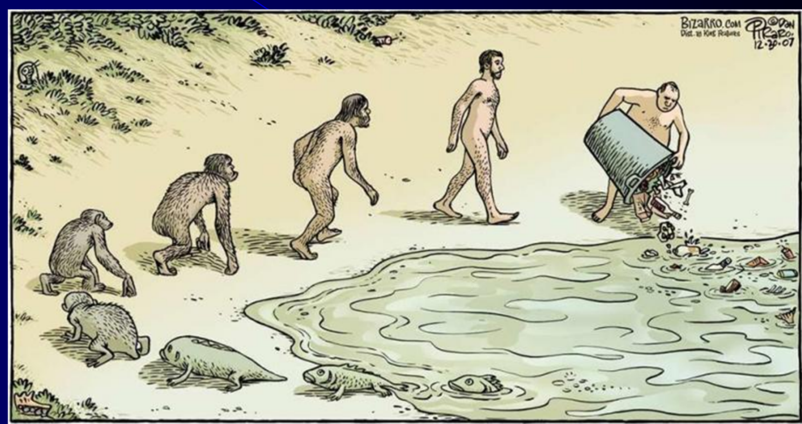
Optimization and Evolutionary Computing

Lecture 2



Outline

- Basic Optimization Background
- Introduction to Genetic Algorithm (GA)

2

Statement of an Optimization Problem

- An Optimization or a mathematical programming problem can be stated as follows

Find $X = \begin{Bmatrix} x_1 \\ \ldots \\ x_n \end{Bmatrix}$ which minimizes *f(X)*

Subject to the constraints

$g_j(X) \leq j = 1,2,\ldots, m$ and

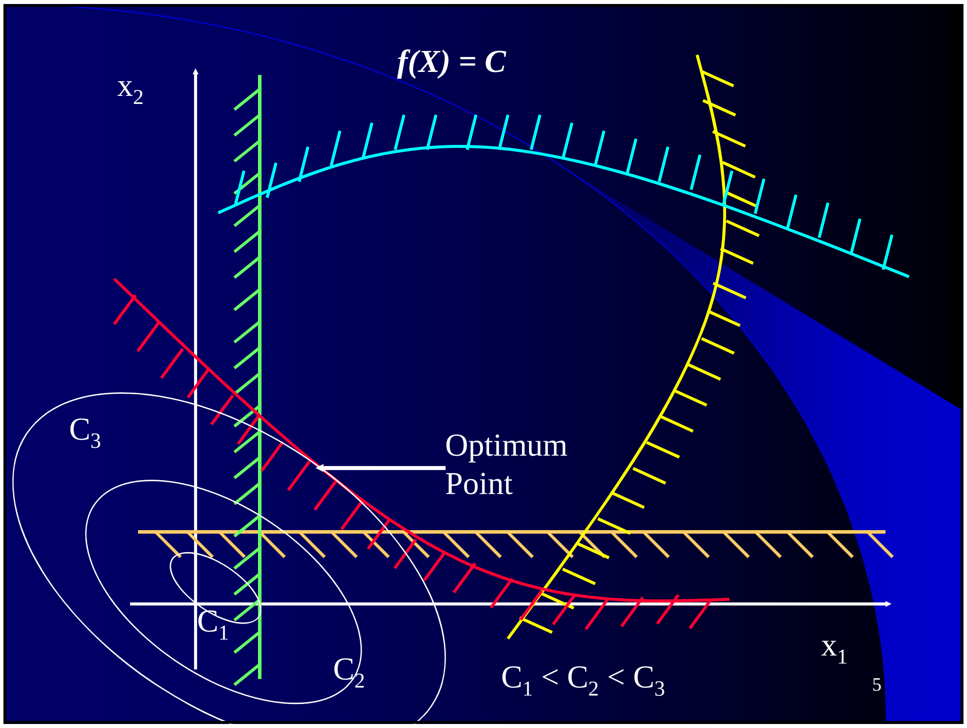
$L_j(X) = 0, j = 1,2,\ldots, p$

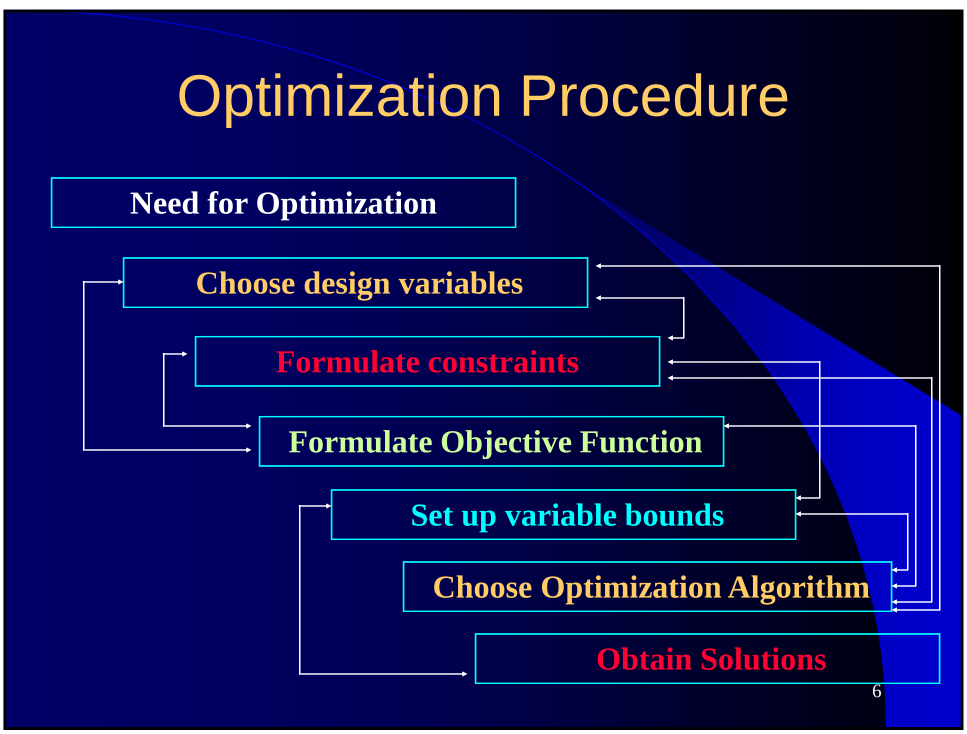
3

Definition of Terminology

- Design Vector
- Design Constraints
- Constraint Surface
- Objective Function

4


f(X) = C
x2
x1
C3
C1
C2
Optimum
Point
C1 < C2 < C3
5


Optimization Procedure
Need for Optimization
Choose design variables
Formulate constraints
Formulate Objective Function
Set up variable bounds
Choose Optimization Algorithm
Obtain Solutions
6

Nontraditional Optimization Algorithms

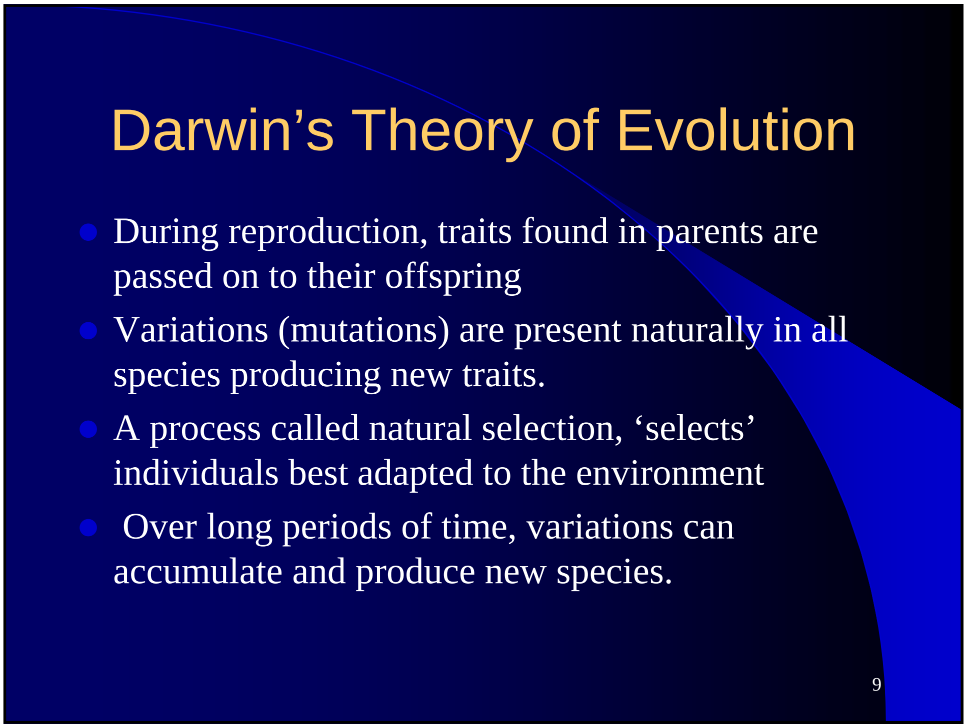
- Genetic Algorithms (GAs) - Mimic the principles of natural genetics and natural selection to constitute search and optimization procedures – Evolutionary Computing Algorithm
- Simulated Annealing – Mimics the cooling phenomenon of molten metals to constitute a search procedure

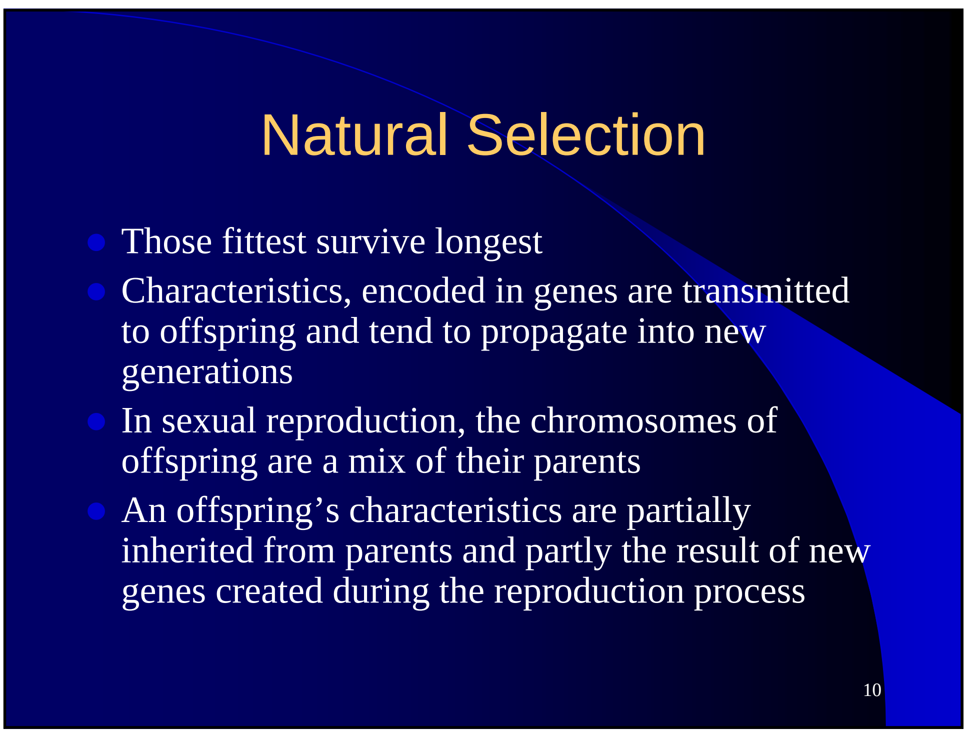
7

Background

- The field is based and inspired by observation and computer simulation of natural processes in the real world.
- Main inspiration
  - Darwin's theory of evolution
- Application of these ideas to complex optimisation problems and machine learning

8


Darwin's Theory of Evolution
During reproduction, traits found in parents are passed on to their offspring
Variations (mutations) are present naturally in all species producing new traits.
A process called natural selection, 'selects' individuals best adapted to the environment
Over long periods of time, variations can accumulate and produce new species.
9


Natural Selection
Those fittest survive longest
Characteristics, encoded in genes are transmitted to offspring and tend to propagate into new generations
In sexual reproduction, the chromosomes of offspring are a mix of their parents
An offspring's characteristics are partially inherited from parents and partly the result of new genes created during the reproduction process
10

Motivation

- Evolutionary Computing as Engineering
  – To find solutions to complex optimisation problems and machine learning problems.
- Evolutionary Computing as Science
  – To simulate real life phenomena
  (e.g. cellular automata, artificial life)

11

Nature-to-Computer Mapping

Nature	Computer
Individual	**Solution to a problem**
Population	**Set of solutions**
Fitness	**Quality of a solution**
Chromosome	**Encoding for a solution**
Gene	**Part of the encoding of a solution**
Crossover and Mutation	**Search operators**
Natural Selection	**Reuse of good (sub-) solutions**

12

Search Space

- The set of all possible encodings of solutions defines the search space.
- One measure of the complexity of the problem is the size of the search space.
- Crossover and mutation implement a pseudo-random walk through the search space.
- Walk is random because crossover and mutation are non-deterministic
- Walk is directed in the sense that the algorithm aims maximise quality of solutions using a fitness function.

13

Local and Global Search

- Local search
  - Looking for solutions near existing solutions in the search space
  - Crossover is the main operator for achieving this
- Global search
  - Looking at completely new areas of the search space.
  - Mutation is the main operator for achieving this.

14

A Generic Evolutionary Algorithm

- Initialise a population
- Evaluate a population
- While (termination condition not met) do
  - Select sub-population based on fitness
  - Produce offsprings of the population using crossover
  - Mutate offsprings stochastically
  - Select survivors based on fitness

15

Genetic Algorithm

- Directed search algorithms based on the mechanics of biological evolution
- 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

16

Genetic Algorithm (Contd.)

- Provide efficient, effective techniques for optimization and machine learning applications
- Widely-used today in business, scientific and engineering circles

17

Classes of Search Techniques

- Search techniques
  - Calculus-based techniques
    - Direct methods
      - Finonacci
      - Newton
    - Indirect methods
  - Guided random search techniques
    - Evolutionary algorithms
      - Evolutionary strategies
      - Genetic algorithms
        - Parallel
          - Centralized
          - Distributed
        - Sequential
          - Steady-state
          - Generational
    - Simulated annealing
  - Enumerative techniques
    - Dynamic programming

18

Working Principle

- Let us consider following maximization problem
- Maximize $f(x), (x_i)^L \leq x_i \leq (x_i)^U, i=1,2,\ldots n$

19

Coding

- Variables x_i's are coded in some string structures
- The coding of variables is not absolutely necessary
- Binary-coded strings having 1's and 0's are mostly used.
- Length of the string is determined according to the desired solution accuracy.

20

Coding (Contd.)

- For example, if 4-bits are used to code each variable in two variable function optimization problem, the strings (0000 0000) and (1111, 1111) would represent the points

  $((x_1)^L \ (x_2)^L \ )^T \qquad ((x_1)^U \ (x_2)^U \ )^T$

21

Coding (Contd.)

- Any other eight-bit string can be found to represent a point in the search space according to a fixed mapping rule.
- Usually following linear mapping rule is used

$$x_i = x_i^{(L)} + \frac{x_i^{(U)} - x_i^{(L)}}{2^{l_i} - 1}(decoded\_value\_of\_S_i)$$

22

Coding (contd.)

- The variable x_i is coded in a sub-string s_i of length $l_{i.}$
- The decoded value of binary sub-string s_i is calculated as

$$\sum_{i=0}^{l-1} 2^i s_i$$

- Where $s_i \in (0,1)$ and the string s is represented as $(s_{l-1}\, s_{l-2} \ldots\ldots s_2\, s_1\, s_{0})$

23

Coding (Contd.)

- Example, four-bit string (0111) has decoded value equal to

  $((1)2^0 + (1)2^1 + (1)2^2 + (0)2^3)$ or 7
- With four bits to code each variable, there are only 2^4 or 16 distinct sub-strings possible.
- The possible accuracy $1/16^{th}$ of the search space
- It is not necessary to code all variables in equal sub-string length.

24

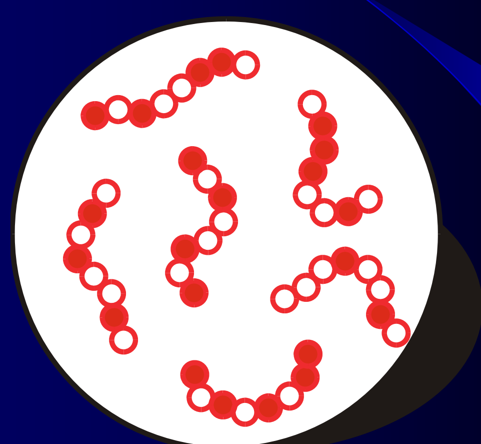
Coding (Contd.)

- Generalization - l_i - bit coding for a variable, the obtainable accuracy in that variable is approximately

$$\frac{x_i^{(U)} - x_i^{(L)}}{2^{l_i}}$$

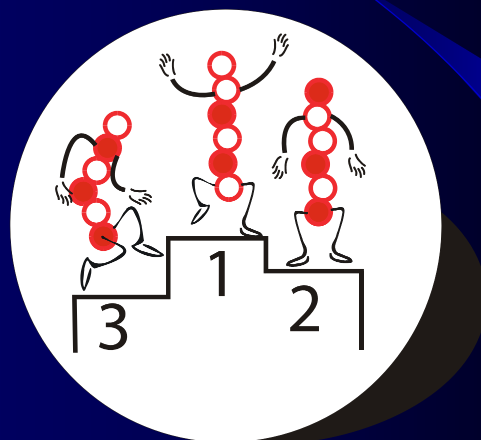
25

A "Population"



26

Ranking by Fitness



27

Fitness Function

- Fitness function **F(x)** is first derived from the objective function and used in successive genetic operations.
- For maximization problem, the fitness function can be considered as a the objective function **F(x)** = *f(x)*.

28

Fitness Function (Contd.)

- Minimization Problem → Convert to maximization problem using suitable transformation → Optimum point remains unchanged
- Commonly used fitness function

$$F(x) = 1/(1+f(x))$$

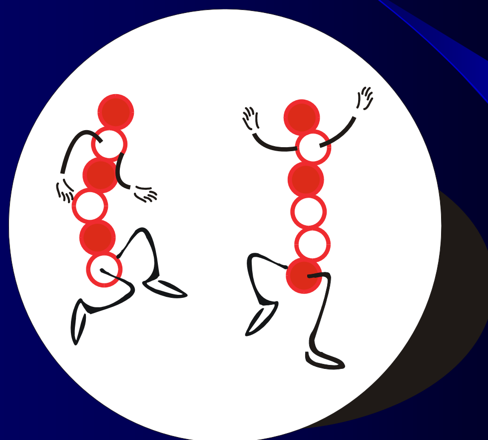
29

GA Operators

- Reproduction
- Crossover
- Mutation

30

Reproduction



31

Reproduction

- Selects the good strings in population and forms a mating pool.
- Also, known as **Selection Operator**
- Exist **many types of reproduction operators** in GA literature.
- Essential idea – Above –average strings must be picked from **current population** and their **multiple copies** must inserted in the **mating pool** in a **probabilistic manner**

32

Reproduction (Contd.)

- Commonly used approach -

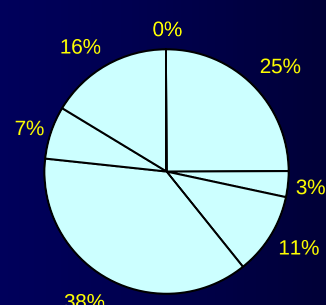
  i- th string in the population is selected with a probability proportional to $\mathbf{F_i}$
- Population size is usually kept fixed in a simple GA, the sum of probability of each string being selected for the mating pool must be ONE.
- Probability for i- the string is

$$p_i = \frac{F_i}{\sum_{j=1}^{n} F_j}$$

33

Reproduction (Contd.)

- Simple selection approach – Roulette-wheel with its circumference marked for each string proportionate to the string’s fitness.



34

Reproduction (Contd.)

- Since the circumference of the wheel is marked according to a string's fitness, this roulette-wheel mechanism is expected to make $\mathbf{F_i} / \underline{\mathbf{F}}$ copies of the i- th string in the mating pool.
- The average fitness of the population is calculated as

$$\underline{\mathbf{F}} = \Sigma\, \mathbf{F_i} / n$$

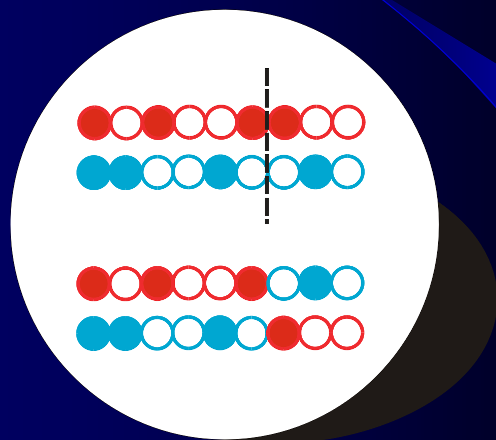
35

Reproduction (contd.)

- Good strings in a population are probabilistically assigned a larger number of copies and a mating pool is formed.
- No new strings are formed in reproduction phase.

36

Crossover



37

Crossover (Contd.)

- New strings are created by exchanging information among strings of the mating pool.
- Many approaches are available in literature.
- Common Approach – Two strings are picked from the mating pool at random and some portions of the strings are exchanged between the strings.

38

Crossover (Contd.)

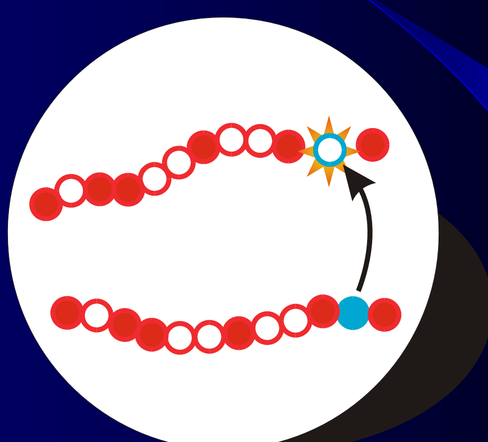
P1 (0 1 1 0 1 0 0 0) → (0 1 0 0 1 0 0 0) C1
P2 (1 1 0 1 1 0 1 0) → (1 1 1 1 1 0 1 0) C2

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)

39

Mutation



40

Mutation (Contd.)

- The need for mutation is to create a point in the neighborhood of the current point, thereby achieving a local search around the current solution.
- Mutation is also used to maintain diversity in the population.

41

Mutation (Contd.)

- Example – Four eight-bit strings
  - 0110 1011
  - 0011 1101
  - 0001 0110
  - 0111 1100
- If true optimum requires 1 in 0 position, then neither reproduction nor crossover operator will be able to create 1 in that position.

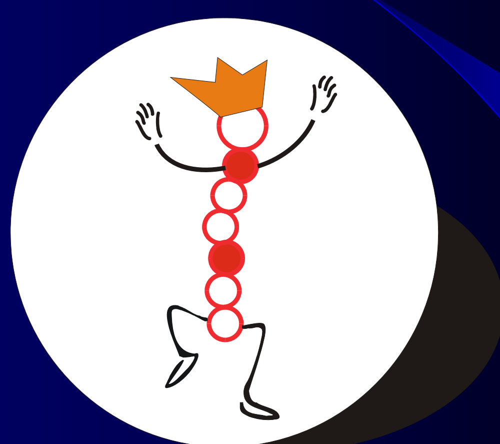
42

Summary of GA Operators

- Reproduction Operator – Selects good strings
- Crossover Operator – Recombines good sub-strings from good strings together to hopefully create a better sub-string
- Mutation Operator – Alters a string locally to obtain better sub-string

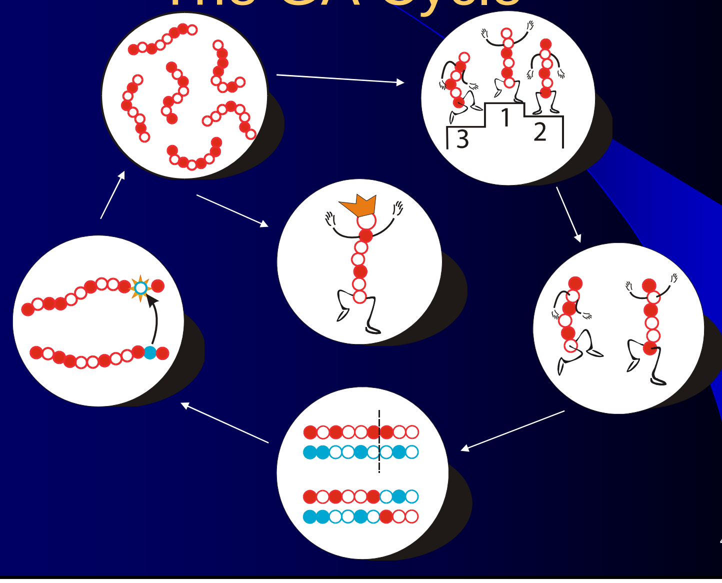
43

Best Design



44

The GA Cycle

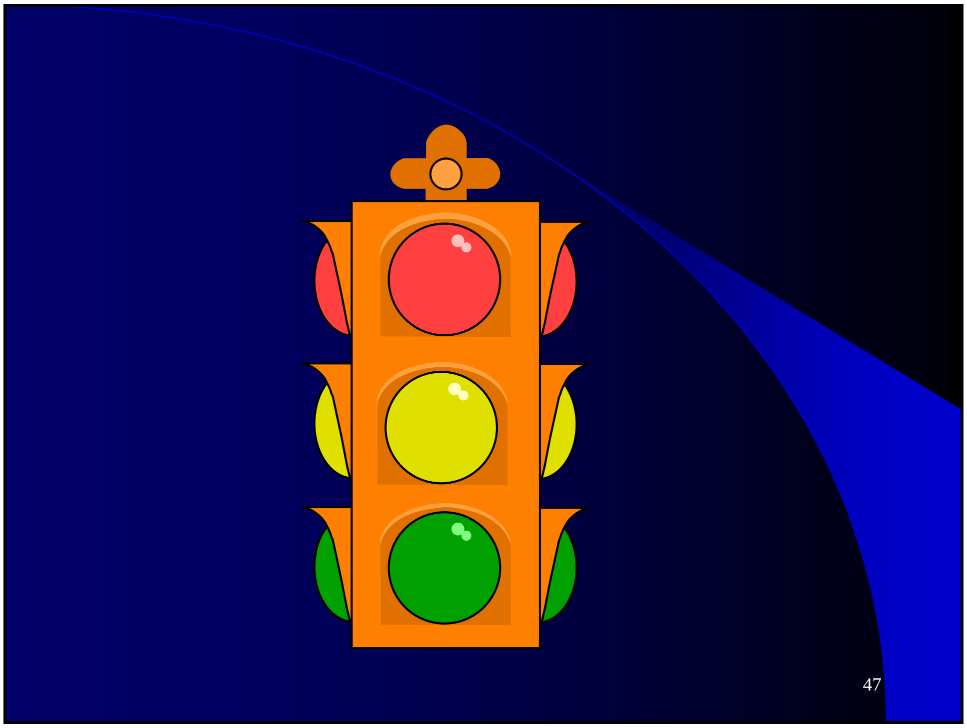


45

Mathematical optimization

- At the end of his course on mathematical methods in optimization, the professor sternly looks at his students and says: 'There is one final piece of advice I'm going to give you now: Whatever you have learned in my course - never ever try to apply it to your personal lives!'
'Why?' the students ask.
'Well, some years ago, I observed my wife preparing breakfast, and I noticed that she wasted a lot of time walking back and forth in the kitchen. So, I went to work, optimized the whole procedure, and told my wife about it.'
'And what happened?!'
'Before I applied my expert knowledge, my wife needed about half an hour to prepare breakfast for the two of us. And now, it takes me less than fifteen minutes...'

46


47