Genetic algorithms

Adnan Shahzada

Evolutionary Computing

- Natural systems as guiding metaphors
- Charles Darwinian Evolution – 1859
- Theory of natural selection
  - It proposes that the plants and animals that exist today are the result of millions of years of adaptation to the demands of the environment
- Over time, the entire population of the ecosystem is said to evolve
  - to contain organisms that, on average, are fitter than those of previous generations of the population

Evolutionary Computing

- Humans have evolved (and continue to evolve) over time….
- We presume that prehistoric species of man were less intelligent than we are.
- As conditions on earth change, species become extinct (e.g. the dinosaurs) and other species emerge
- Can we model this process and ”evolve” new programs and even hardware?

Evolutionary Computing

- Evolutionary computation(EC) techniques abstract these evolutionary principles into algorithms to search for optimal solutions to a problem.

- In a search algorithm
  - A number of possible solutions to a problem are available
  - Task is to find the best solution possible in a fixed amount of time.

- For a search space with only a small number of possible solutions, all the solutions can be examined in a reasonable amount of time and the optimal one found.

- This *exhaustive search, however, quickly becomes impractical as the search space grows in size.*

Evolutionary Computing

- *Traditional search algorithms randomly or heuristically sample the search space (one solution at a time) in the hopes* of finding the optimal solution.

- *The key aspect distinguishing an evolutionary search algorithm from such traditional algorithms is that it is population-based.*

Genetic Algorithms

- The science that deals with the mechanisms responsible for similarities and differences in a species is called Genetics.

- The concepts of Genetic Algorithms are directly derived from natural evolution.

- In the early 70's a person named John Holland introduced the concept of genetic algorithms (Holland, 1975).

Genetic Algorithms

- *The Cell*
  - Every animal/human cell is a complex of many “small” factories that work together.
  - The center of all this is the cell nucleus.
  - The genetic information is contained in the cell nucleus

Genetic Algorithms

- *Chromosomes*
  - All the genetic information gets stored in the chromosomes.
  - The chromosomes are divided into several parts called genes.
    - Genes code the properties of species i.e., the characteristics of an individual.
    - The possibilities of the genes for one property are called allele and a gene can take different alleles.
    - For example, there is a gene for eye color, and all the different possible alleles are black, brown, blue and green

Genetic Algorithms

- Holland was concerned with manipulating strings of binary digits using natural selection and genetic-inspired techniques.

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

A 16-bit binary string of an artificial chromosome

Genetic Algorithms

- Another biological based method.
- Based on the Darwin's theory. (Evolution of species).
- Based on: survival of the most fittest individual
- Key steps: *reproduction*, *survive*
- Try to simulate life.
- Individual = solution
- Environment = problem

Representation

- Like in real life, the process is not done on the solution (individual), but on its representation (chromosome).
- Therefore we don't use knowledge we have about the problem.
- Very often, based on strings. (but not always), like biological chromosomes.

Breeding Ground

- The idea is that you represent the solution to your problem in a structured way
  - the simplest is a string
  - this is not necessarily the best representation, but it is simple to explain

- We then build a population of random solutions and let them ”breed” using genetic operators
  - at each generation we apply ”survival of the fittest” and hopefully better and better solutions evolve over time
  - the best solutions are more likely to survive and more likely to produce even better solutions

Genetic Operators

- Crossover
  - we cut two solutions at a random point and switch the respective parts
    - Typically a value of 0.7 for crossover probability gives good results.
- Mutation
  - we randomly change a bit in the solution
  - occasional mutation makes the method much less sensitive to the original population and also allows ”new” solutions to emerge
    - Typically, a value between 0.001 and 0.01 for mutation probability is used.

Diagram of Crossover

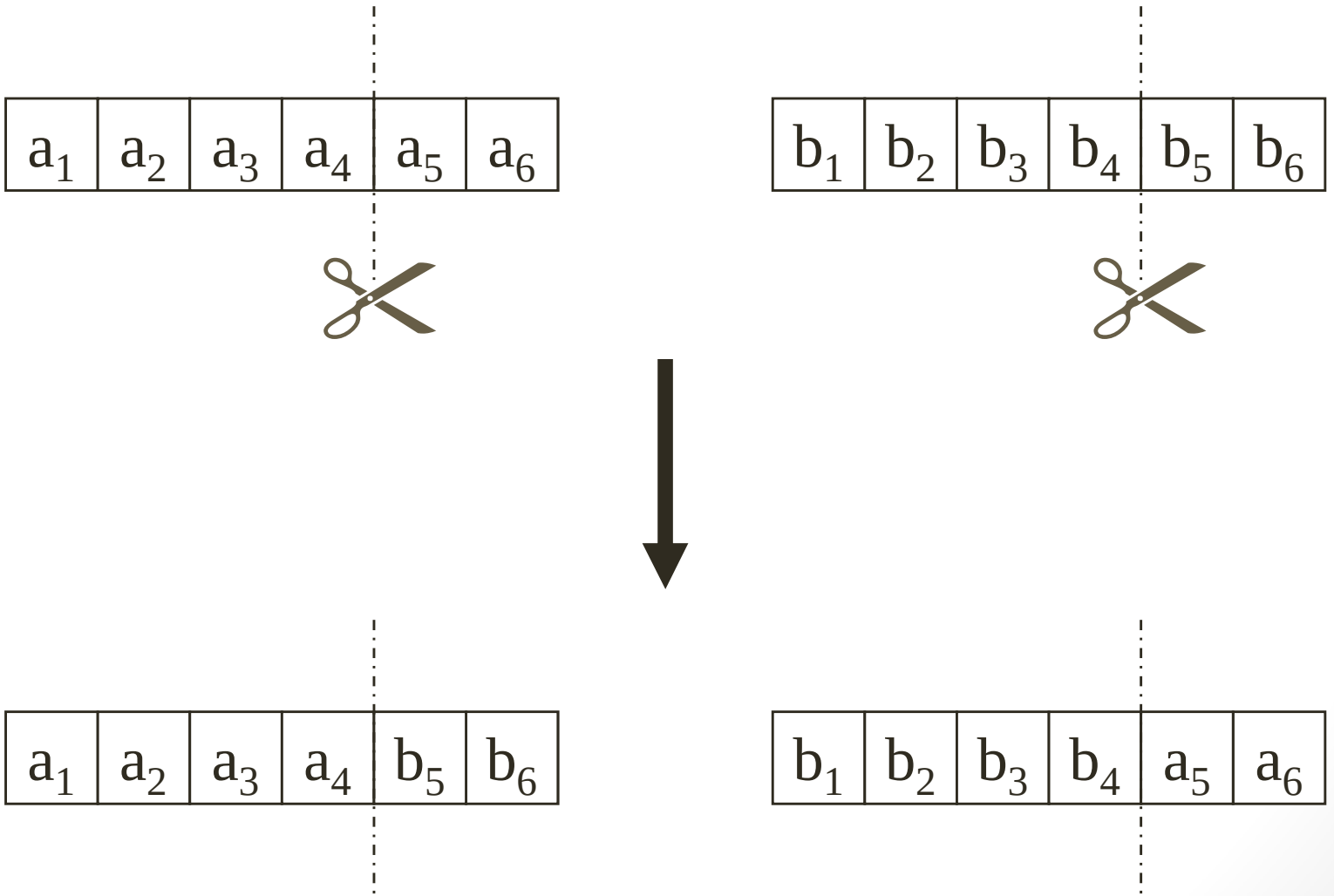
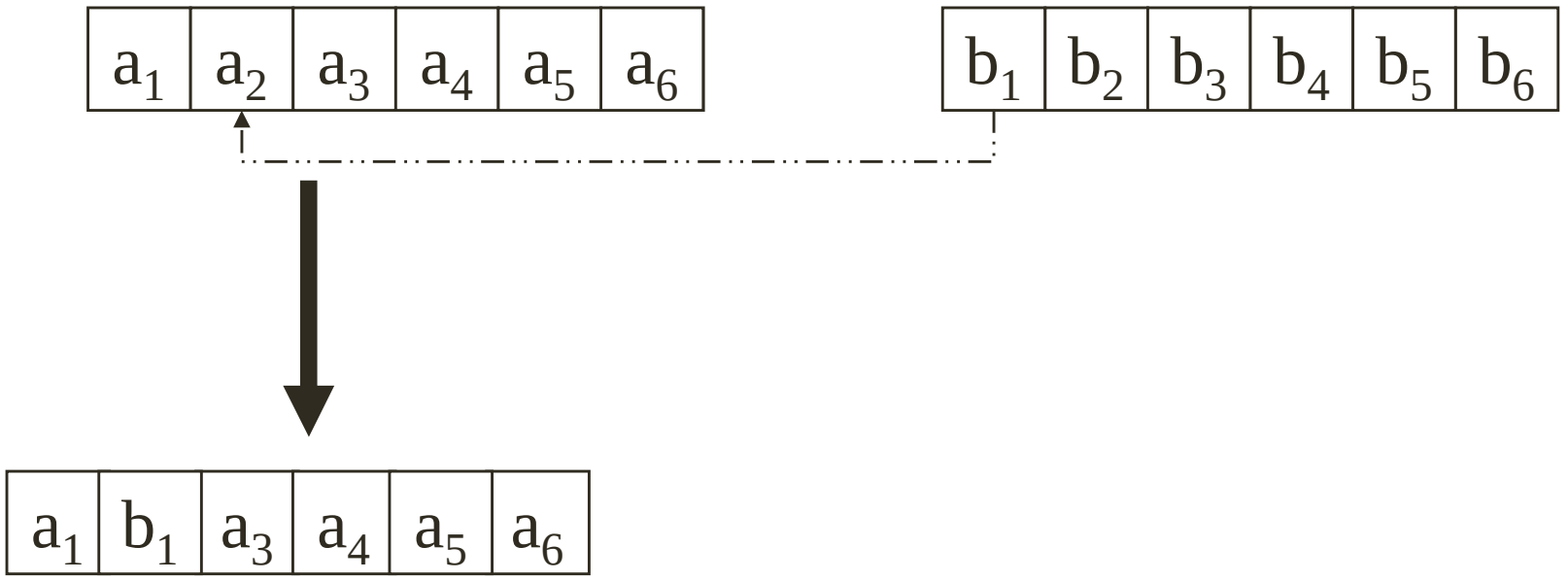


Diagram of Mutation



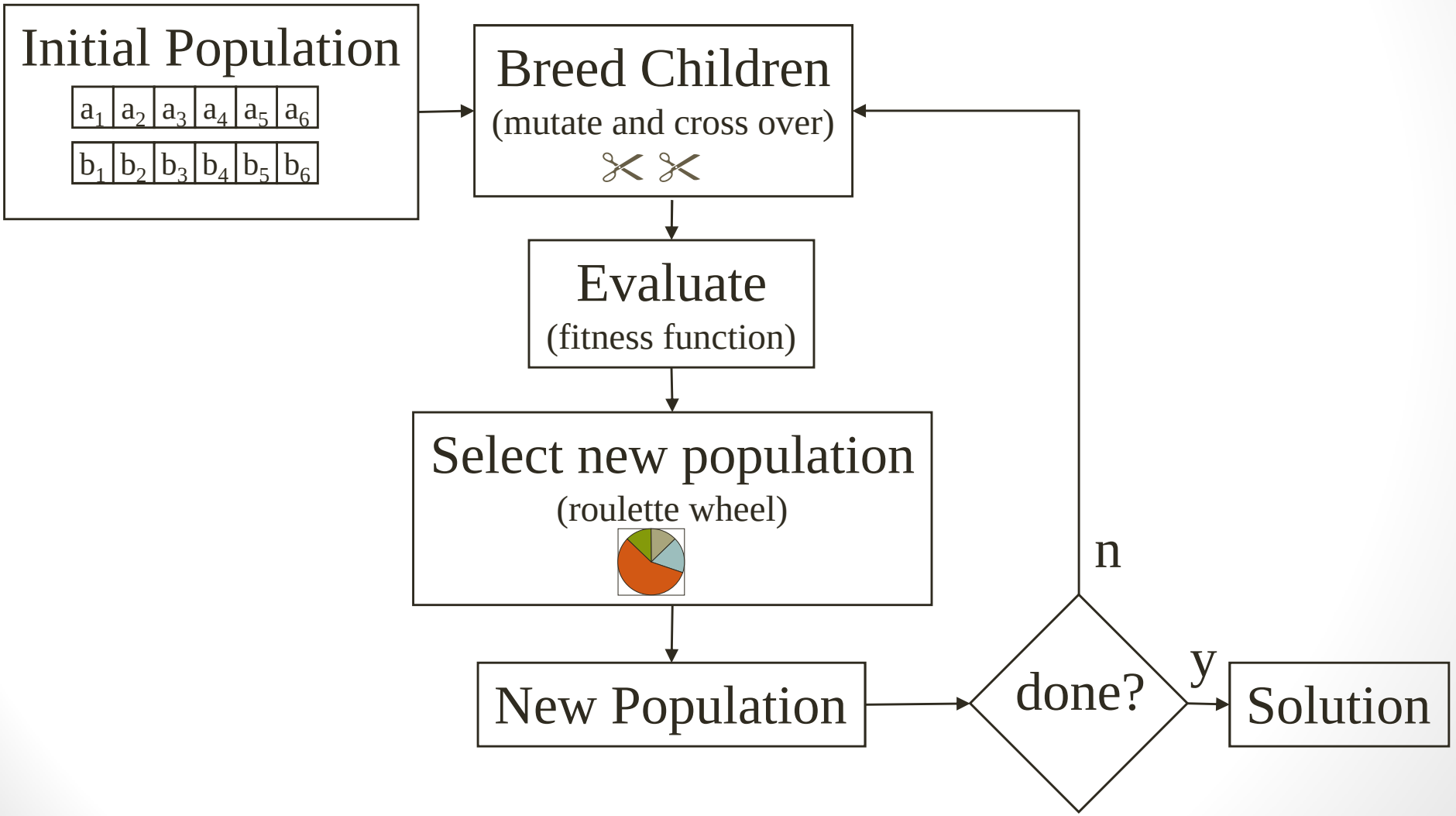
Evaluation and Selection

- We then see how good the solutions are, using an evaluation function
  - often this is a heuristic, especially if it is computationally expensive to do a complete evaluation
  - the final population can then be evaluated more deeply to decide on the best solution

Survival of the Fittest

- We then select the surviving population
- Likelihood of survival is related in some way to your score on the fitness function
  - the most common technique is roulette wheel selection
- Note we always keep the best solution so far (maybe this is the best we can do, so we don't want to lose this solution)

Schematic of GA



Example (Negnevitsky, 2002)

- To find the maximum value of the function $(15x\text{-}x^2)$, where parameter x varies between 0 and 15. Assumption: x takes only integer values.

Integer	Binary code
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Example (Contd.)

- Suppose that the size of the chromosome population N is 6, the crossover probability p_c equals 0.7 and the mutation probability p_m equals 0.001, the fitness function in our example is defined by

  $f(x) = (15x-x^2)$

Example (Contd.)

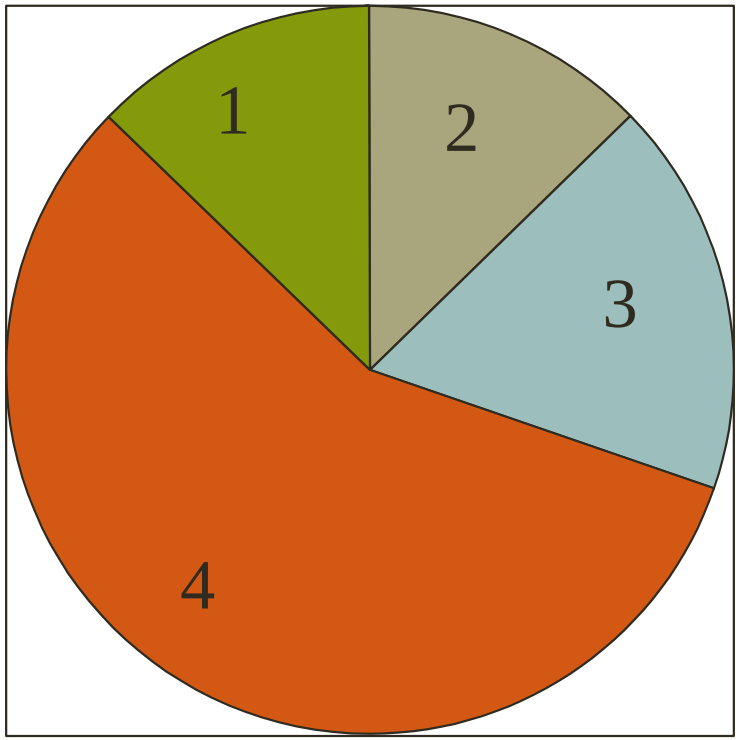
- The GA creates an initial population of chromosomes by filling six 4-bit strings with randomly generated ones and zeros.

Chromosome label	Chromosome string	Decoded integer	Chromosome fitness	Fitness ratio %
X1	1100	12	36	16.5
X2	0100	4	44	20.2
X3	0001	1	14	6.4
X4	1110	14	14	6.4
X5	0111	7	56	25.7
X6	1001	9	54	24.8

Example (Contd.)

- Last column in the previous table shows the ratio of the individual chromosome fitness to the populations total fitness.
- According to the example, chromosomes X5 and X6 have a pretty good chance of being selected for mating, while X3 and X4 have a poor chance.
- Roulette selection wheel is the most commonly used chromosome selection technique.

Roulette selection wheel



- Each member of the population is given a slice of the roulette wheel
- The better the fitness, the bigger the slice
- We then spin the wheel - if your number comes up you survive to the next generation!

Example (Contd.)

- To select a chromosome for mating, a random number (between 0 and 100) is generated.
- The roulette wheel is spun and when the arrow comes to rest on one of the segments, the corresponding chromosome is selected.
- In the current case, the wheel is spun six times.
  - The first two spins might select, X6 and X2;
  - Second two spins might select X1 and X5;
  - The last two spins might select X2 and X5.

Example (Contd.)

- Genetic algorithms assure the continous improvement of the average fitness of the population, and after many generations (typically several hundreds), the population evolves to a near optimal solution.
- In the current case, 0111 and 1000.

Advantages of GA's

- Interesting idea, parallelism
- Very easy to use
- Need no knowledge about the solution
- Could be efficient

Disadvantages of GA's

- Slow (blind search).
- No proof (or hints) about the quality of the solution.
- Sometimes difficult to use them efficiently.
- Tradeoff between mutation and crossover?
- Tradeoff between speed and quality?
- How important is each parameter?

Summary

- It’s a big family, with a lot of possibilities
- Easy to use
- Efficiency ?
- Areas: Graph problems (Graph coloring), optimization (Networks, schedulings etc)