Modelling Complex Systems

Genetic Algorithms

This lecture includes adapted slides of David Sumpter and Yu Liu, and work of Charitini Stavropoulou, Katarzyna Kowalczyk and Emil Åberg

Evolution

- Evolution solves "problems"
- But there is no specific problem needed to be solved, only one general problem: increasing fitness
- We have specific problems

Evolution

- e.g., **eye**

- Large solution space
- Open-ended
- Natural selection (adaptation):
  1. reproduction
  2. mutation
  3. competition (e.g., limited resources)

Genetic Algorithm (GA)

UPPSALA UNIVERSITET

- Large solution space, hard to check every possibility
- Not open-ended (should stop)
- Natural selection in computer:
  1. reproduction?
  2. mutation?
  3. competition?

Genetic Algorithm (GA)

UPPSALA UNIVERSITET

- John Henry Holland, 1970s
- Computer programs that evolve over generations to find (some of) the “fittest” out of a very large number

Basic GA Recipe

UPPSALA
UNIVERSITET

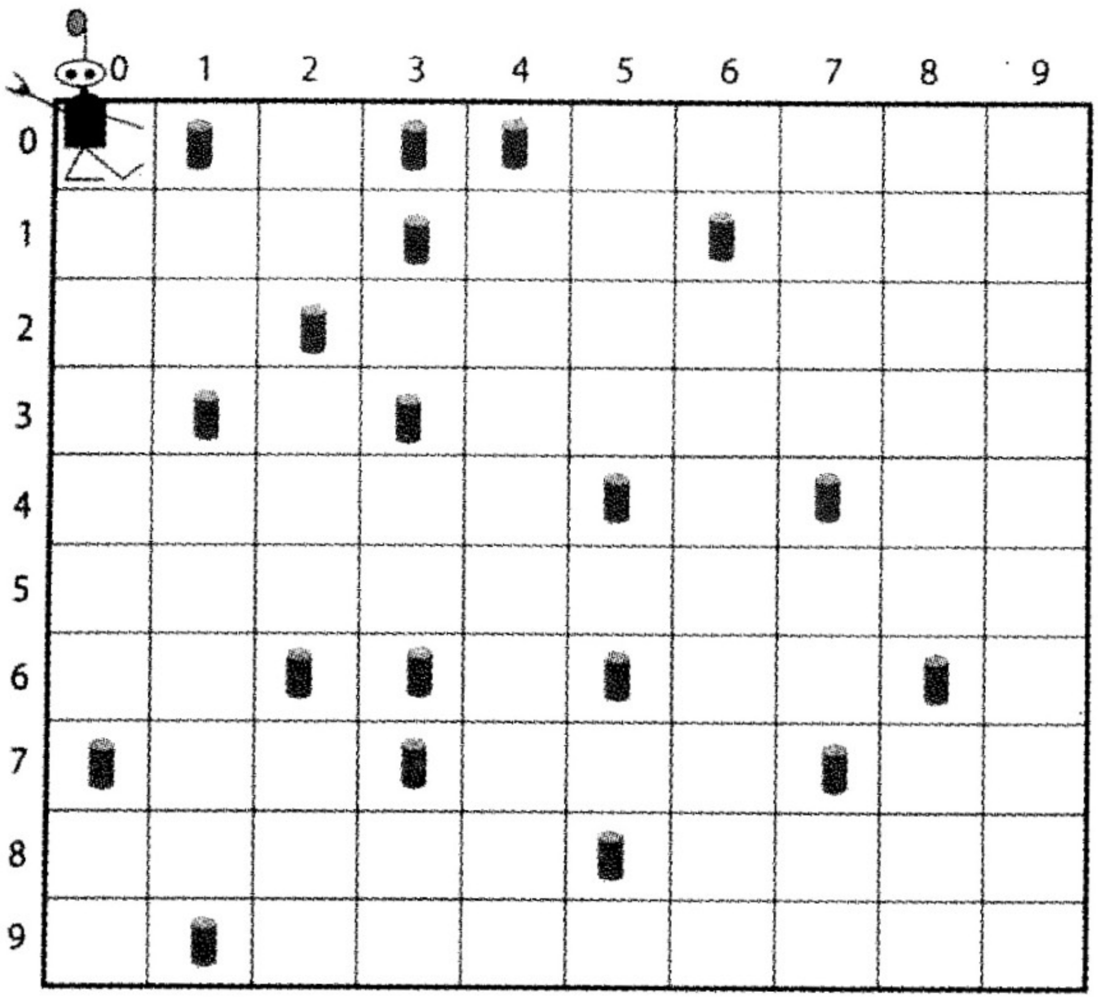
- 1. Define a format (a string) to represent different strategies. We call one strategy as one chromosome.
- 2. Give a population of some random chromosomes
- 3. Calculate each chromosome's fitness
- 4. Evolution: cross-over and mutate
- 6. Repeat from step 3 for enough generations

Basic GA Recipe

UPPSALA
UNIVERSITET

- 1. Define a format (a string) to represent different strategies. We call one strategy as one chromosome.
- 2. Give a population of some random chromosomes
- 3. Calculate each chromosome's fitness
- 4. Evolution: cross-over and mutate
- 6. Repeat from step 3 for enough generations

GA Evolving Robot



GA Evolving Robot: Strategy Format

UPPSALA UNIVERSITET

Situations					Action
North	East	South	West	Here	
-	-	-	-	-	Move north
-	-	-	-	can	Move east
-	-	-	-	wall	Pick up can
-	-	-	can	-	Move
……					
wall	-	can	wall	-	Stay still
……					
wall	wall	wall	wall	wall	Move east

GA Evolving Robot: Strategy Format

UPPSALA UNIVERSITET

Situations					Action
North	East	South	West	Here	
-	-	-	-	-	0
-	-	-	-	can	1
-	-	-	-	wall	6
-	-	-	can	-	4
……					
wall	-	can	wall	-	5
……					
wall	wall	wall	wall	wall	1

GA Evolving Robot: Strategy Format

UPPSALA UNIVERSITET

- 3^5 = 243 situations
- Move north
  Move east
  Move south
  Move west
  Move randomly
  Stay still
  Pick up can

GA Evolving Robot: Strategy Format

UPPSALA
UNIVERSITET

- Each chromosome is a string of 243 digits, each of which is between 0 and 6.
- There are 6^243 = 1.23e189 possible chromosomes.

- 3^5 = 243 situations
- Move north
  Move east
  Move south
  Move west
  Move randomly
  Stay still
  Pick up can

2330032342163034353054600610256251511416226043565433406651151 4
156502206406420510066432161615216520223644333633460133265030 00
406220502431650061113051466642324012456333455241261434413610 20
150630642551654043264463156164510543665346310551646005164

GA Evolving Robot: Measure Fitness

UPPSALA UNIVERSITET

- Given a finite time, the number of cans it picks up.
- The minimum time to pick all cans up.
- **Pick up can correctly +10;**
  Try to pick up but no can -1;
  Crash to the wall -5;
  Otherwise 0.

The fitness should be an average measured in many cases (e.g., 100 cases)

23300323421630343530546006102562515114162260435654334066511514
15650220640642051006643216161521652022364433363346013326503000
40622050243165006111305146664232401245633345524126143441361020
150630642551654043264463156164510543665346310551646005164

GA Evolving Robot: Cross-Over

UPPSALA
UNIVERSITET

1641134312102536034036124143120110423546252530420204451643366 5
6103532215310513144062212061463143215461025652364442202534034 5
3050200562063402633100245341643015163121001221440066401266524 6
35165015412311313245330443321263455500531421306442331 1000

23300323421630343530546006102562515114162260435654334066511514
15650220640642051006643216161521652022364433363346013326503000
40622050243165006111305146664232401245633345524126143441361020
15063064255165404326446315616451054366534631055164600 5164

GA Evolving Robot

UPPSALA UNIVERSITET

- 1. Define a format (a string) to represent different strategies. We call one strategy as one chromosome.
- 2. Give a population of some random chromosomes
- 3. Calculate each chromosome's fitness
- 4. Evolution: cross-over and mutate
- 6. Repeat from step 3 for enough generations

GA Evolving Robot

UPPSALA UNIVERSITET

- 1. Define a format (a string) to represent different strategies. We call one strategy as one chromosome.
- 2. Give a population of some random chromosomes (200)
- 3. Calculate each chromosome's fitness (100 random cases)
- **4. Evolution: cross-over and mutate**
- 6. Repeat from step 3 for 1000 generations

GA Evolving Robot

UPPSALA
UNIVERSITET

- **4. Evolution: cross-over and mutate**

- 4.1 Randomly select chromosome A and B based on their fitness
- 4.2 Randomly select a position and cross-over
- 4.3 By small probability (e.g., p = 0.05), mutate one gene
- 4.4 Repeat from 4.1 until you get 200 chromosomes

GA Evolving Robot

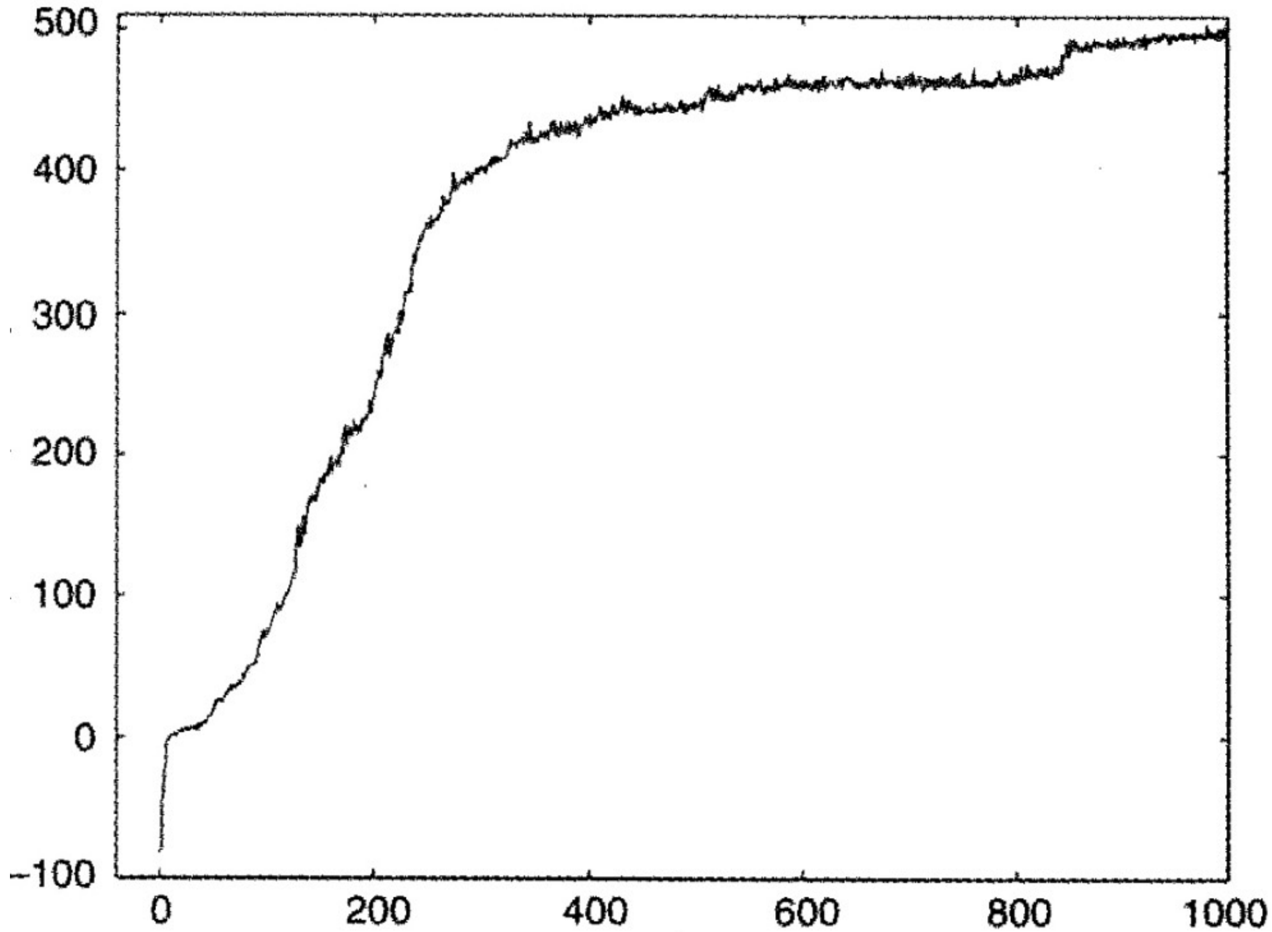
UPPSALA UNIVERSITET

- What parameters do we have in this case?

- 1. fixed population of chromosomes (200)
- 2. number of repeats to calculate average fitness (100)
- 3. mutation rate per chromosome (0.05)
- 4. number of generations (1000)

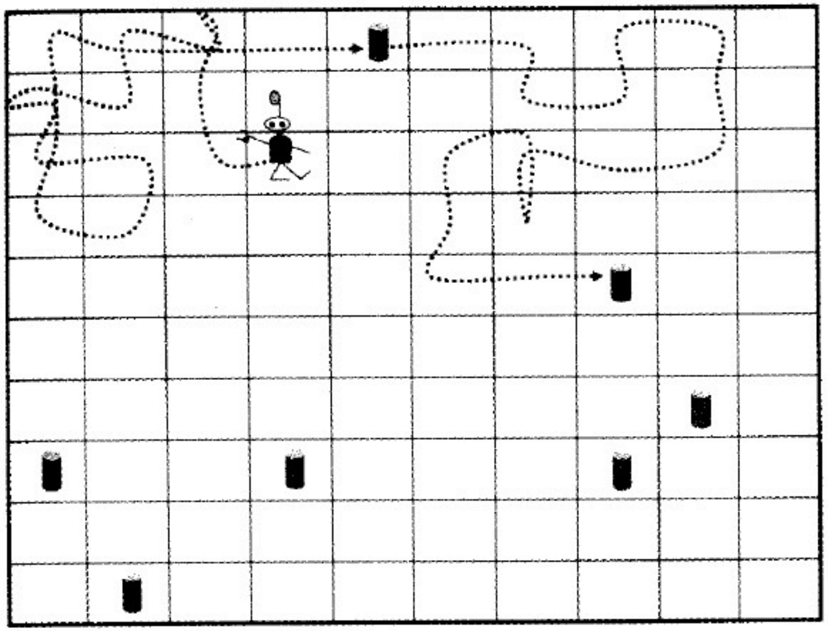
GA Evolving Robot

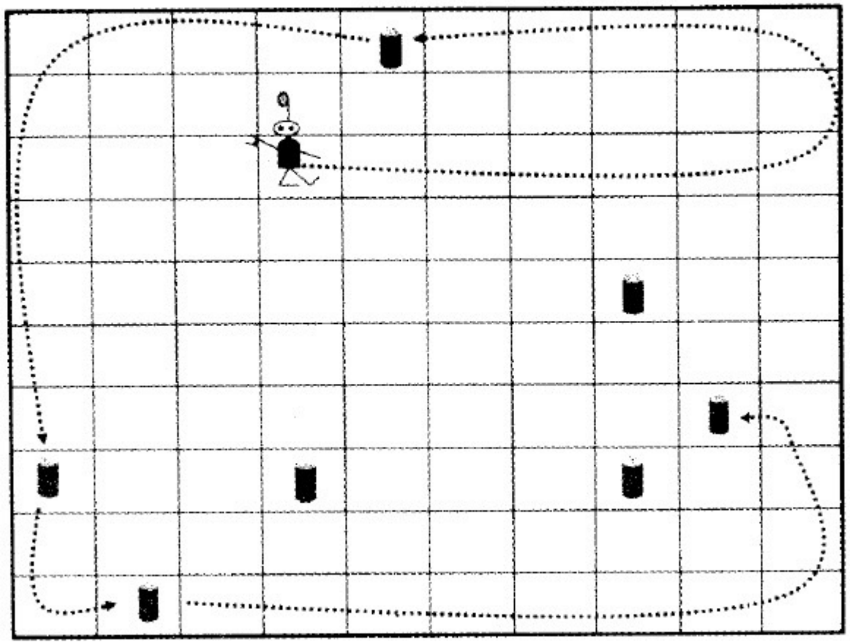
UPPSALA UNIVERSITET



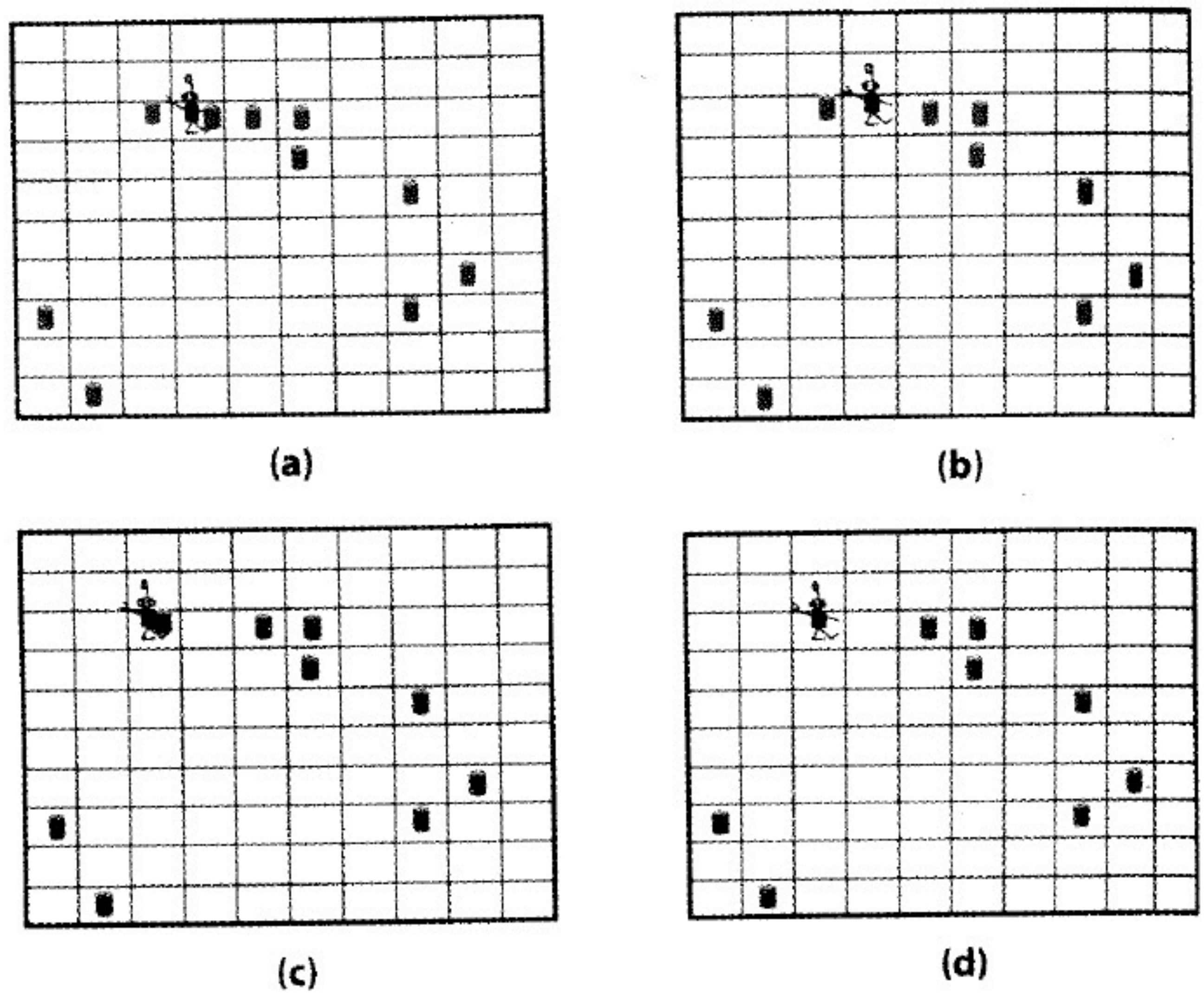
GA Evolving Robot

UPPSALA
UNIVERSITET



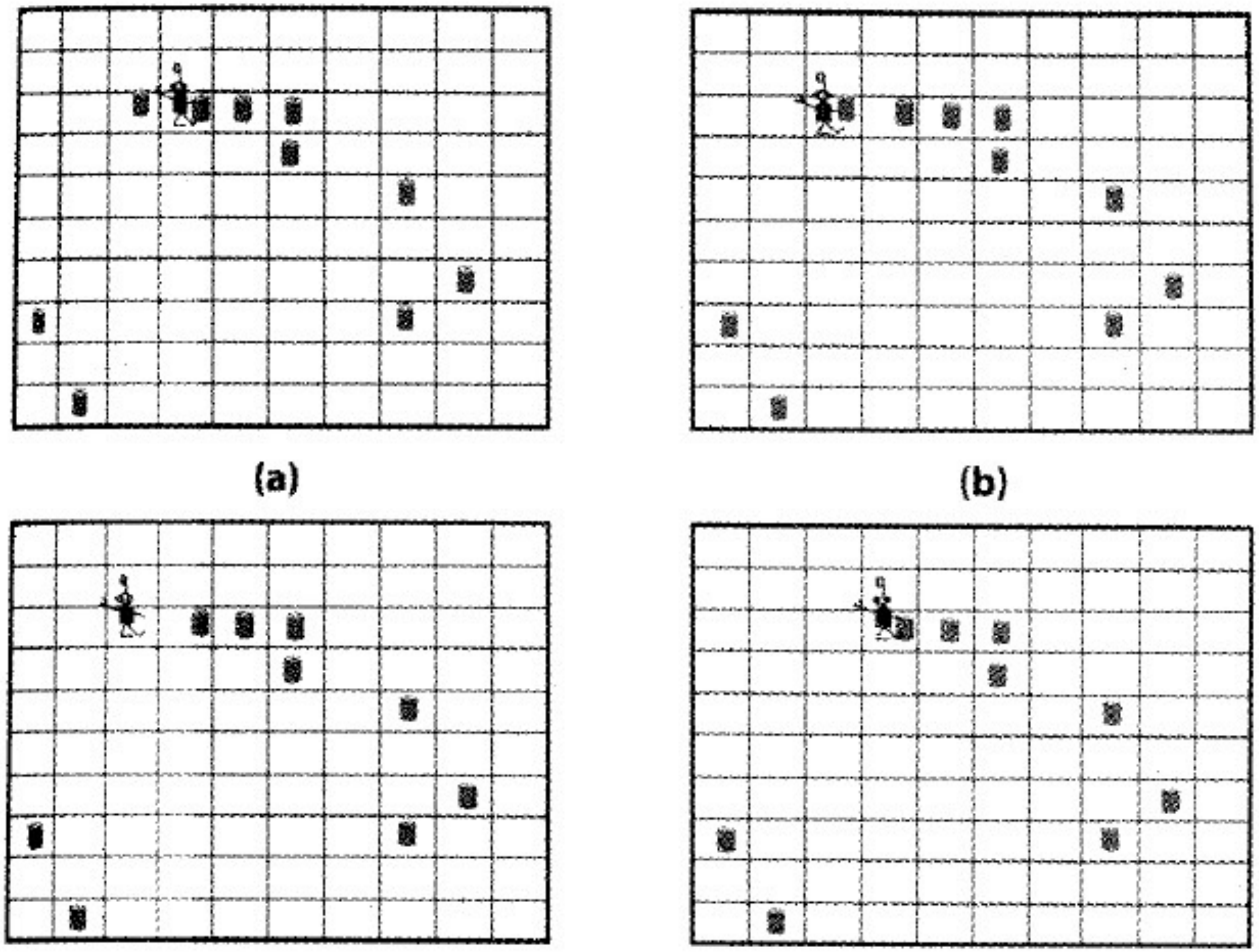


GA Evolving Robot

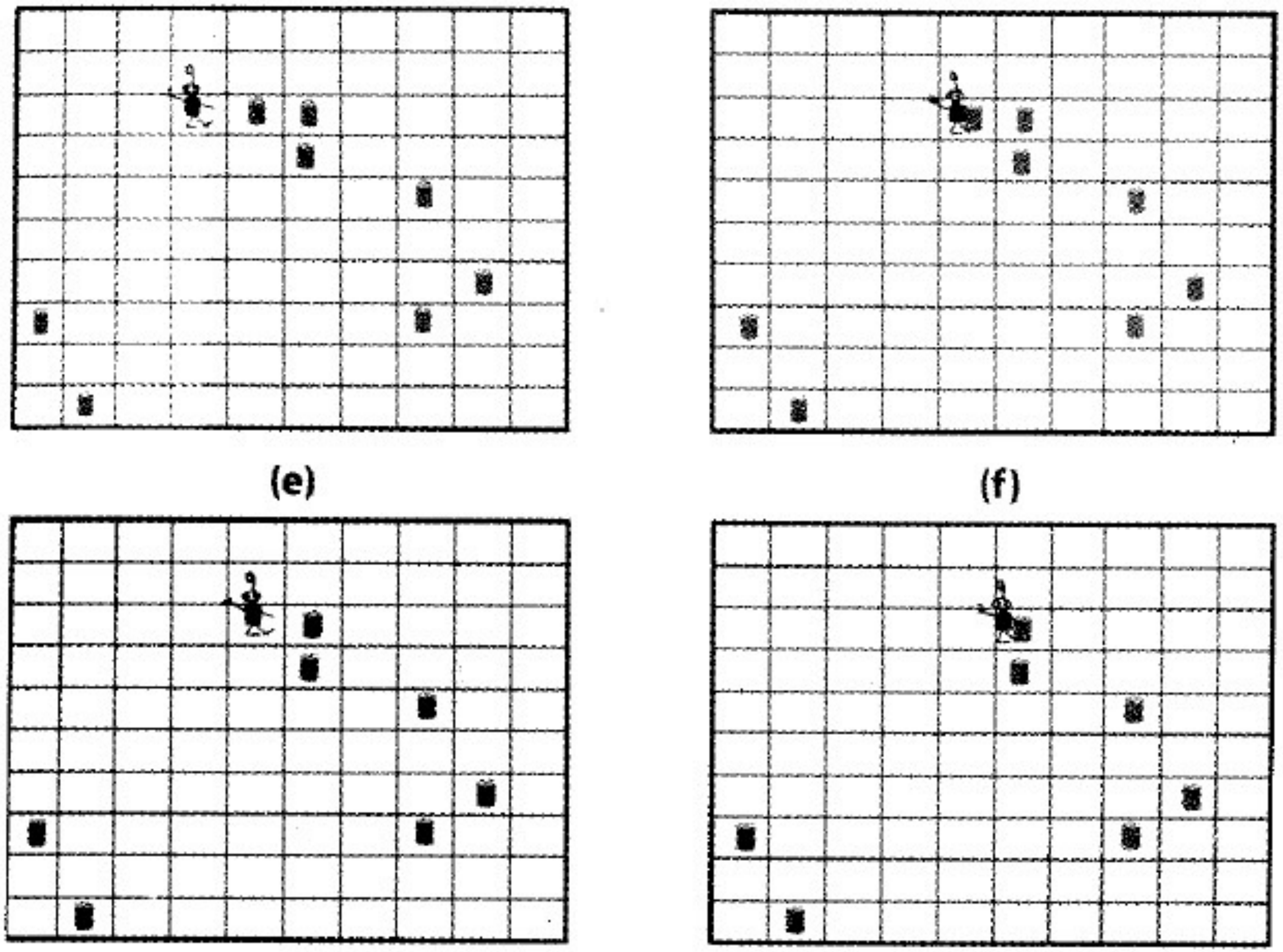


GA Evolving Robot

UPPSALA
UNIVERSITET



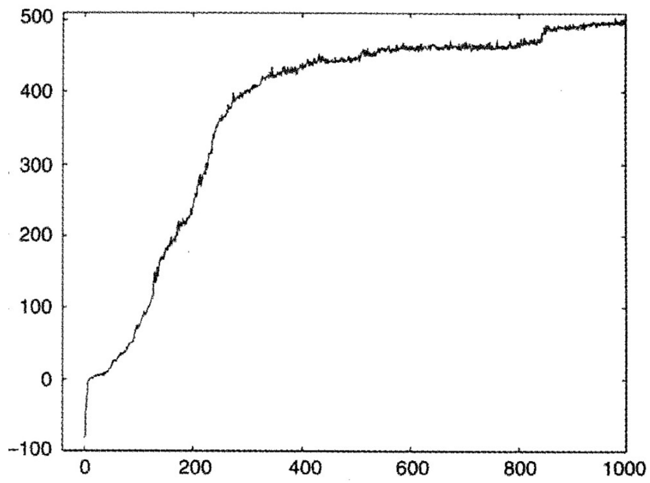
GA Evolving Robot



(e) (f)

GA Evolving Robot

UPPSALA UNIVERSITET



- Independent good genes are easy to appear.
- Cooperative genes are difficult to appear but also very important.

GA Evolving Robot

UPPSALA UNIVERSITET

- Why does GA work?
- A balance between selection, mutation and cross-over.
- 1. Low mutation rate make sure that 1) genes are not easy to be wiped out (both good and bad genes), and 2) there is chance of good innovations.
- 2. Good strategies can always be made of groups of good gene modulars. The cross-over can assemble modulars.
- 3. Selection picks the good genes and good gene modulars.

GA Cellular Automata Computer

UPPSALA UNIVERSITET

- Tell whether there are more black grids or not, based on local information.

GA Cellular Automata Computer

UPPSALA UNIVERSITET

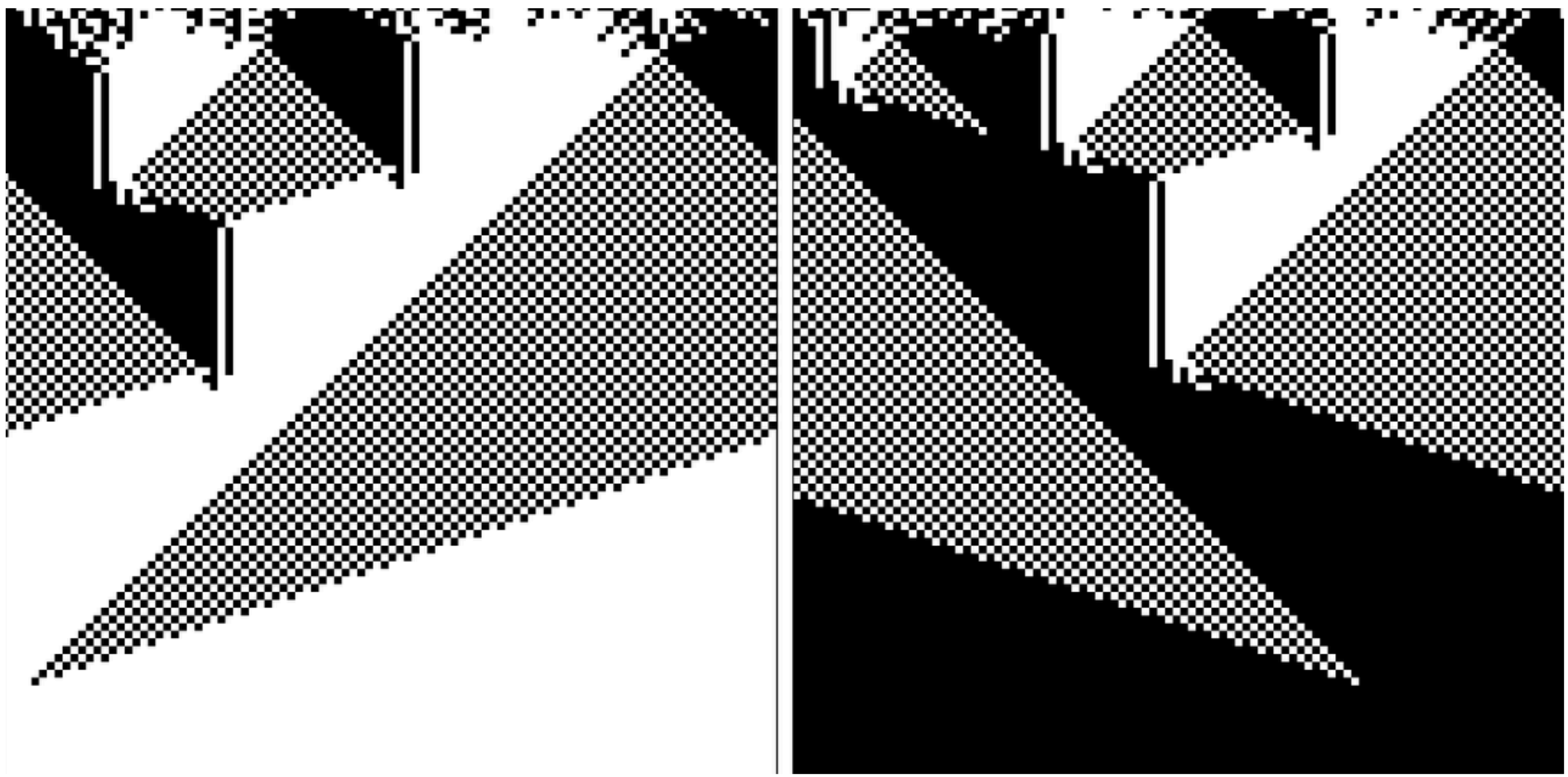
- Given a string of 0 and 1, tell whether there are more 1s or not, based on local information.

- 2^5 = 32 situations;
  Each situation has 2 possible actions, so there are
  2^32 = 4.295e9 strategies.

GA Cellular Automata Computer

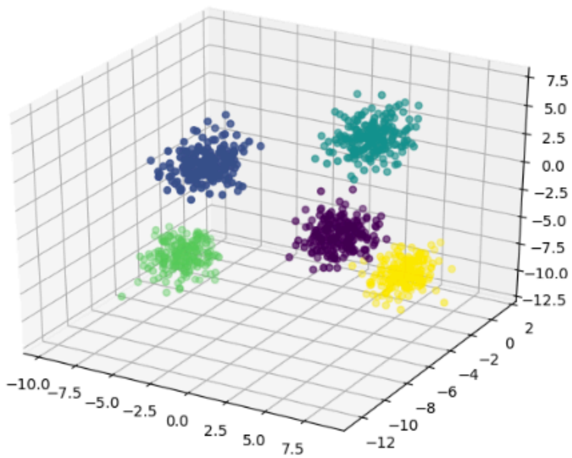
UPPSALA
UNIVERSITET

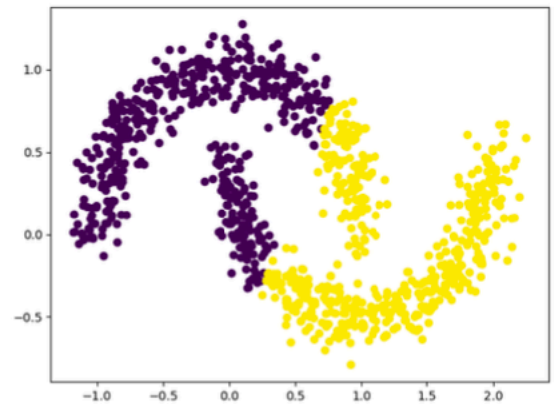
- Given a string of 0 and 1, tell whether there are more 1s or not, based on local information.



Example: GA K-Means

- K-means - way to cluster pts in n-dimensions into k clusters





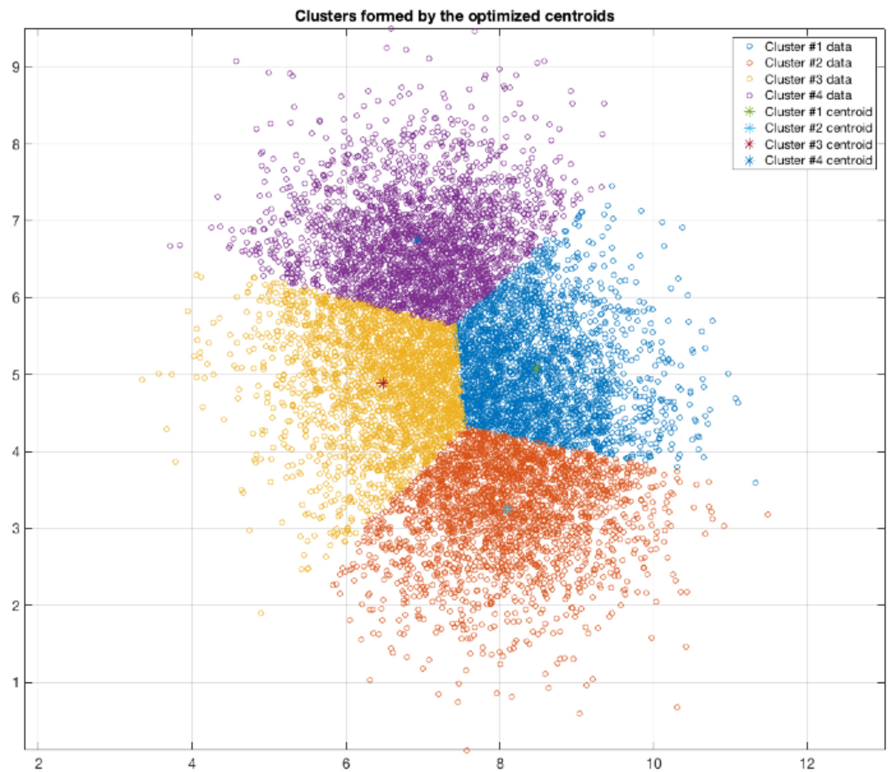
$$\mathscr{M}(C_1,\ldots,C_K)=\sum_{i=1}^{K}\sum_{\boldsymbol{x}_j\in C_i}||\boldsymbol{x}_j-\boldsymbol{c}_i||$$

Charitini Stavropoulou

Katarzyna Kowalczyk

Example: G A K-Means

- K-means - way to cluster pts in n-dimensions into k clusters



$$\mathscr{M}(C_1, \ldots, C_K) = \sum_{i=1}^{K} \sum_{\boldsymbol{x}_j \in C_i} ||\boldsymbol{x}_j - \boldsymbol{c}_i||$$

Charitini Stavropoulou

Katarzyna Kowalczyk

Dataset To Cluster -

UPPSALA UNIVERSITET

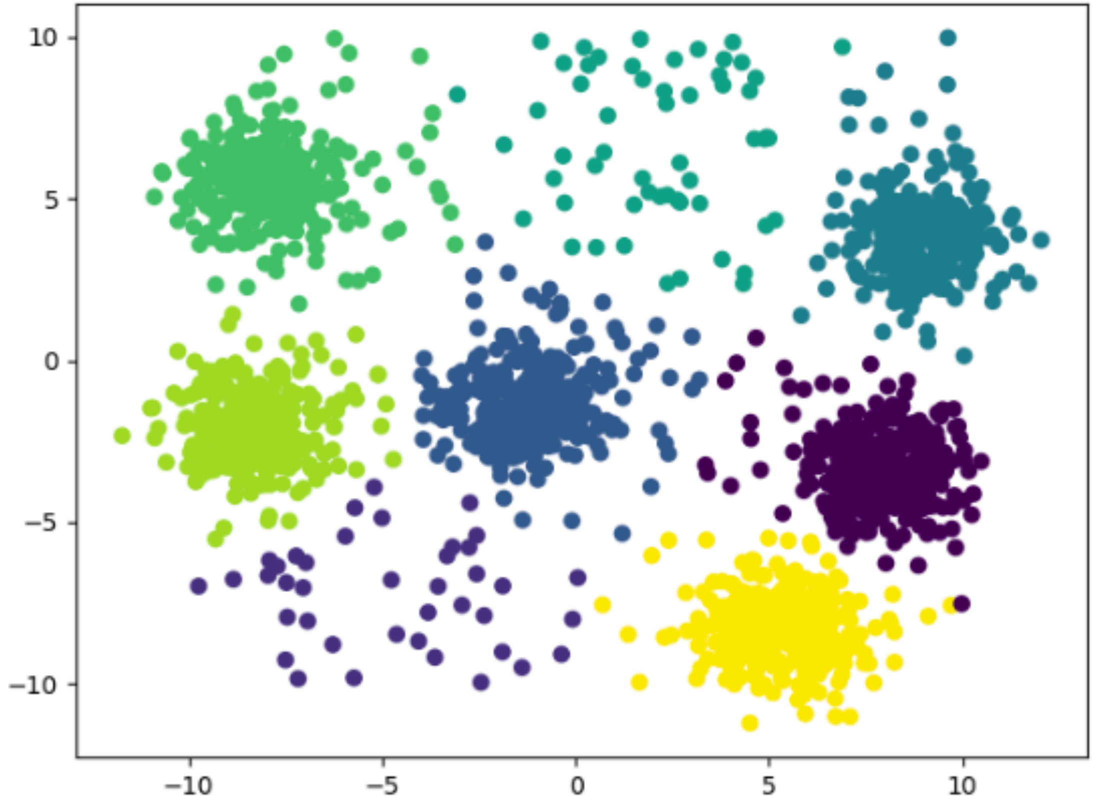


Figure 2: Data set used in the experiments.

Charitini Stavropoulou

Katarzyna Kowalczyk

Selecting Which Chromosomes Breed…

UPPSALA UNIVERSITET

- Tournament - pick groups of s individuals and return individual with highest fitness.

e.g. if $s = 2$ and chromosomes i, j chosen then return $\arg\max\{f_i, f_j\}$

- Roulette wheel - each chromosome i chosen with probability proportional to fitness f_i.

$$\text{Probability of choosing } i = \frac{f_i}{\sum_j f_j}$$

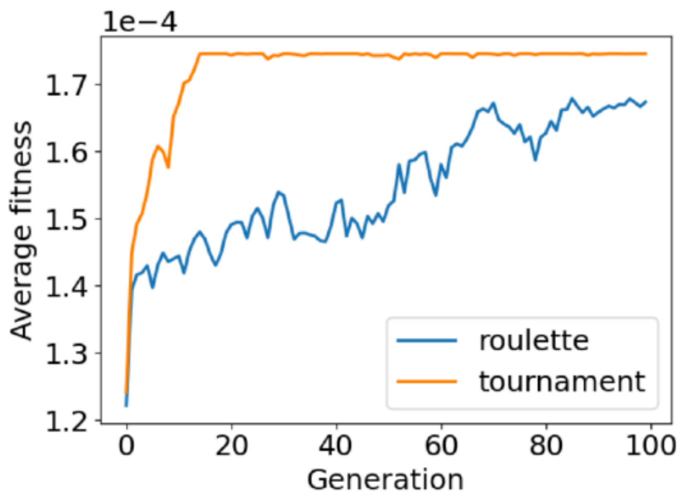
- Fitness - calculated to be $\sim 1/\mathscr{M}$

Charitini Stavropoulou

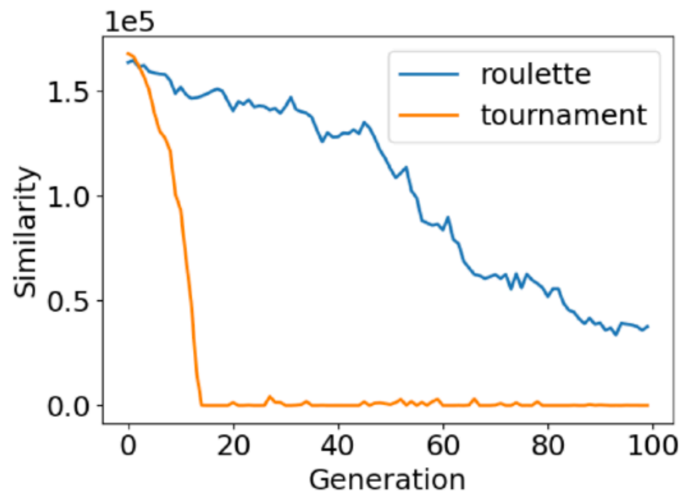
Katarzyna Kowalczyk

Selection Matters

UPPSALA
UNIVERSITET



(a) Average fitness



(b) Generation similarity

Figure 6: Comparison of two selection strategies for the same data set and GA parameters: 100 generations, population of 100, $\mu_c = 0.8$, $\mu_m = 0.01$.

- Generational similarity. Treat each chromosome as a point in $\mathbb{R}^{kn}$ and define to be sum of pairwise distances of chromosomes in generation.

Charitini Stavropoulou

Katarzyna Kowalczyk

Example: Tic Tac Toe

UPPSALA UNIVERSITET

- Aim - find a non-losing strategy!
- Map to chromosomes -

Table 1: First eight rows of the game-base.

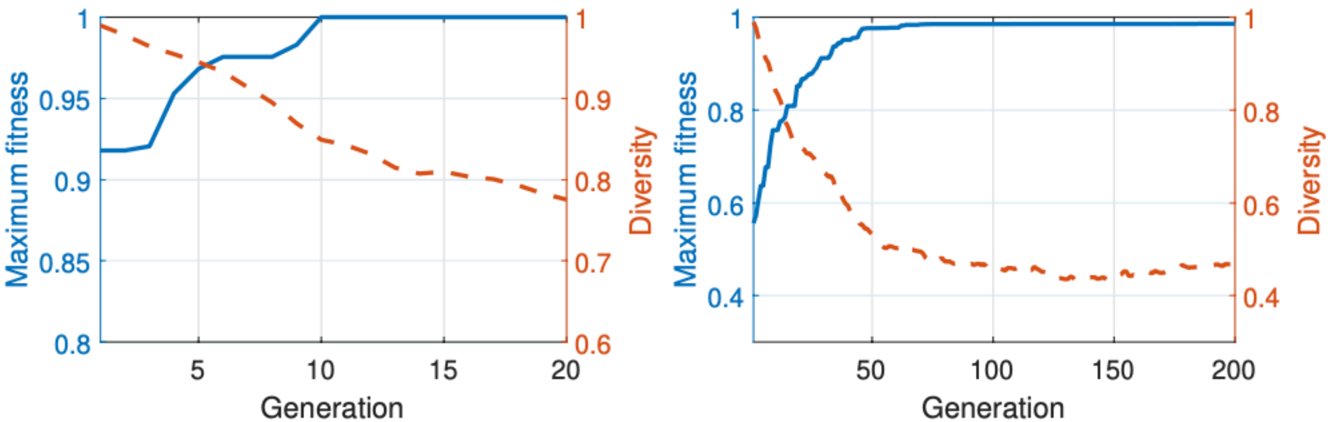
State representation									Game level	Winner status	Valid next states						
0	0	0	0	0	0	0	0	0	0	0	2	654	763	0	0	0	0
1	0	0	0	0	0	0	0	0	1	0	3	429	602	627	650	0	0
1	2	0	0	0	0	0	0	0	2	0	4	122	266	334	387	410	422
1	2	1	0	0	0	0	0	0	3	0	5	72	93	118	0	0	0
1	2	1	2	0	0	0	0	0	4	0	6	29	51	60	68	0	0
1	2	1	2	1	0	0	0	0	5	0	7	12	21	25	0	0	0
1	2	1	2	1	2	0	0	0	6	0	8	9	0	0	0	0	0
1	2	1	2	1	2	1	0	0	7	1	0	0	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

763	3	4	72	68	21	9	0	⋯

Emil Åberg

Example: Tic Tac Toe

UPPSALA
UNIVERSITET



(a) Only first player.

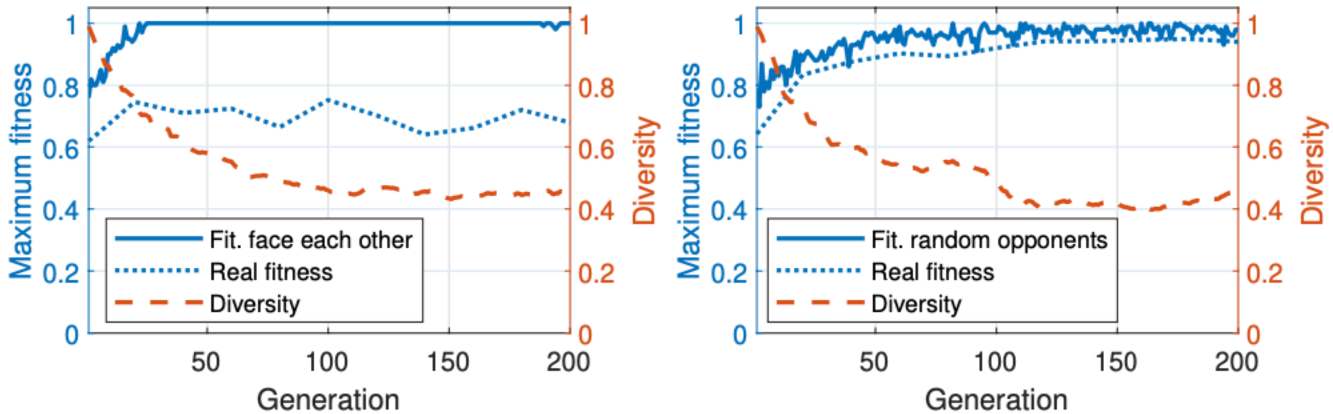
(b) Only second player.

Figure 7: Shows the maximum fitness of a population of size 100 versus generation.

Emil Åberg

Example: Tic Tac Toe

UPPSALA
UNIVERSITET



(a) Simplified fitness calculated by letting strategies face each other.

(b) Simplified fitness calculated by letting strategies face randomly generated strategies.

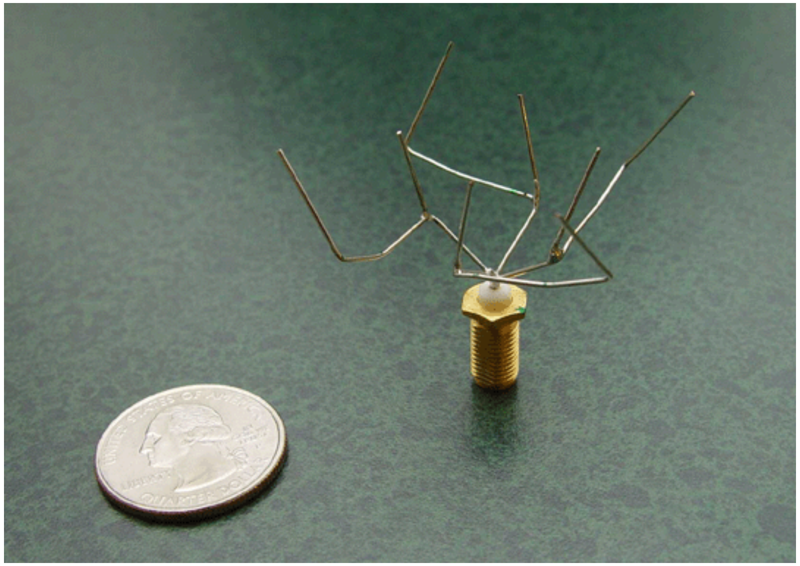
Figure 9: Shows how the maximum fitness calculated in a simple way varies over generations. The real fitness and the diversity is also shown.

Emil Åberg

Comments On GA

UPPSALA UNIVERSITET

- Automated design (e.g., Shape of the plane, antenna)
- Analyse satellite images
- Animations in film (e.g., horses in The Lord of the Rings III)
- Develop new drugs
- Protein folding
- ……



https://en.wikipedia.org/wiki/List_of_genetic_algorithm_applications

Comments On GA

UPPSALA UNIVERSITET

- GA always cannot get the best solution (there may be not a best solution), but can be good enough.
- Biological evolution is open-ended, while we define an end for GA.
- For biological evolution, the whole solution space is not fixed; while for GA generally, the whole solution space is actually fixed.

G A Vs Machine Learning

UPPSALA UNIVERSITET

- The common part is the ability to learn or 'fit' to data for predictions.
- Both have a fitness function - to determine how well the algorithm is performing
- GA is an example of reinforcement learning
- GA group of algorithms, rather than a single algorithm.
- Update rules from group of algorithms to group of algorithms in GA, very different to how one updates algorithms in other machine learning contexts.
- Nice example of reinforcement learning, (but not a GA!) is: arxiv:1707.02286

-see videos here - https://www.youtube.com/watch?v=hx_bgoTF7bs