A slide show of our lecture note

An Introduction of **Bio-inspired Computation**

> **Akira Imada Brest State Technical University Akira Imada, Professor**

Last modified on 14 November 2020



An Image of DNA Spiral





Simplification

adenine ... cytosine ... guanine ... thymine

Our virtual chromosome

Uniform crossover with random number

rnd = 00101110100001011001001110010010010001







Mutation

Truncate Selection

To select parents Create 2 random numbers from 1 to N/2

01	Ó	Ó	Ó	Ċ	Ó	0	0	Ó	Ó	Ó	Ó	Ó	Ó	0	0	0	O	Ó	4
02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
03	Ó	Ó	Ó	Ó	C)	0	0	0	Ó	Ó	Ó	Ó	Ó	0	0	0	0	C	4
04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	K
05	Ó	O	Ó	0	Ó	0		0	O	O	Ó	0	Ó	0	0	0	0	O	1
06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
07	Ó	O	Ó	0	Ö	0	0	0	O	0	Ó	Ó	Ó	0	0	0	0	O	4
08	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
09	0	O	0	0	0	0	0	0	O	O	0	0	0	0	0	0	0	O	4
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6
11	0	0	0	0	0	0	0	0	0	0	0	0	Ö	0	0	0	0	0	4
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
								_			_		_						_
13	Ó	Ó	Ó	Ó	Ó	0	0	Ó	Ó	Ó	Ó	Ó	Ó	O	0		0	Ó	4
13 14	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
13 14 15	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	
13 14 15 16	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	
13 14 15 16 17	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	
13 14 15 16 17 18	0000000	000000	000000	0000000	000000	0000000	0000000	0000000	0000000	0000000	000000	000000	0000000	000000	000000	000000	0000000	0000000	
13 14 15 16 17 18 19	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000	
13 14 15 16 17 18 19 20	000000000	000000000	000000000	000000000	000000000	000000000	000000000	000000000	000000000	000000000	000000000	000000000	000000000	000000000	000000000	000000000	000000000	000000000	
13 14 15 16 17 18 9 20 17	000000000	000000000	000000000	000000000	000000000	000000000	000000000	0000000000	0000000000	0000000000	000000000	0000000000	000000000	000000000	000000000	0000000000	000000000	0000000000	
13 14 15 16 17 18 19 20 17 18	00000000000	00000000000	00000000000	00000000000	0000000000	00000000000	00000000000	00000000000	00000000000	00000000000	00000000000	00000000000	00000000000	0000000000	00000000000	00000000000	00000000000	00000000000	
13 14 15 16 17 18 9 20 17 18 19	000000000000	00000000000	00000000000	00000000000	00000000000	00000000000	000000000000	000000000000	000000000000	000000000000	00000000000	00000000000	00000000000	00000000000	00000000000	000000000000	00000000000	000000000000	
13 14 15 16 17 18 9 20 17 18 9 20	00000000000000	0000000000000	0000000000000	0000000000000	0000000000000	00000000000000	0000000000000	000000000000000000000000000000000000000	00000000000000	00000000000000	0000000000000	0000000000000	0000000000000	0000000000000	0000000000000	000000000000000000000000000000000000000	00000000000000	000000000000000000000000000000000000000	



Summary - Special Terms from Biology

- Chromosome
- Gene
- Crossover
- Mutation
- Fitness (= how good is the chromosome)
- Population
- Generation



II. Toy problems

(1) 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.

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:

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

- **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.

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



How to show the results Fitness vs. generation



(2) Test functions

High-dimensional Test Function

- Problem is minimize or maximize a n-dimensional 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
 - $y = x_1^2 + x_1^2 + x_1^2 + \dots + x_{20}^2$

Minimize У with chromosomes such as (0.54, -0.72, -0.59, $\begin{array}{cccccc} \mathbf{I} & \mathbf{I} & \mathbf{I} \\ \mathbf{X}_1 & \mathbf{X}_2 & \mathbf{X}_3 \end{array}$

Evolve the population till fitness no change

Hints





- Evolve the population till fitness no change

• Create a population of 20 such chromosomes at random + 20 points (Smaller the fitness, the better the chromosome.)

• Plot 20 points of (x,y) in the first, an intermidiate and final generation

y

п

It's 3-D version $z = x^2 + y^2$







 $y = x^2$

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!





Rastrigin - It's 2-D version $y = 3 + x^2 - 3\cos(2\pi x)$







(3) Lucky Dog



A dog in the gridworld (0,0)-(1000,1000) with sausage at (200,800) Dog starts from (500,500) looking for the sausage.



What is Lucky dog?

Evolution of dogs

Fitness vs. Generation

&

The best route

Frid Townerie

Sone of propulation - 300; Size of decomposition - 2000; Multiple permanent - 2020; Multi-permit considerer with 4 provisioner tandom permit from the relation mode.



Lucky dog The best route



Exercise 4 (0,0)-(1000,1000) at (200,800) & (800,800).

Observe what will happen when two sausages exist in the gridworld

Practice

10 dogs start from (5,5) looking for either of 2 sausages.

Create 10 random chromosomes with 8 genes 1,2,3 or 4 to determine the route of each of these 10 dogs.

Original fitness fi is

8 - {Manhattan distance between the final location of the dog and the nearest sausge}.

Calculate Shared Fitnes with $\sigma = 2$ Fi

E.g.

					fi	Fi
(3,2 (1,1	,3, ⁻ ,2,	1,4 1,3	1.3 3.4	3.2.1) I.1.3)		
٠		٠	٠			
•	••	٠	٠	•		



(0,0)

Then evolve these 10 chromosomes till bothe sausages are found!

Show graph of Fi vs. generation and routes of best dog at the begining, intermediate and final!





III. What if we have multiple solutions?



1. Fitness Sharing

(1) Lucky dog with two sausages





(Natalia Vyshinskaya 2013)

A dog lookling for two sausages



- A dog in the gridworld (0,0)-(1000,1000) with sausage at (200,800) and (800,200) Dog starts from (500,500) looking for the sausages.
 - (From the student work by Belous Sophia in 2015)

Two algorithms to get all the solutions 1. Fitness sharing

 $F_s(i) =$

$s(d_{ij}) = \begin{cases} 1 - (d_{ij}/\sigma_{\text{share}})^{\alpha} & \text{if } d_{ij} < \sigma_{\text{share}} \\ 0 & \text{otherwise} \end{cases}$

$$= rac{F(i)}{\displaystyle{\sum_{j=1}^{\mu} s(d_{ij})}}$$

where

Lucky dog fitnes


Now assume dog-i has three other dogs with his territory. The Manhattan distance to each of three dogs are 3,5, and 8 respectively.







Then shared fitness of dog-i **is:** Fi = (600-20)/(1+0.7+0.5+0.2)





When 6 other dogs around dog-A



When 4 other dogs around dog-B



When two other dogs near dog-D





(2) Test function with more than one solution

2-D function minimization - What happens? $y = \sin^6(5\pi x)$



An example of 2D 2 peaks function

Calculate shared fitness of 3 points!



Calculate shared fitness of 4 points below!



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

Exercise 6 1. Minimize

in the following way! (1) Represent value of x by a 10-bit binary chromosome. (3) Evolve this population till fitness dosen't change. 2. Show

(1) the graph of fitness vs generation.

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

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

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

(3) 3-D version of Schwefel's function Exercise 7 1. Minimize

in the following way!

(1) Represent value of x by a 10-bit binary chromosome. (3) Evolve this population till fitness dosen't change.

2. Show

(1) the graph of fitness vs generation.

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

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

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

3-D graph of Schwefel function as a hint



(1) 20-Dimensional Schwefel's function 1. Minimize y in the following way! Exercise 5 $y = x_1 \sin(|x_1|) + x_2 \sin(|x_2|) + \dots + x_{20} \sin(|x_{20}|)$

(3) Evolve this population till fitness dosen't change. 2. Show

(1) the graph of fitness vs generation. (3) Evolve this population till fitness dosen't change.

- (1) Represent each of $x_i (i = 1, \dots, 20)$ by a chromosome with 20 genes. (2) Create a population of 20 chromosomes at random, with fitness being y.

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



2. Crowding Algorithm

2. Crowding Algorithm

- 1. Choose two parents, p_1 and p_2 , at random.
- 2. Produce two children, c'_1 and c'_2 .
- 3. Mutate the children yielding c_1 and c_2 , with a crossover. 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

Crowding - two cases of parents & children



Traveling Sales Person Problem (TSP)

E.g. TSP with 4 cities)

Map of the 4 cities



Distance Matrix

	A	В	С	D
A	0.00	4.13	4.34	1.95
в	4.13	0.00	7.35	5.00
C	4.34	7.35	0.00	2.51
D	1.95	5.00	2.51	0.00

All the possible routes A-B-C-D-A => 4.13 + 7.35 + 2.51 + 1.95 = 15.94A-B-D-C-A => 4.13 + 5.00 + 2.51 + 4.34 = 15.98A-B-C-D-A => 4.34 + 7.35 + 5.00 + 1.95 = 18.64

One-point crossover, e.g., wouldn't work

- tour chromosome
- A-C-B-F-G-D-E (1326745)
- F-B-A-G-D-C-E (6 2 1 7 4 3 5)
- x 2 1 7 4

(1317435) A-C-A-G-D-C-E twice A & C no B & F

Then how to encode TSP into a chromosome?

Step-1. Set i = 1.

currently visited.

Step-3. Remove the city from the list.

Step-4. Set i = i + 1 and repeat Step-2 to Step-4 while $i \leq n$.

- Step-2. If *i*-th gene is n then n-th city in the list is the city to be

E.g. A travel (A-C-B-F-C-D-E).

Can it be encoded to $(1\ 3\ 2\ 6\ 3\ 4\ 5)? \downarrow$

The answer is no!

TSP on 15 cities on the cicle $(x-5)^{2}+(y-5)^{2} = 25$

15 cities rondomly located on the circle (x-5)²+(y-5)²=25
 An example of result ... By Kolesnikov Dmitry (2016)



		Dist	ance n	natrix																				Fit	ness	vs g
٥	¢ Z	\$ A	0 B	¢ C	¢ D	\$ E	0 F	¢ G	¢Н	¢1	\$ J	¢К	¢L	≎M	0 N	70	Î	T				T	T			TF
Z	0																		-	4						
A	2.3842	0														60					-	++	+			
8	4.8078	2.5786	0																							
С	8.8801	9.7203	9.9971	0												50		Ħ			-	11	T	1		T
D	2.8005	0.4309	2.1598	9.8125	0																					
Е	9.8659	9.1923	7.8662	5.9855	9.0141	0										ssa 40					1		T	V		
F	8.8178	7.4390	5.4642	8.2428	7.1441	3.2143	0									Fitn	1	Len	gth o	fthe	circ	e	-		4	
G	9.3267	9.9178	9.9123	1.0852	9.9637	5.0808	7.5797	0								30	1						T			
Н	8.2722	6.6940	4.5520	8.7931	6.3676	4.1935	1.0529	8.2244	0							20										
1.	9.9998	9.7008	8.7467	4.6379	9.5871	1.5875	4.6768	3.6490	5.5815	D											T					
J	9.9957	9.7770	8.9049	4.3369	9.6774	1.9191	4.9718	3.3334	5.8577	0.3367	0					10					_	Ш	4			
К	0.9065	3.2548	5.5829	8.4267	3.6592	9.9732	9.2091	8.9612	8.7475	9.9627	9.9281	0														
L	4.3334	2.0597	0.5332	9.9956	1.6361	8.1843	5.9030	9.9686	5.0203	8.9927	9.1349	5.1326	0			0	L					Ш				Ш.
М	2.4616	4.7015	6.8183	7.4750	5.0774	9.9640	9.7075	8.1517	9.4008	9.7032	9.6169	1.5728	6.4185	0			0	1	5 3		9 9	1	0 4	.,e,	P #	19 50
N	9.8982	9.2736	7,9951	5.8147	9.1037	0.2115	3.4139	4.8975	4.3846	1.3783	1.7110	9.9864	8.3040	9.9439	0											Ger
		·																								





Starting with city Z salesperson should visit all other 24 cities once and only once and return to Z again. Look for minimum length tour!

(6,6)



(0,0)

IV. Iterated Prisoner's Dilemma (IPD)

What is prisoner's dilemma?

- vice versa.
- If both confess, then both will get 4 year in jail.
- If both do not, each get 2 years.

Reward Matrix

Α\B	cooperate	defect
cooperate	1\1	0 \ 10
defect	10\0	3 \ 3

If A confesses and B does not, A will reliesed and B will get 5 years in jail, and



Iterated Prisoner's Dilemma

A random game







How to use chromosome?

Next action dependent on chromosome



How to evolve players?

- Create a population of (E.g. 40) 64-bit binary chromosomes.
- Each chromosome choose T (E.g. 5) other chromosomes to play a game from one to the next.
- Fitness is the number of wins (or the total rewards) at the end of T games.
- Sort the chromosomes according to fitness.
- Select two chromosomes from the upper half of the population to create child.
- Mutate the child.
- Repeat the above procedures to create a next generation till no fitness change for a while.





Try the strongest player with:

- Tit for Tat ... next action is always same as the opponent
- Always Opposite
- Always play with "1"
- Always play with "0"

V. Dimension Reduction by Evolution



Samon Mapping

	3 points in 3D		D	istance	Matrix			Normalized								
	pida chigin		\wedge	в	С	0			\wedge	в	C	0				
A:	(0.328, 0.819, 0.118)	А	0.000	1.245	1.481	0.890		А	0.000	0.846	1.000	0.601				
B:	(0.129, -0.256, 0.713)	в		0.000	1.126	0.769	6	в		0.000	0.760	0.519				
C:	(0.2770.584, -0.354)	C			0.000	0.737	+	C			0.000	0.498				
O:	(0.000, 0.000, 0.000)	0				0.000		0				0.000				
rand	dom 5 points in 2D															
	plus Origin		\times	Y	z	0			×	Y	z	0				
X:	(0.514, -0.223)	×	0.000	1.826	0.632	0.560		×	0.000	1.000	0.346	0.307				
Y:	(-0.861, 0.979)	Y		0.000	1.348	1.304		Y		0.000	0.738	0.714				
Z:	(-0.113, -0.142)	z			0.000	0.182	-	z			0.000	0.273				
O:	(0.000, 0.000)	0				0.000		0				0.000				

chromosome:

(0.514 -0.223 -0.861 0.979 -0.113 -0.142 0.000 0.000) fitness

- 0.846

(0.846 - 1.000) + (1.000 - 0.346) + (0.601 - 0.307) + (0.760 - 0.738) + (0.519 - 0.714) + (0.498 - 0.273)

Chromosome for Samon Mapping



- 1. Assume N points are given in the n-D space.
- 2. Calculate distance matrix R ($N \times N$) whose i-j element is the Euclidean distance between the *i*-th and *j*-th point.
- 3. Also think of a tentative N points in the 2-D space that are located at random at the beginning.
- 4. The distance matrix Q is calculated in the same way as R.
- 5. Then the error matrix P = R Q is defined.
- 6. Search for the locations of N points in the 2-D space that minimizes the sum of element P.

Algorithm



Coordinate 3D points

0 1 2	0,365	-0,073	0,928
1 2	0,558	-0.154	
2	0.500		0,816
	0,593	-0,152	-0,791
3	-0,19	-0,027	0,981
4	0,636	0,156	-0,756
5	-0,661	-0,172	-0,73
6	-0,283	0,103	-0,954
7	0,74	0,29	0,607
8	-0,51	-0,062	0,858
9	0,524	-0,129	0,842
10	-0,362	-0,049	0,931
11	0,645	0,04	0,763
12	0,097	0,048	-0,994
13	0,483	0,166	-0,86
14	-0,564	-0,005	-0,826
15	-0,488	0,228	-0,843
16	0,471	0,055	-0,88
17	0,246	0,032	-0,969
18	0,943	-0,185	-0,277
19	0,529	-0,093	-0,844
20	0	0	0

Then the original 3D distance matrix and its normalization were calculated.

Matrix distance from points in 3D

100	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	0	0,238	1,736	0,559	1,721	1,95	1,998	0,613	0,88	0,19	0,73	0,344	1,945	1,808	1,986	1,988	1,816	1,904	1,342	1,78
1	0,238	0	1,607	0,777	1,604	1,97	1,976	0,524	1,07	0,05	0,93	0,219	1,878	1,708	1,994	1,997	1,711	1,821	1,159	1,661
2	1,736	1,607	0	1,942	0,313	1,26	0,926	1,473	1,99	1,64	1,97	1,567	0,571	0,344	1,167	1,145	0,256	0,431	0,622	0,102
3	0,559	0,777	1,942	0	1,932	1,78	1,942	1,052	0,34	0,74	0,18	0,865	1,998	1,97	1.845	1,865	1,977	1,999	1,701	1,963
4	1,721	1,604	0,313	1,932	0	1,34	0,941	1,373	1,99	1,63	1,97	1,524	0,598	0,185	1,213	1,129	0,23	0,461	0,663	0,285
5	1,953	1,969	1,256	1,781	1,338	0	0,519	1,991	1,6	1,97	1,69	1,995	0,833	1,2	0,215	0,451	1,164	0,96	1,567	1,198
6	1,998	1,976	0,926	1,942	0,941	0,52	0	1,875	1,83	1,98	1,89	1,953	0,386	0,774	0,328	0,264	0,759	0,534	1,429	0,842
7	0,613	0,524	1,473	1,052	1,373	1,99	1,875	0	1,32	0,53	1,2	0,311	1,742	1,494	1,96	1,9	1,529	1,671	1,024	1,515
8	0,877	1,072	1,986	0,344	1,991	1,6	1,833	1,323	0	1,04	0,17	1,163	1,952	1,997	1,686	1,725	1,999	1,979	1,848	1,994
9	0,189	0,05	1,635	0,735	1,627	1,97	1,982	0,527	1,04	0	0,89	0,222	1,893	1,728	1,995	1,997	1,733	1,839	1,197	1,686
10	0,727	0,933	1,972	0,181	1,971	1,69	1,892	1,198	0,17	0,89	0	1,024	1,981	1,992	1,768	1,799	1,996	1,996	1,784	1,986
11	0,344	0,219	1,567	0,865	1,524	2	1,953	0,311	1,16	0,22	1,02	0	1,841	1,636	1,997	1,974	1,653	1,778	1,106	1,617
12	1,945	1,878	0,571	1,998	0,598	0,83	0,386	1,742	1,95	1,89	1,98	1,841	0	0,424	0,685	0,531	0,39	0,151	1,132	0,478
13	1,808	1,708	0,344	1,97	0,185	1,2	0,774	1,494	2	1,73	1,99	1,636	0,424	0	1,061	0,972	0,114	0,293	0,821	0,264
14	1,986	1,994	1,167	1,845	1,213	0,22	0,328	1,96	1,69	2	1,77	1,997	0,685	1,061	0	0,245	1,038	0,824	1,614	1,097
15	1,988	1,997	1,146	1,866	1,129	0,45	0,264	1,9	1,73	2	1,8	1,974	0,631	0,972	0,246	0	0,975	0,77	1,593	1,066
16	1,816	1,711	0,256	1,977	0,23	1,16	0,759	1,529	2	1,73	2	1,653	0,39	0,114	1,038	0,975	0	0,243	0,802	0,163
17	1,904	1,821	0.431	1,999	0,461	0,96	0,534	1,671	1,98	1,84	2	1,778	0,151	0,293	0,824	0,77	0,243	0	1,005	0,334
18	1,342	1,159	0,622	1,701	0,663	1,67	1,429	1,024	1,85	1,2	1,78	1,106	1,132	0,821	1,614	1,593	0,802	1,005	0	0,707
19	1,78	1,661	0,102	1,963	0,285	1,2	0,842	1,515	1,99	1,69	1,99	1,617	0,478	0,264	1,097	1,065	0,163	0,334	0,707	C
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Dimension reduction from 3D sphere to 2D circle

At the begining, a target 3D spere was created and select 20 points randomly from the surface.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	0,119	0,868	0,28	0,861	0,98	0,999	0,307	0,44	0,1	0,36	0,172	0,973	0,904	0,993	0,994	0,908	0,952	0,671	0,89	0,5
1	0,119	0	0,804	0,388	0,802	0,99	0,988	0,252	0,54	0,03	0,47	0,109	0,939	0,854	0,997	0,999	0,856	0,911	0,58	0,831	0,5
2	0,868	0,804	0	0,971	0,155	0,63	0,463	0,737	0,99	0,82	0,99	0,784	0,286	0,172	0,584	0,573	0,128	0,215	0,311	0,051	0,5
3	0,28	0,388	0,971	0	0,965	0,89	0,971	0,526	0,17	0,37	0,09	0,433	0,999	0,985	0,923	0,933	0,989	1	0,851	0,982	0,5
4	0,861	0,802	0,156	0,966	0	0,67	0,471	0,687	1	0,81	0,99	0,762	0,299	0,093	0,607	0,565	0,115	0,231	0,332	0,142	0,5
5	0,977	0,985	0,628	0,891	0,669	0	0,259	0,996	0,8	0,99	0,85	0,998	0,417	0,6	0,108	0,225	0,582	0,48	0,834	0,599	0,5
6	0,999	0,988	0,463	0,971	0,471	0,26	0	0,938	0,92	0,99	0,95	0,977	0,193	0,387	0,164	0,132	0,379	0,267	0,715	0,421	0,5
7	0,307	0,262	0,737	0,526	0,687	1	0,938	0	0,65	0,26	0,6	0,155	0,871	0,747	0,98	0,95	0,765	0,836	0,512	0,758	0,5
8	0,439	0,536	0,993	0,172	0,995	0,8	0,917	0,652	0	0,52	0,08	0,581	0,976	0,999	0,843	0,863	1	0,99	0,924	0,997	0,5
9	0,095	0,025	0,818	0,367	0,814	0,99	0,991	0,264	0,52	0	0,45	0,111	0,947	0,864	0,998	0,999	0,867	0,92	0,599	0,843	0,5
10	0,364	0,467	0,985	0,09	0,985	0,85	0,946	0,599	0,08	0,45	0	0,512	0,991	0,996	0,884	0,9	0,998	0,998	0,892	0,993	0,5
11	0,172	0,109	0,784	0,433	0,762	1	0,977	0,155	0,58	0,11	0,51	0	0,921	0,818	0,999	0,987	0,827	0,889	0,553	0,809	0,5
12	0,973	0,939	0,285	0,999	0,299	0,42	0,193	0,871	0,98	0,95	0,99	0,921	0	0,212	0,343	0,315	0,195	0,076	0,566	0,239	0,5
13	0,904	0,854	0,172	0,985	0,093	0,6	0,387	0,747	1	0,86	1	0,818	0,212	0	0,531	0,486	0,057	0,147	0,411	0,132	0,5
14	0,993	0,997	0,584	0,923	0,607	0,11	0,164	0,98	0,84	1	0,88	0,999	0,343	0,531	0	0,123	0,519	0,412	0,807	0,549	0,5
15	0,994	0,999	0,573	0,933	0,565	0,23	0,132	0,95	0,85	1	0,9	0,987	0,315	0,486	0,123	0	0,488	0,385	0,796	0,533	0,5
16	0,908	0,856	0,128	0,989	0,115	0,58	0,379	0,765	1	0,87	1	0,827	0,195	0,057	0,519	0,488	0	0,121	0,401	0,081	0,5
17	0,952	0,911	0,215	1	0,231	0,48	0,267	0,836	0,99	0,92	1	0,889	0,076	0,147	0,412	0,385	0.121	0	0,503	0,167	0,5
18	0,671	0,58	0,311	0,851	0,332	0,83	0,715	0,512	0,92	0,6	0,89	0,553	0,566	0,411	0,807	0,796	0,401	0,503	0	0,354	0,5
19	0,89	0,831	0,051	0,982	0,142	0,6	0,421	0,758	1	0,84	0,99	0,809	0,239	0,132	0,549	0,533	0,081	0,167	0,354	0	0,5
20	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0




from surface of hyper sphere in 49² = 2401 dimensional space

A visualization of 4D data



Ŧ

x_1	x_2	x_3	x_4	class
5.1	3.5	1.4	0.2	1 (Setosa)
4.9	3.0	1.4	0.2	1 (Setosa)
4.7	3.2	1.3	0.2	1 (Setosa)
• • •	• • •	• • •	•••	
7.0	3.2	4.1	1.4	2 (Versicolor)
6.4	3.2	4.5	1.5	2 (Versicolor)
6.9	3.1	4.9	1.5	2 (Versicolor)
• • •	• • •	• • •	•••	
5.8	2.7	5.1	1.9	3 (Virginica)
7.1	3.0	5.9	2.1	3 (Virginica)
6.3	2.9	5.6	1.8	3 (Virginica)



VI. Sorting Network

1. What is a sorting network? E.g. Bubble sort - 16 items



The number of comparison of the items in this case is:

 $15 + 14 + 13 + \cdots + 2 + 1 = 120$

Then what would be a minimal number of comparison?

- \star 65 comparisons Bose and Nelson (1962).
- \star 63 by Batcher and by Floyd and Knuth (1964).
- * 62 by Shapiro (1969)
- * 60 by Green (1969)

Ex. Sorting algorithm by Knuth et. al (1964) 63 Comparisons



2. A possible sorting algorithm by evolution

- E.g. An integer chromosome to sort 16 items
- with 140 genes each takes an integer from 1 to 16 permitting overlaps
 - $(12\ 01\ 05\ 04\ 16\ 12\ 04\ 14\ 01\ 02\ 06\ \dots,\ 07\ 15\ 08\ 10)$
 - Which compares
 - 12 <=> 01
 - 16 <=> 12
 - 08 <=> 10
 - 05 <=> 04
 -

What should be fitness evaluation?

- FOR i=1 TO n FOR j=1 TO n IF x(i) > x(j) THEN replace x(i) with x(j) and fitness++ ELSE do nothing
 - For example, assuming n = 5

 - $05\ 01\ 03\ 02\ 04 \Rightarrow \text{fitness} = 0\ +\ 3\ +\ 1\ +\ 1\ =\ 5$
 - $01\ 02\ 03\ 05\ 04 \Rightarrow \text{fitness} = 4 + 3 + 2 + 0 = 9$

fitness = 0

- $01\ 03\ 02\ 05\ 04 \Rightarrow \text{fitness} = 4 + 2 + 0 + 0 = 6$
- $04\ 02\ 05\ 01\ 03 \Rightarrow \text{fitness} = 1 + 2 + 0 + 1 = 4$

An algorithm

- 1. Create N a set of random order integers from 1 to 16. 2. Create a population of, say, 20 chromosomes at random. 3. Evaluate fitness of each chromosome ciit-2020-remote o N. 4. Create the next generation by selection, crossover and mutation.
- 5. Repeat from 3. to 4. untill maximum fitness saturates.

Show (i) the best chromosome in the final generation; (ii) The result of applying it to N with the original order (before & after); and (iii) its diagram by ommiting those identical comparisons.



chromosome: [08 14 04 05 15 07 05 01 07 07 10 14 02 05 08 10 08 15 10 00 02 07 07 04 03 05 04 07 08 10 08 11 11 10 11 09 03 03 09 11 15 02 07 03 06 08 07 00 11 03 12 11 02 10 02 15 12 01 06 10 08 12 00 10 01 12 15 05 01 12 12 08 06 04 06 05 06 11 14 08 00 11 13 10 05 13 08 15 01 02 03 02 02 09 03 04 15 10 08 11 01 15 01 07 10 14 10 12 05 11 03 07 04 06 10 07 10 12 01 06 06 04 11 08 06 10 12 01 07 05 04 07 10 06 03 07 12 04 04 09 10 04 04 08 11 04 15 06 07 11]



An example 3. ... By Supruniuk Darya Her chromosome, and its graphic

after [05 03 00 01 04 02 09 08 07 06 10 12 15 11 14 13] fitness = 100 real compares = 73



Generation	Best chromosome	Origin After applying	Comparisons	Fitness
0	0209145701135791161585520121161042519 1341515812112181513141414450211671153 7915102115251785132122310115128110134 1513612131113128156011812611153151434 7144101394120514410741114010330331154 1411514376212719151191569812411701490 3131661446111611123474136118911293802 1014159624110111110414141514118153238 148912412253303726101241110311	1514024126108153911713	73	81
50	1010113128144581205155111594512572513 9404701321349121014134828451710705102 7515323549129412814615913144513813034 1413131316153313131214121415515101212 4891491496101399136754313011011310610 2152127150147124781119945151579911910 4121011714071131910123131113121406558 2913493151244121343615512175471052371 4136571401331008350641458	0135426789101211151314	76	113
100	1210013128144581201315515159451257251 3950450132131394012134828651707441010 7415310354989129286141599144713121303 4615131356713991314414155151012124891 4169610139513675431301142310610215313 7158147121278111994131511579911910612 1011712279131910123131113121406518211 1259315124412130367512135431051237613 6571421331000350641558	0123456789101112131415	80	120

Good work! But what was indicated was not by using interger chromosomes but binary ones!

Or, see another example



NEL	fitness = 101
before	09 15 08 05 02 07 06 11 14 03 13 10 00 01 04 1
after	00 01 04 02 03 07 08 05 11 15 10 06 12 09 14 1
NH2	
hefore	01 00 09 01 00 15 02 10 05 11 12 14 08 07 03 1
after	01 00 06 03 02 05 04 09 14 10 08 07 12 11 15 1
Nua	fitness = 107
before	00 14 10 04 01 12 15 07 06 02 08 09 11 05 03 1
after	00 01 04 02 03 06 05 07 10 14 08 11 12 09 15 1
Nat	The second s
before	01 10 13 15 11 08 04 09 06 03 02 00 14 05 07 1
after	01.03 00 02 04 08 05 07 10 15 11 06 12 09 14 1
NUS	fitness = 101
before	06 02 08 15 09 11 14 00 13 04 07 05 10 12 01 0
after	00.02 04 01 03 08 05 06 10 15 11 09 12 07 14 1

(From the student work by Radchuk Aliona in 2015)



A more biological diploydy chromosoms Hillis's proposal (1992)

(1001 1000 1010 1101 1110 0100 1110 0011) (1001 1000 1010 1101 1110 0100 1110 0011) (1001 1000 1010 1101 1110 0100 1110 0011) (1001 1000 1010 1101 1110 0100 1110 0011) (1001 1000 1010 1101 1110 0100 1110 0011) (1001 1000 1015 1101 1110 0100 1110 0011) (1001-1005 1018 1101 1118 8108 1118 8011) (1004 1000 1010 1101 1110 0100 1110 0011) (1001 1000 1010 1101 1110 0100 1110 0011) (1001 1000 1010 1101 1110 0100 1110 0011) (1001 1000 1010 1101 1110 0100 1110 0011) (1001 1000 1010 1101 1110 0100 1110 0011) (1001 1000 1010 1101 1110 0100 1110 0011) (1001 1000 1010 1101 1110 0100 1110 0011) (1001 1000 1010 1101 1110 0100 1110 0011) (1001 1000 1010 1101 1110 0100 1110 0011)

(1001 1000 1010 1101 1110 0100 1110 0011) (1001 1000 1010 1101 1110 0100 1110 0011)

(1001 1000 1010 1101 1110 0100 1110 0011) (1001 1000 1010 1101 1110 0100 1110 0011)

15 pairs of 32-bit chromosomes Each chromosome consists of eight 4-bit strings called codons Each codon is an integer between 0 and 15 60 comparisons: world record, so far!

(no proof this is minimum, though)

(1001 1000 1010 1101 1110 0100 1110 0011) (1001 1000 1010 1101 1110 0100 1110 0011)

(1001 1000 1010 1101 1110 0100 1110 0011) (1001 3000 1010 1101 1110 0100 1130 0011)

(1001 1000 1010 1101 1110 0100 1110 0011) (1001 1000 1010 1101 1110 0100 1110 0011)

(1001 1000 1010 1101 1110 0100 1110 0011) (1001 1000 1010 1101 1110 0100 1110 0011)

(1001 1000 1010 1101 1110 0100 1110 0011) (1001 1000 1010 1101 1110 0100 1110 0011)

How a pair of chromosomes each with 480-bit

sorts 16 items?

			_	-			_	-				
	. 0	• •		12			03	04	- 0	sio	***	
00	01	02	0.0	04	0.5	06	07	08	00	10	11	11
C	1	ĸ	~	D	1	M	34	25	P	J	E	0
~		14		100		2.0		101		·)	12	-
.	1.1	~	~	P		241	1	(19		ñ	- mel	hart
C	D	ĸ	~	F	T.	M	14	C	P	J	R	Τ.
		-			19		-4	6.8.8	01	1)	der	NO B
C	D	ĸ	A	F	1	M	14	C.	P	J	E	C
1	1	-		-	1.1		.*	104	0.	· · ·		-
0	D	P.	~	K		241	12	(04		5-	15	- C
C	D	F	~	F	т	M	11	C	P	J	E	C
		- T		74.5			+	607	01	·)	exci	haint
C	D	$\mathbf{F}_{\mathbf{L}}$	A	P	1	H	M	C	P	-3	E	C
-												
	1.5		^		. *			(01	01	5	exci	nam'
C	A	F	D	F	I	H	M	C	P	J	E	C
							E					
							1				hop	HOT.
A	ES.	C	D	E	F	G	34	1	J	ĸ	L	N

0011000100110001001111010011 00101101 0101 1010 0011 1010 0011 1000 0011 1001 0110



of identical pair - ? fitness - ?







Excercise - Evolution of diploydy chromosomes

 (1) Create your target random order string of 16 alphabets from A to P
(2) Observe evolution from a poulation of randome 40 diploydy chromosomes untill maximum fitness does not change

What should be shown are:

(1) Original string with 16 alphabets in random order; (2) The result of string after sort by the final best chromosome; (3) The best chromosome in the final generation; (4) The diagram of comparisons of the best chromosome; after removing multiple comparisons and self-comparisons; (5) (i) What is its fitness value? And (ii) how many real comparisons it includes?

(6) Three graphs during evolution



(continued)

The best chromosome can sort other random string?

Can the best chromosome, whose fitness was how the original one target string of 16 letters, will be able to sort another new random string correctly?



Excercise - Evolution of diploydy chromosomes to sort multiple strings

(1) Create two target random order 16 alphabets from A to P. Fitness this time is the sum of how it sorts these two strings. (2) Observe evolution from a poulation of randome 40 diploydy chromosomes untill maximum fitness does not change. (3) Apply the best chromosome to 10 random strings including the original two!

Show a set of 10 strings with its BEFORE & AFTER!





IV. What if more than one fitness function exist?

Parete Optimal Solution

If X is better than Y for all the Fitnesses then it is said that X dominates Y.

When a solution is not dominated by any others it is called Non Dominated Solution

Parete Optimal Solution

or

Rank - When we try evolution for Parete optimum

- Order according to
- the number of how many others in the population it dominates.

E.g. who dominates whom and how rank is counted.

	test-1	test-2	test-3	test-4	test-5	dominated by	dominates	rank
Α	9	9	9	8	7	0	3	1
в	5	4	5	3	6	1	2	2
\mathbf{C}	3	3	4	2	3	2	1	3
D	2	2	3	1	2	3	0	4
Е	1	1	1	1	9	0	0	4



Algorithm

- 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.
- 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.



Excercise

- sponding to 0 and 6, respectively.

- 4. Plot these 20 points on each of two graphs (e.g., red and blue color).

Try it with two fitness 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 $(0000 \cdots 00)$ and $(1111 \cdots 11)$ being corre-

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).

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.





FIRST					
0111110111	x=2.950147	y1=1.102192	y2=0.902779	rank=8	
1110000010	x=5.266862	y1=1.604940	y2=10.67238	rank=0	18
1101111110	x=5.243402	y1=1.546048	y2=10.51965	rank=1	
1000001100	x=3.073314	y1=0.858747	y2=1.152002	rank=8	16
1011101000	x=4.363636	y1=0.132231	y2=5.586777	rank=3	
1101111101	x=5.237537	y1=1.531497	y2=10.48164	rank=2	14
0000110101	x=0.310850	y1=13.60982	y2=2.853226	rank=3	-
000000101	x=0.029326	y1=15.76625	y2=3.883558	rank=0	12
0111101011	x=2.879765	y1=1.254926	y2=0.773987	rank=8	10
0110100110	x=2.475073	y1=2.325401	y2=0.225695	rank=5	10
0000011011	x=0.158358	y1=14.75821	y2=3.391646	rank=1	8
1010001101	x=3.829912	y1=0.028930	y2=3.348578	rank=6	
0111001000	x=2.674487	y1=1.756985	y2=0.454932	rank=5	6
0000101101	x=0.263930	y1=13.95822	y2=3.013940	rank=2	1.4 50
0111000101	x=2.656891	y1=1.803940	y2=0.431506	rank=5	4
0111101000	x=2.862170	y1=1.294657	y2=0.743337	rank=8	1978
1000100100	x=3.214076	y1=0.617676	y2=1.473981	rank=8	2
1011000000	x=4.129032	y1=0.016649	y2=4.532778	rank=4	
0001011110	x=0.551320	y1=11.89339	y2=2.098675	rank=4	
0100101011	x=1.753666	y1=5.046018	y2=0.060681	rank=5	

		LAST			
0111110111	x=2.950147	y1=1.102192	y2=0.902779	rank=0	
1010101101	x=4.017595	y1=0.000310	y2=4.070691	rank=0	
1010101100	x=4.011730	y1=0.000138	y2=4.047058	rank=1	
1000001100	x=3.073314	y1=0.858747	y2=1.152002	rank=0	
1010101010	x=4.000000	y1=0.000000	y2=4.000000	rank=3	
1010101011	x=4.005865	y1=0.000034	y2=4.023495	rank=2	
0101010101	x=2.000000	y1=4.000000	y2=0.000000	rank=3	
0101010010	x=1.982405	y1=4.070691	y2=0.000310	rank=1	
0111101101	x=2.891496	y1=1.228782	y2=0.794764	rank=0	
0110010001	x=2.351906	y1=2.716213	y2=0.123838	rank=0	
0101010011	x=1.988270	y1=4.047058	y2=0.000138	rank=2	
1010001101	x=3.829912	y1=0.028930	y2=3.348578	rank=0	
0111001010	x=2.686217	y1=1.726026	y2=0.470894	rank=0	
0101000110	x=1.912023	y1=4.359646	y2=0.007740	rank=0	
0110100001	x=2.445748	y1=2.415700	y2=0.198691	rank=0	
0111010111	x=2.762463	y1=1.531497	y2=0.581350	rank=0	
1001011100	x=3.542522	y1=0.209286	y2=2.379374	rank=0	
1010100101	x=3.970674	y1=0.000860	y2=3.883558	rank=0	
0101010110	x=2.005865	y1=3.976574	y2=0.000034	rank=3	
0101011010	x=2.029326	v1=3.883558	v2=0.000860	rank=1	



M. Martin

5

V. Evolving Finite State Machine

What is Finite State Automaton?

Finite State Machine is a machine that makes an action depending on imput and its current state then change state to a new one. For example:

Current State	inputs	actions	New State
Δ	0	a	A
~	1	С	В
P	0	b	D
D	1	a	С
0	0	d	С
	1	b	D
D	0	С	D
	1	d	A

Transition Table



Its diagram

Simplest Example

current state	input	action
Α	empty	go up one
		Or
	A	O/go up

Now let's think of a simple FSA which explors a grid world. This FSA has only one state A. It receives input from a cell in front. All cells are empty. So, input is always 0. Action is always "Go one step in front."



An FSA with 2 inputs 2 states and 4 actions

current state	input	action	new state
200	0	go-front-one-step	Α
A	1	turn-right-and-go-one-step	B
	0	go-front-one-step	С
В	1	do-whatever	B
	0	turn-right-and-go-one-step	A
C	1	do-whatever	С

0/go-front-one-step

1/turn-right-and-go-one-step

0/go-front-one-step

=> 1 (6, 6)(0,0)

input

=> 0



0/turn-left-and-go-one-step

Excersize



- A little more complicated FSA
- FSA which avoids obstacle then return to the original route
- Try to design this FSA with input and actions being same as before but with more states



Representation of FSA with binary chromosome





action	
go-front-one-step	=> 00
turn-right-and-go-one-step	=> 01
turn-right-and-go-one-step	=> 10
do-whatever	=> 11

ate	input	action	new state
	0	go-front-one-step	Α
	1	turn-right-and-go-one-step	В
	0	turn-left-and-go-one-step	Α
	1	do-whatever	В





Excersize

- From random chromosom to FSA
- 1. Create a binary chromosome with 12 genes at random.
- 2. Translate it to state transition table of FSA with 2 states, 2 inputs and 4 outputs.
- 3. Starting from (4,0) run the FSA on gridworld of 7x7 where only one obstacle locates at (3,3).
- 4. Report (1) chromosome, (2) table or diagram of FSA, and (3) trace of FSA.
- 5. Repeat from 1. to 4. above 10 times

Excersize

- 1. Create a population of N random binary chromosome with 12 genes each corresponding to FSA
- 2. Make each of the FSA move 8 steps and record the final location as (X, Y). 3. evolve the population with fitness being |X - 4| + |Y - 7| until fitness becomes
- stable.
- 4. Report evolution by graph of generation vs. fitness

An evolution of FSA





The John Muir trail in 32 x32 troidal grid

「「「「「「」」			83	18	1	H.							Ź											I
H		\square	+	+	-		-					A	_	_			_					_	+	+
H				t	t	ii.																	1	t
1.001.000	813		-	+	-		-	-	-				_	_	-	-	-	_	_	1.1		34		
10000	11	Ħ		t	E															R	_		T	1
H		H	-	-				815					_		_		_		_			-	+	4
Ht		H					60) (1)	119		H			-				-	-					+	+
H	1000	鼦	8 I	88													225					\square	\neg	ļ
Ht	H	H	+	t	t			10	14	8	÷.	H	-		102	112	592	519			Η	+	+	+
		Π	-								11												1	
H+	H	H	+	+	+	-	-	H	1991	11	815	Н	-		-	H		-	-		H	+	+	+
H			-	t				1															1	
H+		H	+	-	110	49		¥8	-	\vdash		Н	-		-		-	-	-		Н	+	+	+
				20																			1	
P	H	1.5		10	⊢	-	-	-	-	\vdash	-		-		-	-	-	-	-		-	+	+	+
	tt	I																					\pm	
HH	1 22		+	⊢	-		-					\square	-					_	_			+	+	-
H				t	t								-										\pm	
\square									223		21.5		110	ii.	-			_	_				4	_
Ht	1					813	12						11		-							+	+	+
			-												_			_					7	

10



A solution using FSM by Jefferson et al. (1992)



Input: 0 => unmarked cell ahead 1 => marked cell ahead

Output: M => move 1 step backward **R** => trun to the right L => turn to the left


Another example of a bigger FSM





An example of grown FSM by evolution





An exampleof quite huge FSM for memory

Giles, Horne and Lin (1995)



(2) Genetic Programming



VI. Growing Structure by Evolution Genetic Programming (GP)



What is LISP?

- LISP is the second-oldest high-level language after FORTLAN
- LISP is developed in 1958 by John McCarthy while kindle he was at the MIT
- McCarthy published its design in a paper in Communications of the ACM in 1960, entitled "Recursive Functions of Symbolic Expressions and Their Computation by Machine, Part I"
- LISP was first implemented by Steve Russell on an IBM 704 computer using punched cards







Genetic Programming (GP) evolves programs in LISP programming language



Program Language LISP can be represented as tree structure

(+ 1 2 (IF (> time 10) 3 4))





- E.g.



Another example







$x_1 + ((x_2 - 3.1) * x_1)$

A crossover



A mutation





Let's apply GP to Neural Network



An Example of neural network by Genetic Programming



A neural network for the symmetry of 40 input units by GP

By F. Gruau and D Whitley "Adding Learning to the Cellular Development of Neural Networks: Evolution and the Baldwin Effect" in Evolutionary Computation (1993)



A neural network for the parity of 21 input units

Also By F. Gruau and D Whitley "Adding Learning to the Cellular Development of Neural Networks: Evolution and the Baldwin Effect" in Evolutionary Computation (1993)



In case you are interested in LISP

Try

macOS **\$ brew install clisp** (See left screen shot)

Windows **\$** scoop install roswell



Welcome to GNU CLISP 2.49 (2010-07-07) <http://clisp.cons.org/>

Copyright (c) Bruno Haible, Michael Stoll 1992, 1993 Copyright (c) Bruno Haible, Marcus Daniels 1994-1997 Copyright (c) Bruno Haible, Pierpaolo Bernardi, Sam Steingold 1998 Copyright (c) Bruno Haible, Sam Steingold 1999-2000 Copyright (c) Sam Steingold, Bruno Haible 2001-2010

[1]>

s-iMac:~ cyra	no\$ cli	sp						
iiii	00000		0	0000000	00000		00000	
IIIII	8	8	8	8	8	0	8	8
\`+'/ I	8		8	8	8		8	8
`-+-' /	8		8	8	00000		8000	00
!'	8		8	8		8	8	
	8	0	8	8	0	8	8	
+	00000		8000000	0008000	00000		8	

Type :h and hit Enter for context help.





Thank you for attending this course!

I've been happy to have this chance of giving you this lecture!