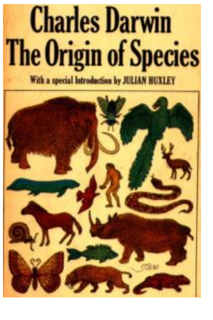
Genetic Algorithms (GAs)

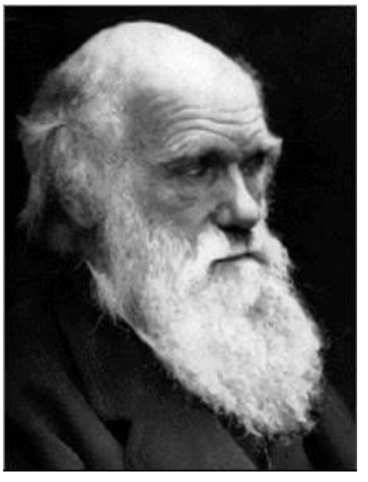
N. Vassilas
Department of Computer Engineering
University of West Attica

What is a Genetic Algorithm?

- Genetic algorithms (GAs) are techniques to solve problems which need optimization
- GAs are a subclass of Evolutionary Computation
- GAs are based on Darwin's theory of evolution



Charles Darwin 1809 - 1882



"A man who dares to waste an hour of life has not discovered the value of life"

Genetic Algorithms

- Directed search algorithms based on the mechanics of biological evolution
- Provide efficient, effective techniques for optimization and machine learning applications
- “Genetic Algorithms are good at taking large, potentially huge search spaces and navigating them, looking for optimal combinations of things, solutions you might not otherwise find in a lifetime.”

  - *Salvatore Mangano, Computer Design, May 1995*

Genetic Algorithms

- Developed by John Holland, University of Michigan (1970's)
  - To understand the adaptive processes of natural systems
  - To design artificial systems software that retains the robustness of natural systems
- Widely-used today in business, scientific and engineering circles

Components of a GA

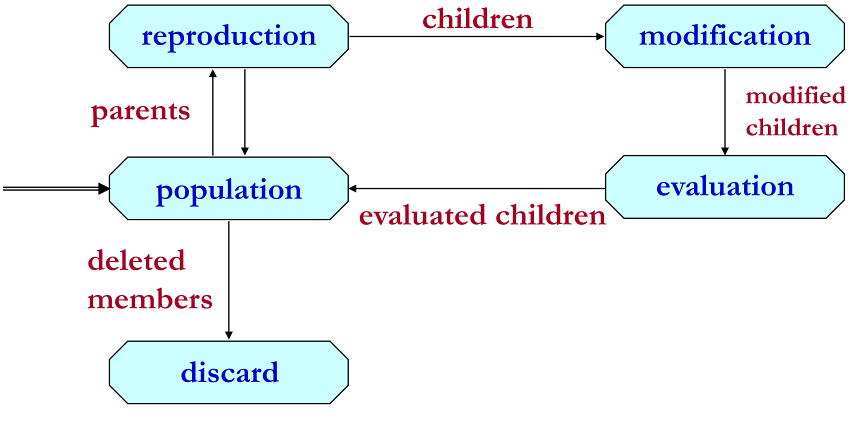
A problem to solve, and ...

- Encoding technique (gene, chromosome)
- Initialization procedure (creation)
- Evaluation function (environment)
- Selection of parents (reproduction)
- Genetic operators (mutation, crossover)
- Parameter settings (practice and art)

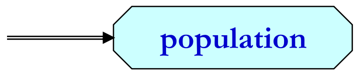
Algorithm GA

```
Given a method of encoding solutions of a problem into
chromosomes and a chromosome evaluation (fitness) function,
a GA consists of the following steps:

1: Initialize a population of chromosomes

2: Evaluate each chromosome in the population

3: Create new chromosomes by mating current chromosomes.
   Apply crossover and mutation operators.

4: Delete less fit members of the population.

5: Evaluate and insert the new chromosomes in the population

6: If stopping criterion is satisfied, then stop and return
   the best chromosome; otherwise, go to step 3.

END GA
```

The GA Cycle of Reproduction



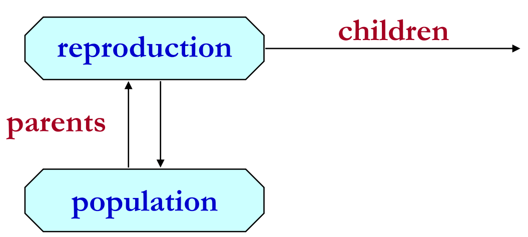
Population



Chromosomes could be:

- Bit strings (0101 ... 1100)
- Real numbers (43.2 -33.1 ... 0.0 89.2)
- Permutations of element (E11 E3 E7 ... E1 E15)
- Program elements (genetic programming)
- ... any data structure ...

Reproduction



Parents are selected at random with selection chances biased in relation to chromosome evaluations.

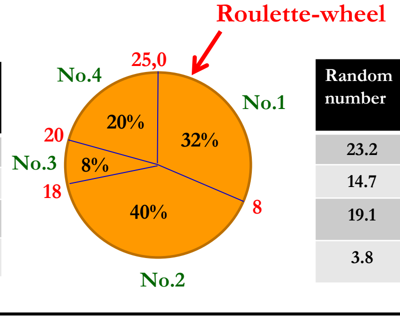
Reproduction

- Reproduction is the process in which chromosomes are copied according to their fitness value.
- This operator corresponds to natural selection (survival of the fittest).
- A fitness value is assigned to each individual in the population with high scores denoting good fit.
- Typical methods for parent selection: roulette-wheel parent selection, rank parent selection, elitism and tournement parent selection techniques.

Roulette-wheel parent selection

- Compute total fitness of all population members
- Generate n, a random number between 0 and total fitness
- Return the first member whose fitness, added to the fitness of the preceding population members (the running total), is greater than or equal to n.
- Example:

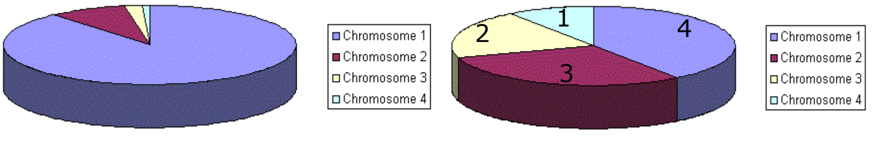
	Chromo -some	Fitness	% of total	Running total
1	01110	8	32	8
2	11000	10	40	18
3	00100	2	8	20
4	10010	5	20	25



Random number	Chosen member
23.2	4
14.7	2
19.1	3
3.8	1

Rank parent selection

- Roulette-wheel selection has problems when the fitnesses differ very much.
- Rank selection first ranks the population and then every chromosome receives fitness from this ranking. The worst will have fitness 1, second worst 2 … and the best will have fitness N (number of chromosomes in population).
- Use roulette-wheel on new running total of fitness values



Before ranking *After ranking*

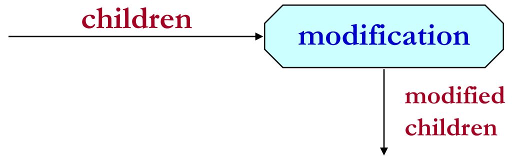
Elitism (cloning)

- Idea of elitism comes from the following: When creating new population by crossover and mutation, we have a big chance, that we will loose the best chromosome.
- Elitism is name of method, which first copies (clones) the best chromosome (or a few best chromosomes) to new population. The rest is done in classical way.
- Elitism can very rapidly increase performance of GA, because it prevents losing the best found solution.

Tournement parent selection

- Tournament selection involves running several "tournaments" among a few individuals chosen at random from the population.
- Variant 1: Select P (usually P = 2) individuals at random. The individual with the highest evaluation becomes the parent. Repeat to find a second parent.
- Variant 2: Select P individuals at random. Use roulette-wheel on the P individuals to select first parent. Repeat to select other parents.
- Variant 3: Select P individuals at random. Use rank selection method on the P individuals to select first parent. Repeat to select other parents.

Chromosome Modification



- Modifications are stochastically triggered
- Operator types:
  - Crossover (recombination)
  - Mutation

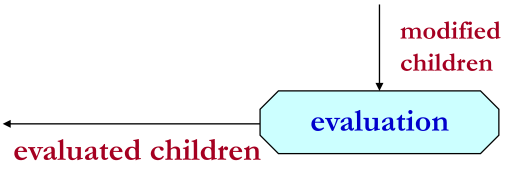
Crossover

```
                ↓
P1  (0 1 [1] 0 1 0 0 0)      (0 1 [0] 0 1 0 0 0)  C1
P2  (1 1 [0] 1 1 0 1 0)      (1 1 [1] 1 1 0 1 0)  C2
```

Crossover is a critical feature of genetic algorithms:

- It greatly accelerates search early in evolution of a population
- It leads to effective combination of schemata (subsolutions on different chromosomes)

Evaluation

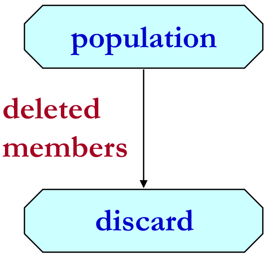


- The evaluator decodes a chromosome and assigns it a fitness value
- The evaluator is the only link between a classical GA and the problem it is solving

Fitness

- A measure of the goodness of the organism
- Expressed as the probability that the organism will live another cycle (generation)
- Basis for the natural selection simulation
  - Organisms are selected to mate with probabilities proportional to their fitness
- Probabilistically better solutions have a better chance of conferring their building blocks to the next generation

Deletion



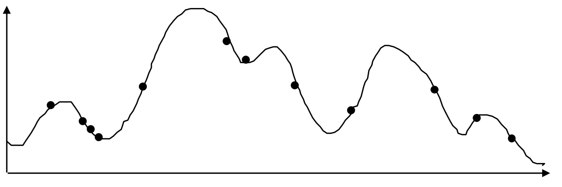
- **Generational GA:**

  entire populations replaced with each iteration

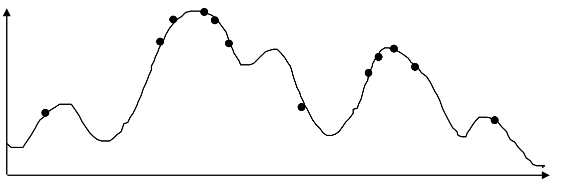
- **Steady-state GA:**

  a few members replaced each generation

An Abstract Example

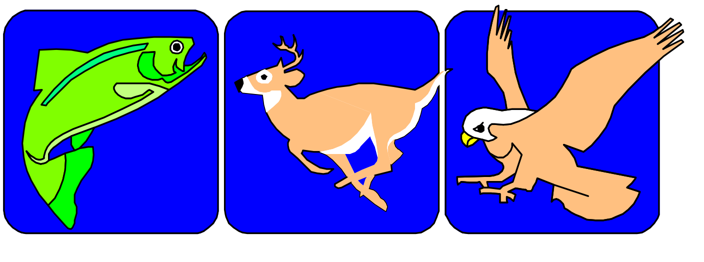


Distribution of Individuals in Generation 0



Distribution of Individuals in Generation N

A Simple Example



"The Gene is by far the most sophisticated program around."

- Bill Gates, Business Week, June 27, 1994

Traveling Salesman Problem (TSP)

Find a tour of a given set of cities so that:

- each city is visited only once
- the total distance traveled is minimized

Representation

Representation is an ordered list of city numbers known as an order-based GA.

1) London	3) Dunedin	5) Beijing	7) Tokyo
2) Venice	4) Singapore	6) Phoenix	8) Victoria

CityList1 (3 8 7 2 1 6 4 5)
CityList2 (2 5 7 6 8 1 3 4)

Crossover

Parent1	(3	8	7	2	1	6	4	5)
Parent2	(2	5	7	6	8	1	3	4)
Child1	(5	8	7	2	1	6	3	4)
Child2	(3	5	7	6	8	2	1	4)

This operator is called order crossover: we select a subset from the first parent and then add that subset to the offspring. Any missing values are then added from the 2nd parent starting from the beginning in the order that they are found.

Crossover (2nd variant)

Parent1	(3	8	7	2	1	6	4	5)
Parent2	(2	5	7	6	8	1	3	4)
Child1	(6	8	7	2	1	3	4	5)
Child2	(1	5	7	6	8	4	3	2)

In this version of order crossover, instead of ordering the remaining values of the second parent starting from the beginning of the chromosome, the remaining values are cyclically ordered starting from the second cut point.

Mutation

Mutation involves reordering of the list:

Before:	(5	8	$\boxed{7}$	2	1	$\boxed{6}$	3	4)
After:	(5	8	6	2	1	7	3	4)

TSP Example: 5 Cities

(5,3,4,1,2)	(2,4,1,3,5)	(4,3,1,5,2)
(2,3,4,1,5)	(4,3,1,2,5)	(3,4,5,2,1)
(3,5,4,1,2)	(4,5,3,1,2)	(5,4,2,3,1)
(4,1,3,2,5)	(3,4,2,1,5)	(3,2,5,1,4)

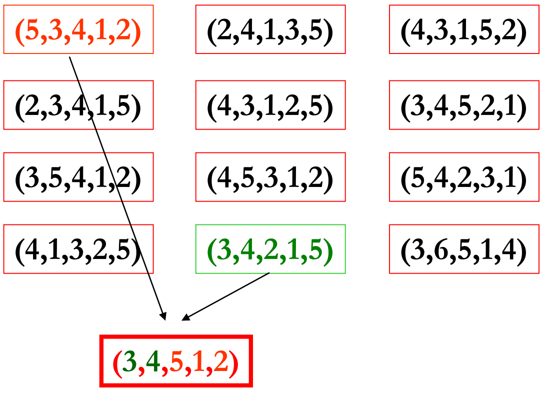
Initial population

Select Parents

(5,3,4,1,2)	(2,4,1,3,5)	(4,3,1,5,2)
(2,3,4,1,5)	(4,3,1,2,5)	(3,4,5,2,1)
(3,5,4,1,2)	(4,5,3,1,2)	(5,4,2,3,1)
(4,1,3,2,5)	(3,4,2,1,5)	(3,2,5,1,4)

Pick the two parents using a parent selection method.

Create Off-Spring



Create More Offspring

(5,3,4,1,2)	(2,4,1,3,5)	(4,3,1,5,2)
(2,3,4,1,5)	(4,3,1,2,5)	(3,4,5,2,1)
(3,5,4,1,2)	(4,5,3,1,2)	(5,4,2,3,1)
(4,1,3,2,5)	(3,4,2,1,5)	(3,1,5,1,4)

(3,4,5,1,2) (5,4,2,1,3)

Mutate

(5,3,4,1,2)	(2,4,1,3,5)	(4,3,1,5,2)
(2,3,4,1,5)	(4,3,1,2,5)	(3,4,5,2,1)
(3,5,4,1,2)	(4,5,3,1,2)	(5,4,2,3,1)
(4,1,3,2,5)	(3,4,2,1,5)	(3,1,5,1,4)

(3,4,5,1,2) (5,4,2,1,3)

Mutate

(5,3,4,1,2)	(2,4,1,3,5)	(4,3,1,5,2)
(2,3,4,1,5)	(4,3,1,2,5)	(3,4,5,2,1)
(3,5,4,1,2)	(4,5,3,1,2)	(5,4,2,3,1)
(4,1,3,2,5)	(3,4,2,1,5)	(3,1,5,1,4)

(3,4,5,1,2) (5,4,2,1,3)

Eliminate

(5,3,4,1,2)	(2,4,1,3,5)	~~(4,3,1,5,2)~~
~~(2,3,4,1,5)~~	(4,3,1,2,5)	(3,4,5,2,1)
(3,5,4,1,2)	(4,5,3,1,2)	(5,4,2,3,1)
(4,1,3,2,5)	(3,4,2,1,5)	(3,1,5,1,4)

(3,4,5,1,2) (5,3,2,1,4)

Eliminate the worst chromosomes.

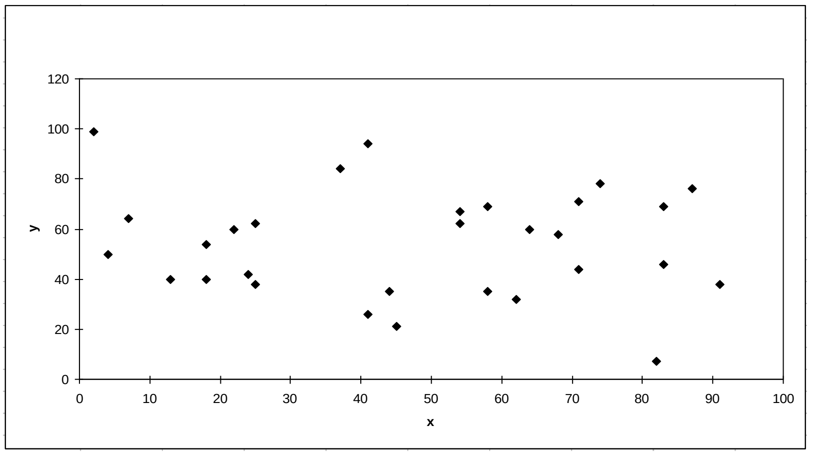
Integrate

(5,3,4,1,2)	(2,4,1,3,5)	(5,3,2,1,4)
(3,4,5,1,2)	(4,3,1,2,5)	(3,4,5,2,1)
(3,5,4,1,2)	(4,5,3,1,2)	(5,4,2,3,1)
(4,1,3,2,5)	(3,4,2,1,5)	(3,1,5,1,4)

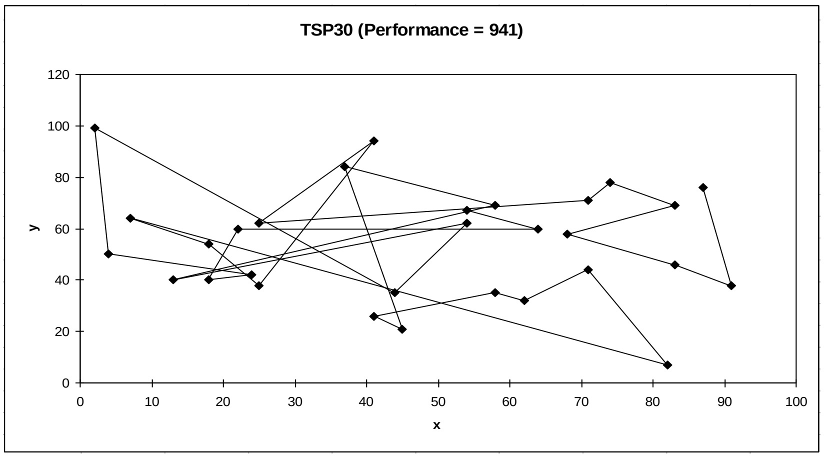
Restart

(5,3,4,1,2)	(2,4,1,3,5)	(5,3,2,1,4)
(3,4,5,1,2)	(4,3,1,2,5)	(3,4,5,2,1)
(3,5,4,1,2)	(4,5,3,1,2)	(5,4,2,3,1)
(4,1,3,2,5)	(3,4,2,1,5)	(3,1,5,1,4)

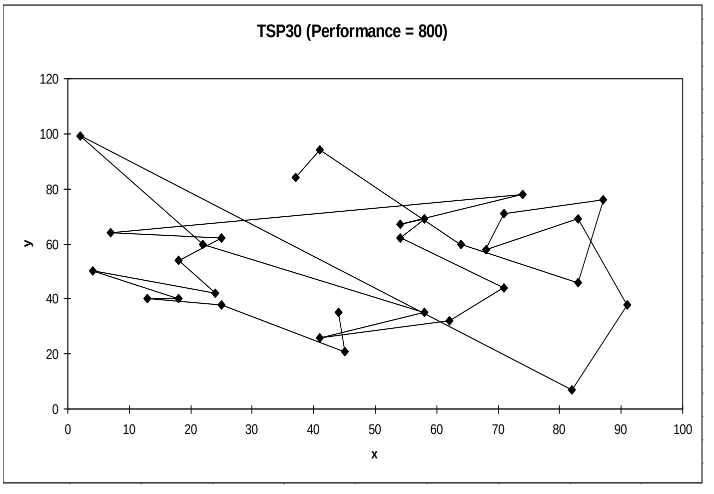
TSP Example: 30 Cities



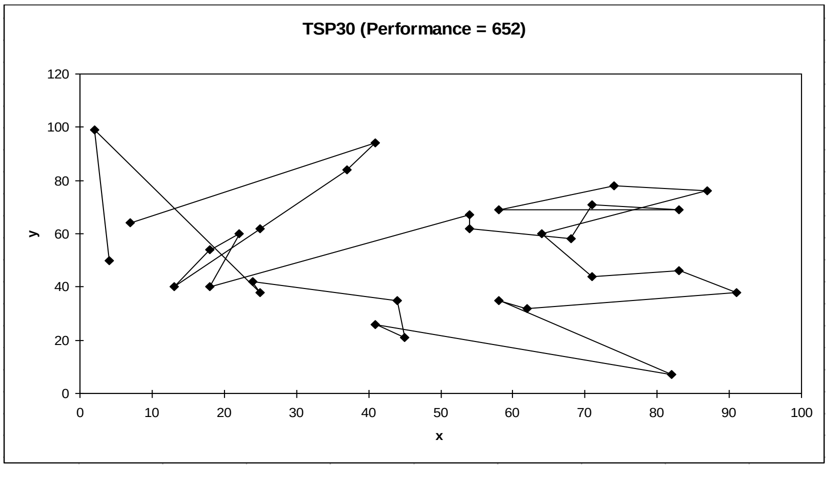
Solution i (Distance = 941)



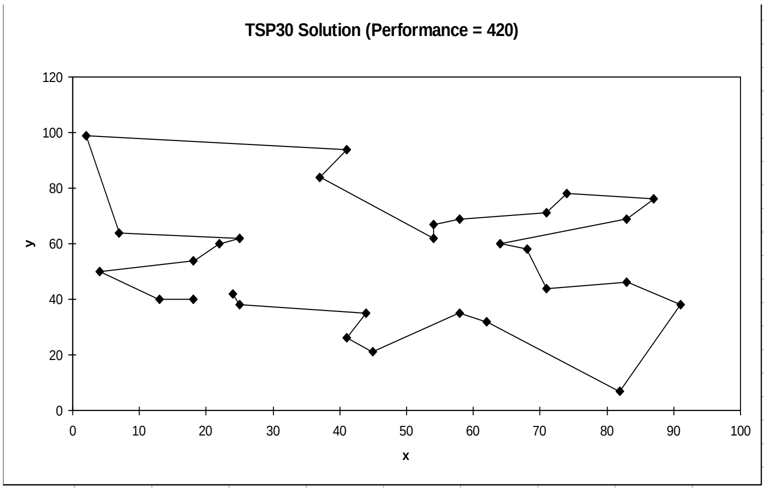
Solution j (Distance = 800)



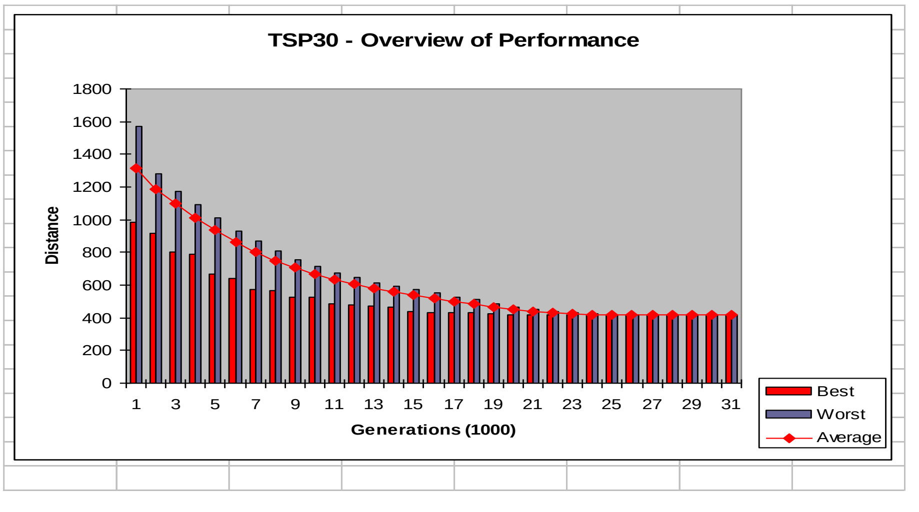
Solution k (Distance = 652)



Best Solution (Distance = 420)



Overview of Performance

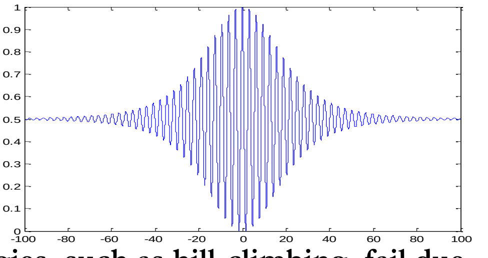


A Difficult Problem

- Find the maximum of the following function

$$f(x, y) = 0.5 - \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{[1.0 + 0.001 (x^2 + y^2)]^2}$$

where $x, y \in [-100, 100]$.



Graph of the function for fixed y at its optimal value

- Most optimization strategies, such as hill-climbing, fail due to the very high oscillation of the function.
- **SOLUTION:** Use a genetic algorithm

GA Solution

Chromosome Encoding

- Let us use L = 44 bits for encoding both x and y as a binary string.

Chromosome Decoding

- Split the 44-bit chromosome into two 22-bit strings, one for each of x and y, interpreted as integers in base-2 notation.
- Multiply x and y by $200/(2^{22} - 1)$ to map the values from the range $[0, 2^{22} - 1]$ to the range [0, 200].
- Subtract 100 from x and y to map them in [-100, 100].

Fitness function

- Use f(x, y) (positive function) as the fitness function.

GA Solution

One chromosome:

$$\underbrace{0000101000011000000001}_{x\,=\,165377}\underbrace{1000101010001110111011}_{y\,=\,2270139}$$

$x = 165377 \downarrow x = -92.11$

$y = 2270139 \downarrow y = 8.25$

Fitness score: $f(-92.11, 8.25) = 0.495$

GA Solution

- Initial population: 100 random 44-bit strings.
- Three operators used: roulette-wheel parent selection, simple crossover with random mating and simple mutation.
- Parameters: $n = 100$ (population size), $p_c = 0.65$ (crossover probability) and $p_m = 0.008$ (mutation probability).
- After 40 generations, the 4 best chromosomes are:

chromosome	fitness score
01111001011000101101011000100001100100010011	0.9930
01111011111000101101111000101000110110010010	0.9926
01111011110000100101011000101000110110010011	0.9925
01111011110000000101011000101000110110010001	0.9925

Issues for GA Practitioners

- Choosing basic implementation issues:
  - representation
  - population size, mutation rate, ...
  - selection, deletion policies
  - crossover, mutation operators
- Termination Criteria
- Performance, scalability
- Solution is only as good as the evaluation function (often hardest part)

Benefits of Genetic Algorithms

- Concept is easy to understand
- Modular, separate from application
- Supports multi-objective optimization
- Good for “noisy” environments
- Always an answer; answer gets better with time
- Inherently parallel; easily distributed

Benefits of Genetic Algorithms (cont.)

- Many ways to speed up and improve a GA-based application as knowledge about problem domain is gained
- Easy to exploit previous or alternate solutions
- Flexible building blocks for hybrid applications
- Substantial history and range of use

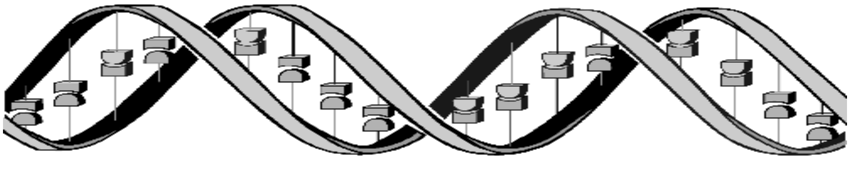
When to Use a GA

- Alternate solutions are too slow or overly complicated
- Need an exploratory tool to examine new approaches
- Problem is similar to one that has already been successfully solved by using a GA
- Benefits of the GA technology meet key problem requirements

Some GA Application Types

Domain	Application Types
Control	gas pipeline, pole balancing, missile evasion, pursuit
Design	semiconductor layout, aircraft design, keyboard configuration, communication networks
Scheduling	manufacturing, facility scheduling, resource allocation
Robotics	trajectory planning
Machine Learning	designing neural networks, improving classification algorithms, classifier systems
Signal Processing	filter design
Game Playing	poker, checkers, prisoner's dilemma
Combinatorial Optimization	set covering, travelling salesman, routing, bin packing, graph colouring and partitioning

Conclusions



Question: **'If GAs are so smart, why ain't they rich?'**

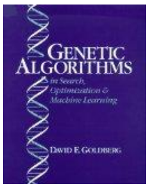
Answer: **'Genetic algorithms are rich - rich in application across a large and growing number of disciplines.'**

- David E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning

GA Readings

- Wendy Williams
  www.dbai.tuwien.ac.at/staff/musliu/ProblemSolvingAI/Class9GATutorial.ppt

- David E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989



- John H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, 1992