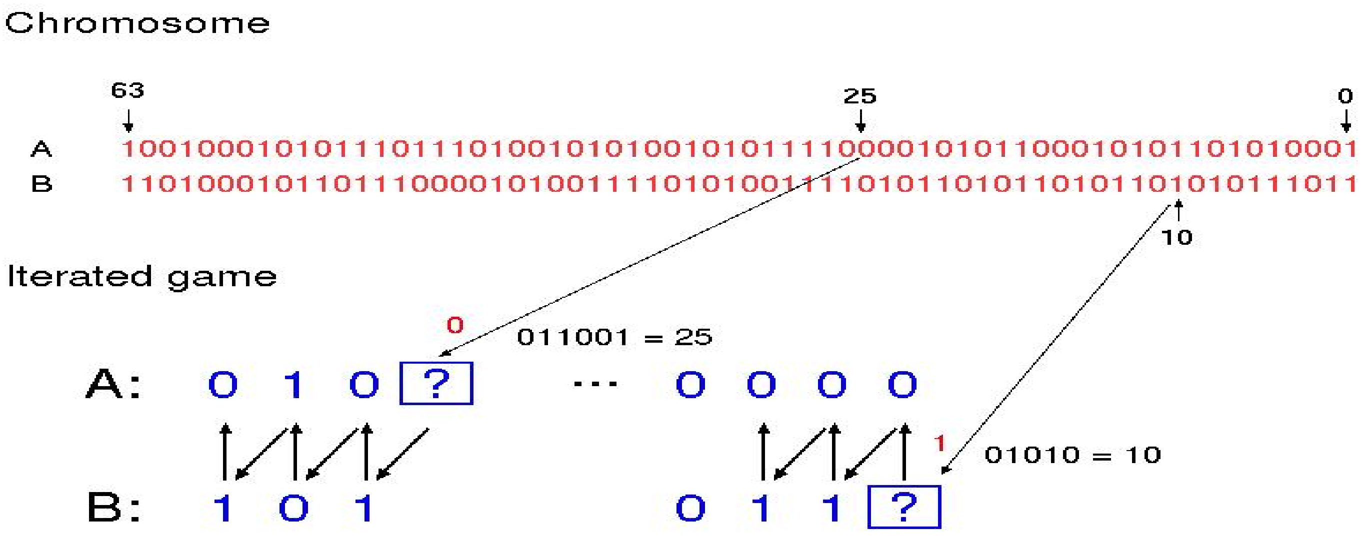
when		→	each will get	
A	B		A	B
1	1		1	1
1	0		5	0
0	1		0	5
0	0		3	3

An iterated game



Try: Random vs Random and Best vs Random, Always-1, Always-0, Tit-for-tat

Chromosome as a game strategy



Next action dependent on chromosome