Evolutionary Computation & DNA Computing

Poras Bharucha
What is Evolutionary Computation?

- Sub field of Computational Intelligence.
- Uses iterative progress, such as growth or development in population.
- The population is selected in guided random search to obtain desired results.
Evolutionary Techniques

- There different evolutionary techniques, such as:
  - Evolutionary Algorithms
  - Swarm Intelligence
  - Differential Evolution
  - Cultural Algorithms
  - Harmony Search
Evolutionary Algorithms

Random Input Population

Computational Intelligence

Evolutionary Computation

Evolutionary Algorithms

Biological Evolution

New Generation Population

Crossover

Mutation

New Input

011010 01110010

011000

1100
Evolutionary Algorithm Techniques

- Genetic Algorithms
- Evolutionary Algorithms
- Learning Classifier System
- Evolution Strategy
- Genetic Programming
- Evolutionary Programming
Genetic Algorithms

- Most popular amongst Evolutionary Algorithms
- Uses techniques inspired by evolutionary biology
- Used for optimization problems
- Finds approximate or exact solution to the problem
Advantages of Genetic Algorithms

- Optimises the problem efficiently
- Can deal with a wide range of problem
- Finds acceptably good solution quickly
The Process of Genetic Algorithm

1. Create a population
2. Determine Fitness of each individual
3. Select the individuals
4. Perform Crossover
5. Perform Mutation
6. Condition Satisfied
7. Display Result
8. Next Generation
Step 1: - Create Population

- Selects random input from initial solutions
- Size of population: the type of problem
- Tries to cover entire range of possible solution
- Ex: - A person has to reach a destination, there are several routes to the destination, need to find the best route
- So the input would be randomly selected different possible routes to the destination
Step 2: - Define the fitness of each individuals

- Fitness function judges the fitness of each individual
- Individual score depends on the fitness
- Fitness function is decided according to the problem
- Ex: - short and less costly route - high score
- long and costly - low score
Step 3: - Selecting Individuals

- Individuals are selected for breeding

- Generic selection algorithms used for selection

- Different generic algorithms: -
  
  - Roulette Wheel Selection/Fitness Proportionate Selection
  
  - Tournament Selection
Roulette Wheel Selection/Fitness proportionate Selection

- Similar to pie chart
- More fit the candidate – Higher is the probability of selection
- Weak candidate – Lower probability of selection
- Process repeated = No. of candidates
- Selection of fittest candidate not guaranteed

The wheel is rotated and candidate at selection point is selected.
Tournament Selection

- Tournament as the word suggest

- Candidates are selected randomly

- Candidate with best fitness - Winner

- Prize = Winner is selected for further process

<table>
<thead>
<tr>
<th>Candidates</th>
<th>Fitness Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Candidate 1</td>
<td>2</td>
</tr>
<tr>
<td>Candidate 2</td>
<td>5</td>
</tr>
<tr>
<td>Candidate 3</td>
<td>3</td>
</tr>
<tr>
<td>Candidate 4</td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Candidates Selected</th>
<th>Fitness Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Candidate 2</td>
<td>5</td>
</tr>
<tr>
<td>Candidate 3</td>
<td>3</td>
</tr>
</tbody>
</table>

= Candidate 2
Step 4: - Crossover

- Two candidates from selected pool are selected randomly
- Random Value between 0 - 1 is generated
- Checked with probability of crossover i.e. 0.7
- Random value > 0.7 = No Crossover
- Random value < 0.7 = Crossover
- Resultant offspring can be weak or strong
Different types of crossover techniques are there

- **One-point crossover**
  - Single crossover point is selected randomly
  - Data beyond that is swapped

Ex: -

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

One-Point Crossover
Crossover Continued....... 

- **Two-point crossover**
  - Two points are selected randomly
  - Data between two points is swapped

Ex: -

```
0 0 1 0 1 1 1 1 0 1 0 1 1 0 1 1

1 1 1 0 1 0 0 0 1 1 0 0 0 0 0 1

0 0 1 0 1 0 0 0 0 1 0 1 1 0 1 1

1 1 1 0 1 1 1 1 1 1 0 0 0 0 0 1
```
Step 5: - Mutation

- Done to maintain genetic diversity

- Arbitrary bit is changed from original state

- Mutation occurs infrequently

- Random value between 0-1 is generated

- Checked with probability of mutation i.e. 0.001
Mutation Continued......

- Random value > 0.001 = No Mutation
- Random value < 0.001 = Mutation
- May result in weak or strong offspring

Ex: -

```
0 0 1 0 1 0 0 0 0 1 0 1 1 0 1 1
```

```
0 0 1 0 0 0 0 0 0 1 0 1 1 0 1 1
```
Genetic Algorithm Continued......

- New generation = New set of input
- Entire process is repeated
- Terminated when desired result is achieved
DNA Computing Implementing Genetic Algorithms (1999)

Junghuei Chen, Eugene Antipov, Bertrand Lemieux, Walter Cedeño, David Harlan Wood
Main Challenge

- Select DNA strands
- Physically separate them according to their fitness
Advantages of implementing genetic algorithms in DNA

- Genetic algorithms generally manipulate population based on bit strings
- Uses both crossover and mutation
- Computing time using DNA is proportional to number of generations
- No. of generation reduced – point wise mutation and crossover
Continued......

- Process population that is very large
- Resulting larger genetic variations
- Larger genetic variations – higher fitness level
- Few generations
Continued......

- Massive information storage using DNA
- Better than using deterministic algorithm, as even error contributes
- Implementing crossover are possible as variations of Sexual PCR
- Works faster than supercomputer
Problems for DNA genetic algorithms

- There are three problems for DNA genetic algorithms
  - The MAX 1s Problem
  - The Royal Road Problem
  - The Cold War Problem
The Royal Road Problem

- **Target Strand** – Fixed length, N blocks

- **Contribution from candidate** – Perfect Match to corresponding block

- **Fitness** – Number of Perfectly matched blocks
  
  = Find Some blocks that perfectly match the target block
Continued......

Ex: -

Block 1

A C G A
T G C T

Block 2

T C C T
A G G T

Target Strand

Candidate Strand

Block 1 – Perfect Match, Fitness Score – 1
The Cold War Problem

- Different then MAX 1s & Royal Road Problem
- The target of the problem is not known
- Two sets of population evolve simultaneously i.e. Target & Candidate
- Calculate Fitness – Target & Candidate pairs
Continued......

- Separate Candidate from Target
- Breed candidate
- Breed Target
- Combine the offspring
Continued......

Ex:

Pair 1

ACGAT
TGCT

Pair 2

TCCTT
AGGTA

Pair 3

TCGAT
AGCT

Pair 1

ACGAT
TGCT

Pair 3

TCGAT
AGCT

Target Strands

Candidate Strands

Selected For Breeding

Denature

ACGAT
TGCT

New Offspring

ACGAT
TGCT
The MAX 1s Problem

- Input – Randomly selected strands
- Target is known
- Strands are of fixed length
- Target – Generate more fit strands
Process of MAX 1s

- Evaluate fitness by hybridizing
- Separate them physically
- Select fit Candidates & purify them for breeding
- Amplify with point wise mutation and reserve
Continued......

- Apply crossover

- Combine reserve and generate new offspring

- Use new generation as input

- Terminate – target achieved
Process of Extracting DNA Physically & Generating New Offspring
Design

- Target strand is designed

```
\-----------TARGET-----------
\----------CG-CLAMP---------/------ 80 A'g-------\n5' -> CGCCCTCCGGCCGCAGGCACGGCCGCAAAAAAA.........AAAAAAA -> 3'
3' <- GCCGCGGCGGGCCAGGCGGGGTTTTTTTTTTTTTTTTTTTGTGATCGCTCAGCATAAT <- 5'
\----------CG-CLAMP COMPL----/------- 80 T'g-------/------ TAIL ------/
\----------------PERFECT CANDIDATE------------------------/
```

Target Strand designed for experiment
Continued......

- CG-Clamp avoids strands hybridizing to itself
- Candidate strands are extended by tail at 5’ end
- Candidate strands have known primer sites at both ends
- Imperfect candidate does not have 80 consecutive T’s
Fitness Evaluation and Separation

- 2-dimensional Denaturing Gradient Gel Electrophoresis (DGGE) used to evaluate fitness

- The strands are placed at the top of the gel

- Electric charge – Strands travel vertically downwards

- Speed of migration – Depends on the hybridization
Continued......

- **Left** – No Denaturation

- **Centre** – Intermediate Denaturation

- **Right** – Denaturation
Selection

- Fittest candidate are selected
- Fittest candidates – lowest on vertical line
- Purification – Target strands are separated from candidate strands
- Purified candidates are amplified
Continued......

- Point wise mutation can also be induced

- PCR primer – Phosphorylated at 5’ end

- Resultant strands – Portion reserved, remaining used for breeding
Single Point Crossover

- Portion selected for breeding – Nicked using DNase I
- Similar amount of unnicked strands from reserved portion is introduced
- The mixture is denatured and then reannealed
- Several possible configuration will be formed
Some intact complement of candidate strand enforces alignment

Partial candidate strand is extended adding DNA polymerase

After this the remaining portion is combined with reaction products

Complementary strands are digested by -exonuclease

Purification by length – Using Denaturing Gel Electrophoresis
Thank You!!!!!