Evolutionary Algorithms

CS367

Patricia J Riddle

Biological Foundations

- Each cell of an organism contains a complete set of genes.
- Genotype is the collection of genes of an organism.
- Phenotype is the morphology, physiology and behavior of an organism.
- Phenotype is determined by genotype and learning/interaction with the environment.

Biological Foundations II

- Variety is manifested as variation in the chromosomes.

- Chromosomes are DNA molecules consisting of nucleotides:
  - A (adenine)
  - C (cytosine)
  - G (guanine)
  - T (thymine)

- Chromosomes can be interpreted as character strings in nature's base-4 alphabet.

- The evolutionary process performs genetic operations on these.

Biological Foundations III

- Fitness is determined by how adapted an organism is to its environment.
- Survival of the fittest and natural selection [Darwin, 1859].
- Evolution is a search process for phenotypes.
- There is a tendency towards phenotypes that are more adapted to the environment.
- Variation in genotypes results in variation in phenotype.

Necessary Conditions

- An entity has the ability to reproduce itself.
- There is a population of such self-reproducing entities.
- There is some variety among the self-reproducing entities.
- Some difference in ability to survive in the environment is associated with the variety.

Genetic Algorithms

- Genetic algorithms are search algorithms inspired by evolutionary processes.
- They are highly parallel mathematical algorithms.
- They transform a set (population) of individual mathematical objects (typically fixed length character strings) into a new population.
- Each individual is associated with a fitness value, which plays a significant role in the transformation process.
- First proposed in the 1970s, but still no adequate theory to explain why they work so well. - see page 19

Sketch of the Genetic Algorithm

- Population of solutions (initialised randomly).
- Variation operators (e.g. mutation and crossover).
- The goodness of a solutions is quantified using a fitness function.
- Selection based on fitness.
- Repeat for a number of generations.

Fitness-Proportionate Reproduction

- Copy individuals in the current population into the next generation with a probability proportional to their fitness.

- The effect of fitness-proportionate reproduction is to improve the average fitness of the population.

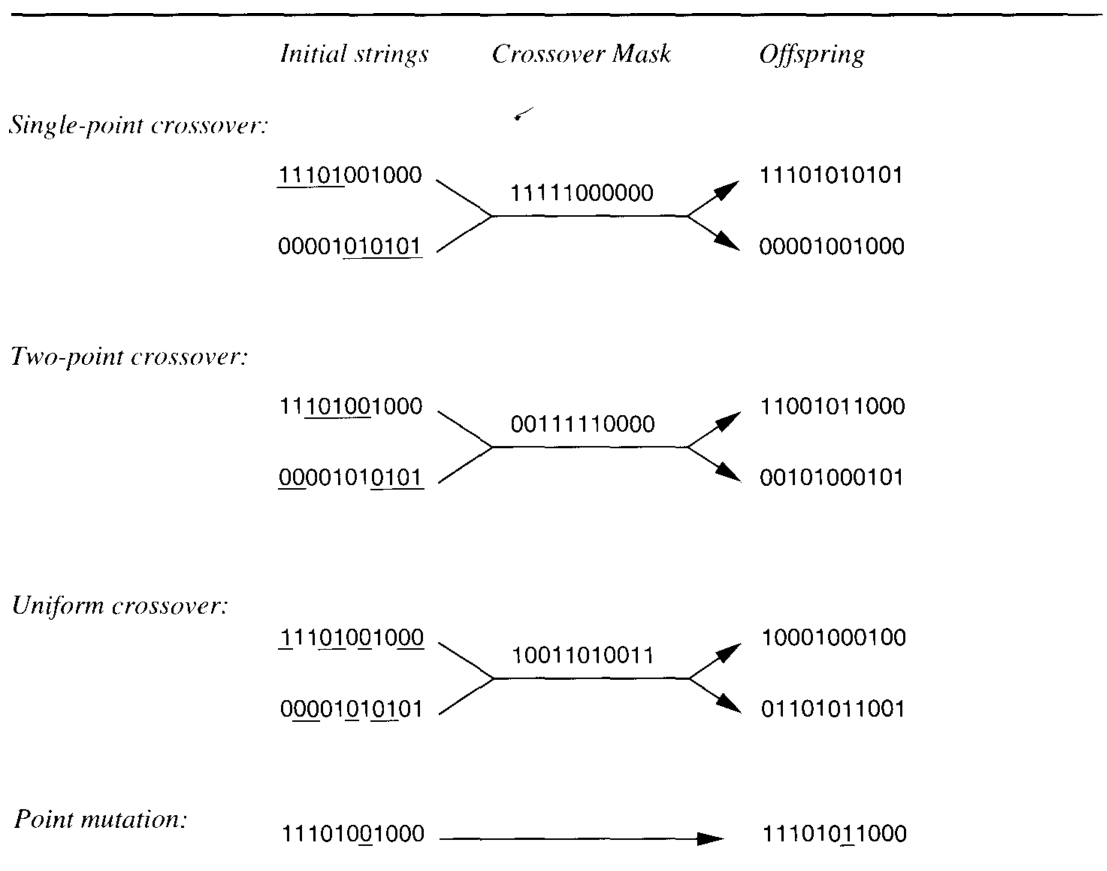
Crossover

- Select a number between 1 and $L - 1$; this number becomes the crossover point.
- Take two parents and cut them at the crossover point.
- Combine the crossover fragment of parent 1 with the remainder of parent 2.
- Combine the crossover fragment of parent 2 with the remainder of parent 1.

The Genetic Algorithm

1. Randomly create an initial population of individual fixed-length strings.
2. Iteratively perform the following substeps on the population of strings until the termination criterion has been satisfied:
   - Evaluate the fitness of each individual in the population.
   - Create a new population of strings by applying at least the first two of the following three operations:
     - Copy existing strings to the new population (mating pool).
     - Create two new strings by genetically recombining randomly chosen substrings from two existing strings (crossover).
     - Create a new string from an existing string by randomly mutating the character at one position in the string (mutation).
3. The best individual string that appeared in any generation is designated as the result of the genetic algorithm for the run.

Genetic Operators



The Hamburger Restaurant Problem

- Find the business strategy for a chain of four hamburger restaurants that maximizes the profit (optimization problem).
- The strategy consists of making three binary decisions:
  - **Price** - Should the price of the hamburger be 50 cents or $10?
  - **Drink** - Should wine or cola be served with the hamburger?
  - **Speed of service** - Should the restaurant provide slow, leisurely service by waiters in tuxedos or fast, snappy service by waiters in white polyester uniforms?

Representing the Problem

- There are three decision variables.
- Each variable can assume one of two possible values.
- Therefore it would be natural to represent each possible business strategy as a character string of length $L = 3$ over an alphabet of size $K = 2$.
- This yields a search space of 2 3 = 8 possible business strategies.

Example of Business Strategies

Restaurant	Price	Drink	Speed	Representation
1	high	cola	fast	011
2	high	wine	fast	001
3	low	cola	leisurely	110
4	high	cola	leisurely	010

The Rich Uncle's Heir

- No guidance as to what business strategy produces the highest payoff in the environment in which the restaurants operate.
- No information about which of the three variables are the most important:
  - Are all variables relevant?
  - Can the variables be changed independently or are they interrelated?
  - Can a global optimum be obtained by a stepwise procedure of varying one variable at a time?

More about Rich Uncle's Heir

- No knowledge about the magnitude of the maximum profit when making the right decision or the loss when making the wrong decision.

- No insight into how the operating environment changes over time:
  - Are the public's tastes fickle?
  - Do the rules of the game change?

The Approach

- Test a different initial random strategy in each of the four restaurants for one week.

- Evaluate the feedback given by the managers of the restaurants.

Terminology

Restaurants	**Population**
Number of restaurants	Population size
Feedback	Fitness value
Strategies of the week	Generation

The Initial Generation

	Generation 0		
i	String X_i	Fitness $f(X_i)$	$F(X_i)/\Sigma f(X_i)$
1	011	3	.25
2	001	1	.08
3	110	6	.50
4	010	2	.17
total		12	
worst		1	
average		3	
best		6	

Theory of GAs

- Cameron Skinner
  - 2pt crossover
  - 3 tournament selection

  - Discovery & Retention
  - How handling overcrowding?
    - Use seeding algorithm

  - When should you us a GA?

Overcrowding

- Here the whole population becomes copies and mutations of that one guy in the front row.

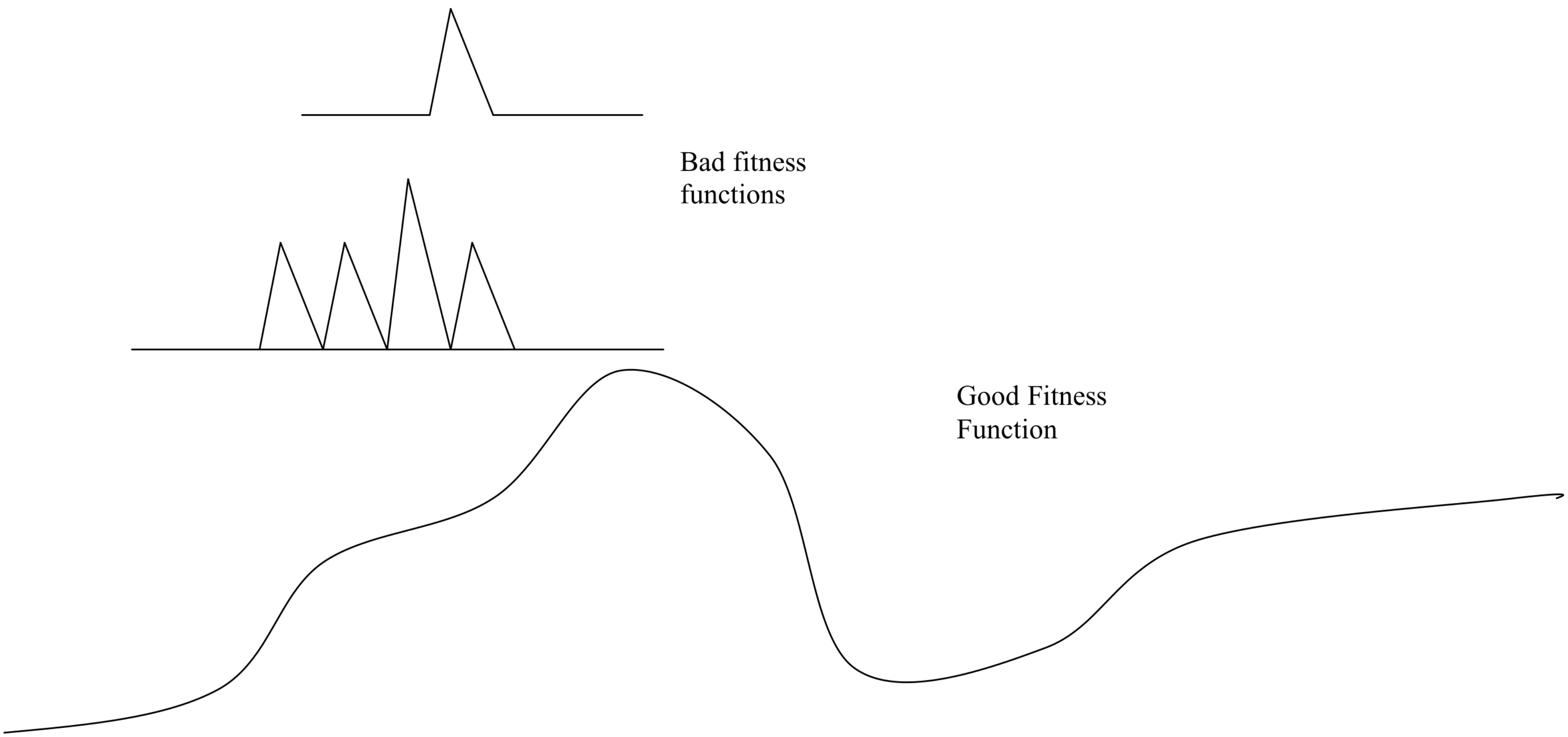
Seeding Algorithm

- Key is two populations:
  - Regular population - retention
  - Seeding population - discovery

What is the Bias?

- Fitness function
  - Searching for individuals with high fitness
  - Children of individuals with high fitness will have high fitness
- Representation
  - Blocks of close genes

Good and Bad Fitness Functions



Blocks of genes

- The closer two Genes are to each other, the more likely they are to be passed together to the same child.
  - Assuming uniform crossover is not used

- This must be taken into account when a representation is designed for GAs
  - Feathers and web feet should be near each other but blue eyes can be further away

Lets watch some movies!!

Genetic Programming

- Search for a highly fit individual computer program in the space of possible computer programs.
- Starts with an initial population of randomly generated computer programs.
- Each individual program is measured in terms of how well it performs in the particular problem environment.

Program Tree

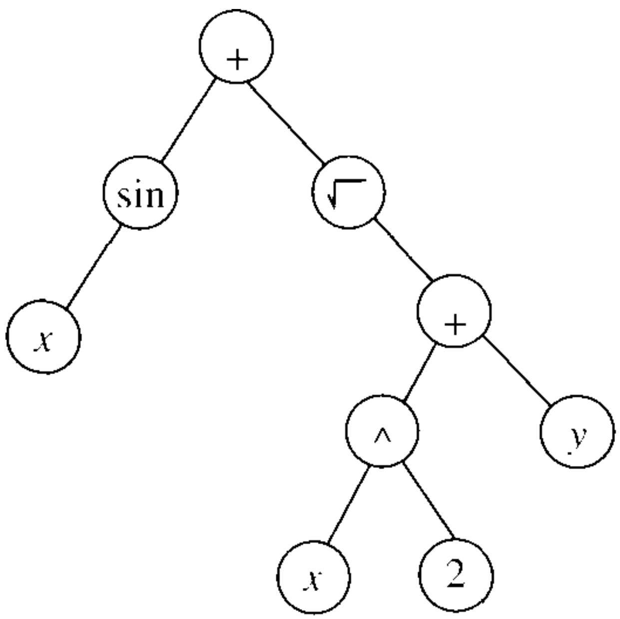


FIGURE 9.1
Program tree representation in genetic programming. Arbitrary programs are represented by their parse trees.

Using Lisp for Genetic Programming

- S-expression can be represented as a rooted tree with ordered branches and labeled nodes.

- Mutation can be achieved by changing the label of a node.

- Crossover is performed by swapping a branch of the tree with a branch from another tree.

Crossover of Program Trees

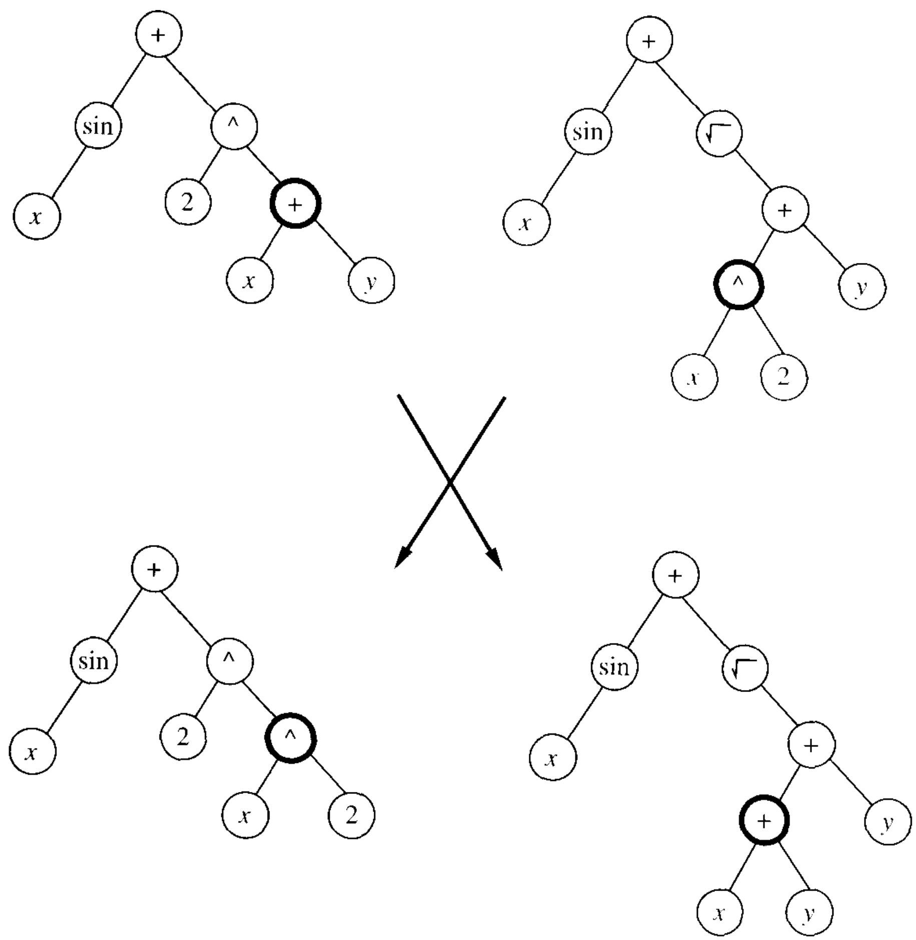


FIGURE 9.2
Crossover operation applied to two parent program trees (top). Crossover points (nodes shown in bold at top) are chosen at random. The subtrees rooted at these crossover points are then exchanged to create children trees (bottom).

Some Implementation Issues

- Make sure that the offsprings are executable (e.g., by using robust functions).
- Alternatively, assign a very low fitness value to offsprings that cause an error.
- To evaluate an offspring with different data, generate a lexical closure for the offspring.
- Alternatively, use global variables.

Summary

- Learning which uses evolution as its search method
- Phenotype vs genotype
- Fitness Function & Representation are important
- Discovery & Retention
- Overcrowding
- Genetic Programming