Εισαγωγή στους Εξελικτικούς Αλγόριθμους

Παναγιώτης Αδαμίδης
Email: adamidis@it.teithe.gr
URL: http://aetos.it.teithe.gr/~adamidis

Introduction to Evolutionary Computation

The EvoNet Flying Circus

Brought to you by *P. Adamidis*
The EvoNet Training Committee

Q What is the most powerful problem solver in the Universe?

A The (human) brain

that created "the wheel, New York, wars and so on" (after Douglas Adams)

A The evolution mechanism

that created human brain (after Darwin et al.)

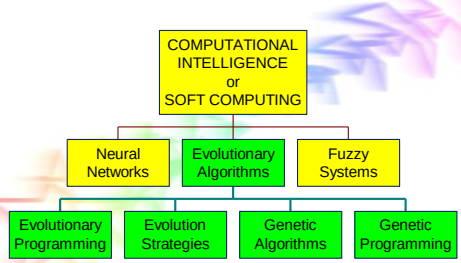
Building problem solvers by looking at and mimicking:

- brains → neurocomputing
- evolution → evolutionary computing

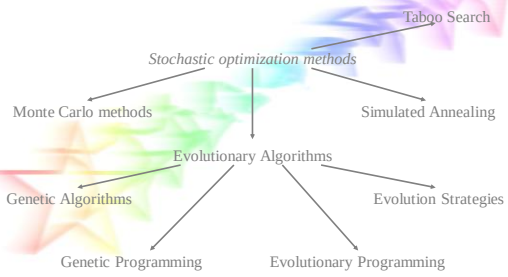
Table of Contents

- Taxonomy and History
- The Metaphor
- The Evolutionary Mechanism
- Domains of Application
- Performance
- Sources of Information

Taxonomy



Taxonomy (2)



History (1)

History (2)

- L. Fogel 1962 (San Diego, CA): *Evolutionary Programming*
- J. Holland 1962 (Ann Arbor, MI): *Genetic Algorithms*
- I. Rechenberg & H.-P. Schwefel 1965 (Berlin, Germany): *Evolution Strategies*
- J. Koza 1989 (Palo Alto, CA): *Genetic Programming*

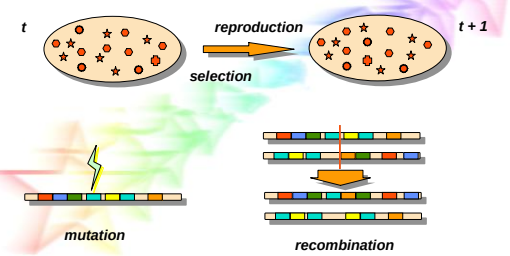
History (3)

1859 Charles Darwin: inheritance, variation, natural selection
1957 G. E. P. Box: random mutation & selection for optimization
1958 Fraser, Bremermann: computer simulation of evolution
1964 Rechenberg, Schwefel: mutation & selection
1966 Fogel et al.: evolving automata - "evolutionary programming"
1975 Holland: crossover, mutation & selection - "reproductive plan"
1975 De Jong: parameter optimization - "genetic algorithm"
1989 Goldberg: first textbook
1991 Davis: first handbook
1993 Koza: evolving LISP programs - "genetic programming"

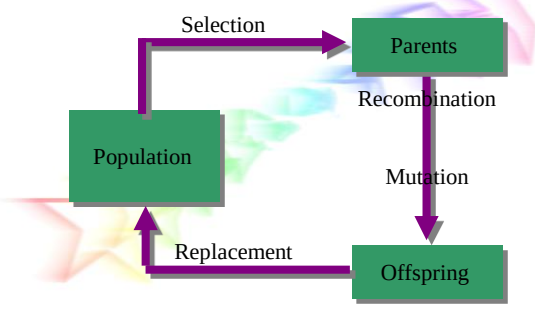
The Metaphor

EVOLUTION		PROBLEM SOLVING
Individual	⟷	Candidate Solution
Fitness	⟷	Quality
Environment	⟷	Problem

The Ingredients



The Evolutionary Cycle



Pseudocode

```
generation = 0;
SeedPopulation(popSize);          // at random or from a file
while(!TerminationCondition())
{
    generation = generation + 1;
    CalculateFitness();           // ... of new genotypes
    Selection();                  // select genotypes that will reproduce
    Crossover(p_cross);           // mate p_cross of them on average
    Mutation(p_mut);              // mutate all the offspring with Bernoulli
                                  // probability p_mut over genes
}
```

Evolution Programs

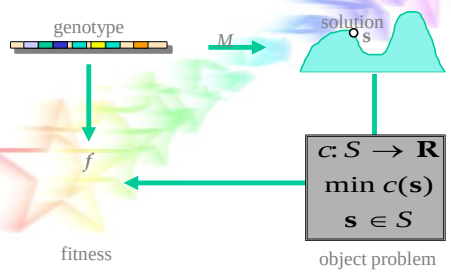
Slogan:

Genetic Algorithms + Data Structures = Evolution Programs

Key ideas:

- use a data structure as close as possible to object problem
- write appropriate genetic operators
- ensure that all genotypes correspond to feasible solutions
- ensure that genetic operators preserve feasibility

Object problem and Fitness



Κριτήρια Τερματισμού

- Μετά από ορισμένο αριθμό Γενεών
- Η ποιότητα του καλύτερου ατόμου δεν βελτιώνεται για ένα προκαθορισμένο αριθμό γενεών
- Εύρεση του ολικού βέλτιστου

The Evolution Mechanism

- Increasing diversity by genetic operators
  - mutation
  - recombination
- Decreasing diversity by selection
  - of parents
  - of survivors

Genetic-based Optimisation

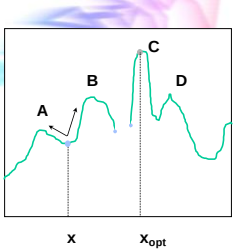
- Η διαδικασία βελτιστοποίησης στηρίζεται:
  - προνομιακή επιβίωση και αναπαραγωγή του καλύτερου (preferential survival and reproduction of the fittest)
  - διατήρηση ενός πληθυσμού που παρουσιάζει ποικιλία (maintenance of a population with diverse members)
  - κληρονομικότητα της γενετικής πληροφορίας των γονέων (inheritance of genetic information from parents)
  - περιστασιακή μετάλλαξη γονιδίων (occasional mutation of genes)

Domains of Application

- Numerical, Combinatorial Optimisation
- System Modeling and Identification
- Planning and Control
- Engineering Design
- Data Mining
- Machine Learning
- Artificial Life

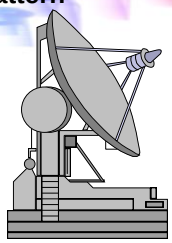
Domains of Application (2)

- Numerical optimization
  - Difficult functions
    - Multimodal
    - Discontinuous
    - Nonlinear
  - Parameter optimization
    - Structural optimization
    - Parametric design



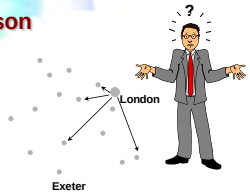
Domains of Application (3)

- Image processing and pattern recognition
  - Satellite image processing
  - Medical X-ray image processing
  - Facial composites of criminal suspects



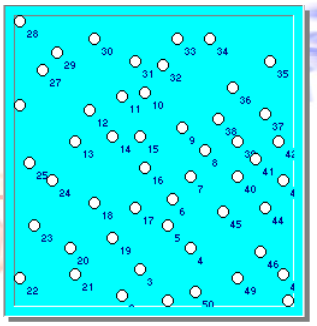
Domains of Application (4)

- Combinatorial optimization
  - Travelling salesperson
  - Scheduling
  - Networks
  - Bin packing and pallet loading



Travelling Salesperson

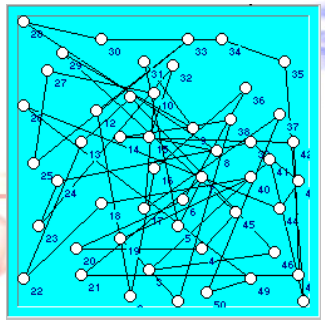
Example



- 50 cities
- Size = 50!

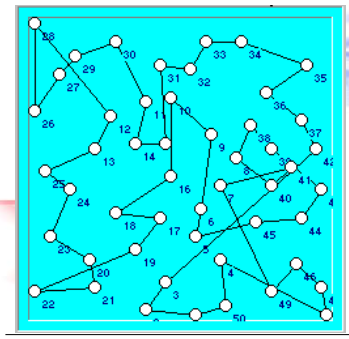
TSP Solver Demo v. 1.0 by V.V. Myagkih, O.V. Savelev and V.M. Kureichik

Travelling Salesperson (2)



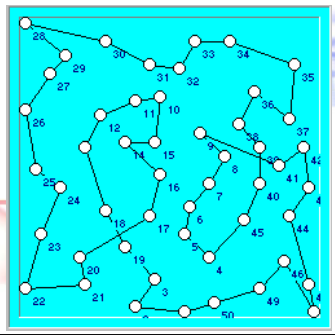
- Random start
- Not very good

Travelling Salesperson (3)



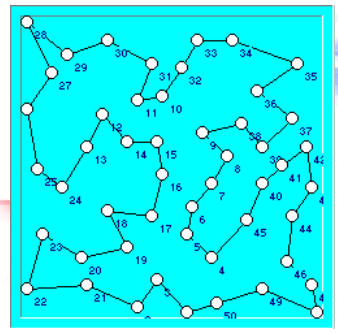
- After 50 generations
- Better solution
- Not perfect

Travelling Salesperson (4)



- After 800 generations
- Better solution
- Still few knots

Travelling Salesperson (5)



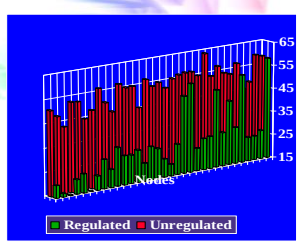
- After 50,000 generations
- Near-optimal solution
- No knots

Domains of Application (5)

- Design
  - Pipe network strengthening
  - Metro Toronto (Canada)
  - Problem size = $9.16 \cdot 10^{847}$
  - Solution cost:
    - Previous study: $123 Mil.
    - GA study: $110 Mil.
    - over 10% savings

Domains of Application (6)

- Control
  - Excess pressure minimization
  - $1.4 \cdot 10^{14}$ solutions
  - Solution
    - Before GA: 1257.5 m
    - GA: 473.5 m
    - over 60% reduction
  - 50,000 evaluations



Performance

- Acceptable performance at acceptable costs on a wide range of problems
- Intrinsic parallelism (robustness, fault tolerance)
- Superior to other techniques on complex problems with
  - lots of data, many free parameters
  - complex relationships between parameters
  - many (local) optima

Advantages

- No presumptions w.r.t. problem space
- Widely applicable
- Low development & application costs
- Easy to incorporate other methods
- Solutions are interpretable (unlike NN)
- Can be run interactively, accommodate user proposed solutions
- Provide many alternative solutions

Disadvantages

- No guarantee for optimal solution within finite time
- Weak theoretical basis
- May need parameter tuning
- Often computationally expensive, i.e. slow

Summary

EVOLUTIONARY COMPUTATION:

- is based on biological metaphors
- has great practical potentials
- is getting popular in many fields
- yields powerful, diverse applications
- gives high performance against low costs
- AND IT'S FUN !