Genetic Algorithms

Part 2: What is a Genetic Algorithm?

Fall 2009

Instructor: Dr. Masoud Yaghini

Outline

- Definition
- Genetic Algorithm Phases
- Simple Genetic Algorithm (SGA)
- References

Definition

Definition

- A population of individuals exists in an environment with limited resources
- **Competition** for those resources causes selection of those **fitter individuals** that are better adapted to the environment
- These individuals act as seeds for the generation of new individuals through recombination and mutation
- The new individuals have their fitness evaluated and compete for survival.
- Over time **natural selection** causes a rise in the fitness of the population

Definition

- **Genetic Algorithms** are
  - **Bio-Inspired artificial intelligence class,**
  - **stochastic**,
  - **population-based** algorithms

- Typically applied to:
  - hard problems with a large search space
  - discrete optimization

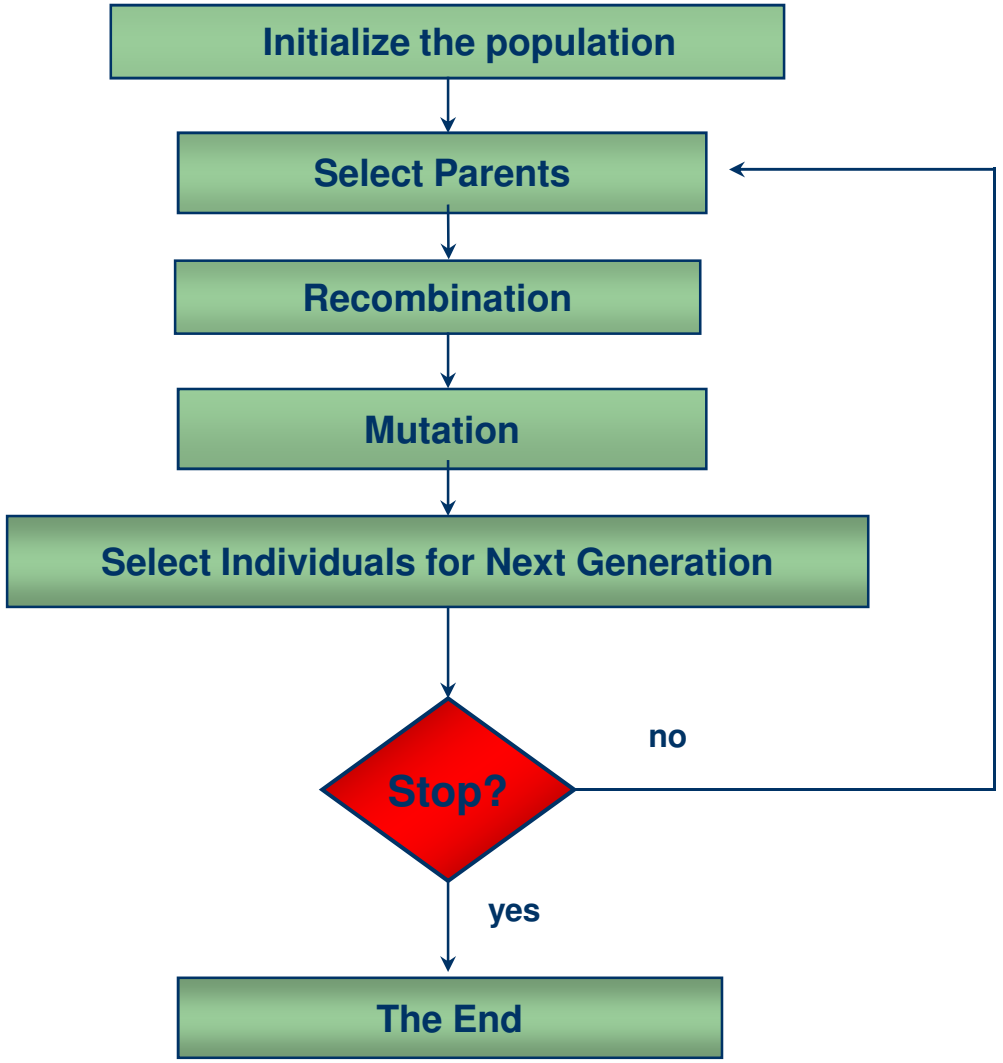
- Developed by **John Holland**, USA in the 1970's

Genetic Algorithm Phases

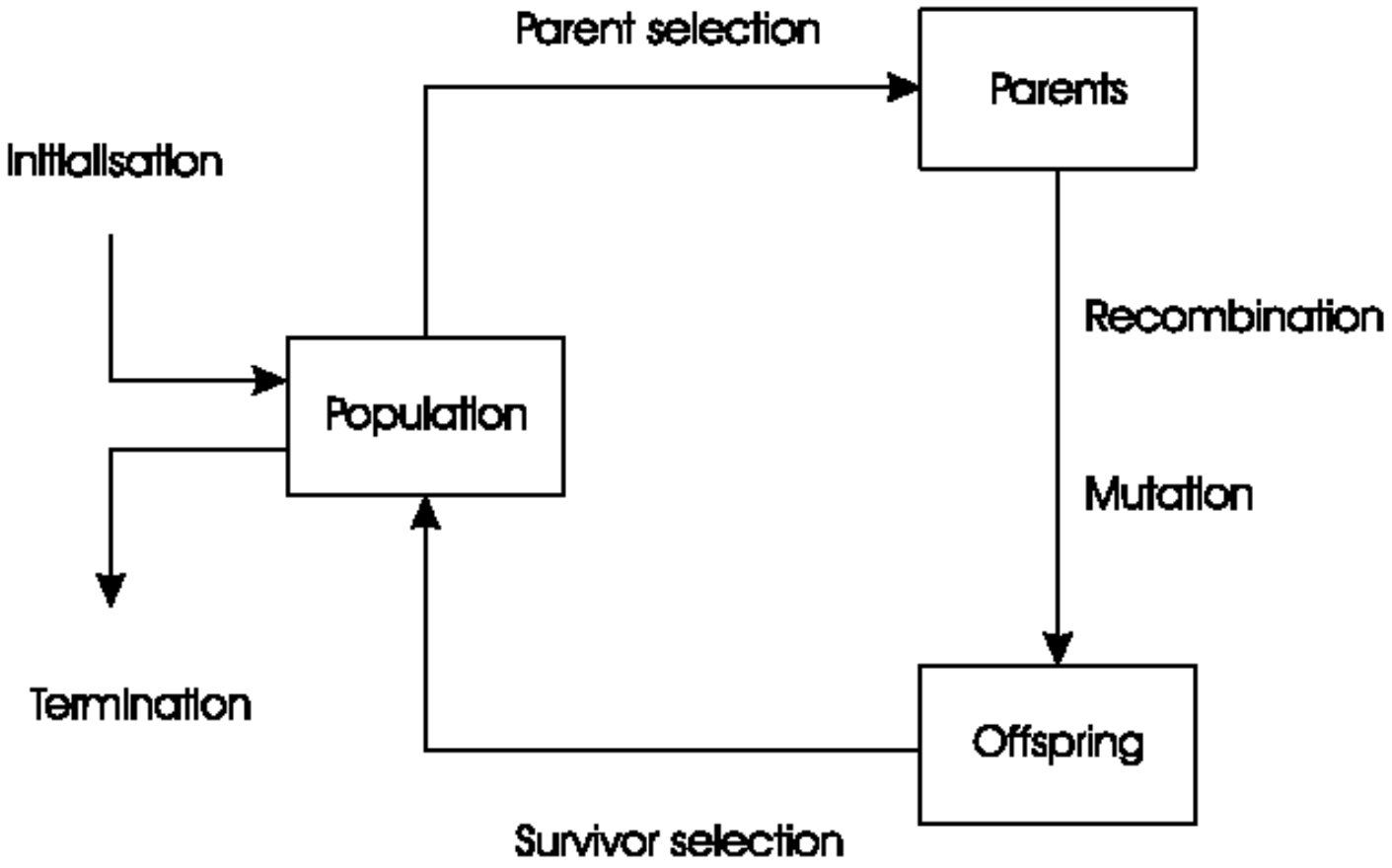
Pseudo-code for typical GA

```
BEGIN
  INITIALISE population with random candidate solutions;
  EVALUATE each candidate;
  REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
    1 SELECT parents;
    2 RECOMBINE pairs of parents;
    3 MUTATE the resulting offspring;
    4 EVALUATE new candidates;
    5 SELECT individuals for the next generation;
  OD
END
```

GA Algorithmic Phases



General Scheme of GA



Simple Genetic Algorithm (SGA)

Simple Genetic Algorithm (SGA)

- Holland's original GA is now known as the **simple genetic algorithm** (SGA)

- Other GAs use different:
  - Representations
  - Mutations
  - Crossovers
  - Selection mechanisms

SGA summary

Representation	Binary strings
Recombination	1-point crossover
Mutation	bit-flipping with fixed probability
Parent selection	Fitness-Proportionate
Survivor selection	All children replace parents
Speciality	Emphasis on crossover

Simple example – $f(x) = x^2$

- **Finding the maximum of a function:**
  - **$f(x) = x^2$**
  - Range [0, 31] → Goal: find max ($31^2 = 961$)
  - Binary representation: string length 5 = 32 numbers (0-31)

genotype	0 0 1 0 1	
mapping	2^4 2^3 2^2 2^1 2^0 16 8 4 2 1	
phenotype	0*16+0*8+1*4+0*2+1*1 = 5	
fitness	25	= f(x)

x^2 example

- **x^2 example:**
  - Representation: Binary code
  - Population size: 4
  - Recombination: 1-point crossover
  - Mutation: Bit-flipping with fixed probability
  - Parent selection: Fitness-Proportionate
  - Initialization: Random
- We show one generational cycle done by hand

x^2 example: selection

String no.	Initial population	x Value	Fitness $f(x) = x^2$	$Prob_i$	Expected count	Actual count
1	0 1 1 0 1	13	169	0.14	0.58	1
2	1 1 0 0 0	24	576	0.49	1.97	2
3	0 1 0 0 0	8	64	0.06	0.22	0
4	1 0 0 1 1	19	361	0.31	1.23	1
Sum			1170	1.00	4.00	4
Average			293	0.25	1.00	1
Max			576	0.49	1.97	2

X^2 example: crossover

String no.	Mating pool	Crossover point	Offspring after xover	x Value	Fitness $f(x) = x^2$
1	0 1 1 0 | 1	4	0 1 1 0 0	12	144
2	1 1 0 0 | 0	4	1 1 0 0 1	25	625
2	1 1 | 0 0 0	2	1 1 0 1 1	27	729
4	1 0 | 0 1 1	2	1 0 0 0 0	16	256
Sum					1754
Average					439
Max					729

X^2 example: mutation

String no.	Offspring after xover	Offspring after mutation	x Value	Fitness $f(x) = x^2$
1	0 1 1 0 0	1 1 1 0 0	26	676
2	1 1 0 0 1	1 1 0 0 1	25	625
2	1 1 0 1 1	1 1 0 1 1	27	729
4	1 0 0 0 0	1 0 1 0 0	18	324
Sum				2354
Average				588.5
Max				729

The simple GA

- SGA Shows many **shortcomings**:
  - Representation is too restrictive
  - Mutation & crossovers only applicable for bit-string & integer representations
  - Selection mechanism sensitive for converging populations with close fitness values
  - Generational population model (step 5 in SGA) can be improved with explicit survivor selection

References

References

- Eiben and Smith. **Introduction to Evolutionary Computing**, Springer-Verlag, New York, 2003.
- J. Dreo A. Petrowski, P. Siarry E. Taillard, **Metaheuristics for Hard Optimization**, Springer-Verlag, 2006.

The End