Objectives of this Session

I wish to provide a foundation to:

1. Evolutionary algorithms (EAs)
2. Biological inspiration to EAs
3. Paradigms of evolutionary computing (EC)
4. Chromosome representation in EC
5. Initial population in EC
6. Fitness function in EC
7. Selection operators in EC
8. Reproduction operators in EC
9. Stopping criteria in EC
10. EC optimization versus classical optimization

Intended Outcomes of this Session

At the end of this session, the student will be able to:

1. Relate Evolutionary Algorithms (EAs) to biological evolution
2. Judge if a given problem can be approached using EAs
3. Compare traditional optimization with evolutionary computing-based optimization
4. Formulate a problem into optimization problem solvable by EA
5. List and outline the paradigms of evolutionary computing
6. Arrange the major operators in a generic EA in the form an algorithm
7. Summarize the importance of the initial population, selection operators, reproduction operators and stopping criteria
8. List the types of aforementioned operators and discuss their relative advantages
9. Choose a particular selection operator suitable for a given problem

Recommended Resources for this Session

1. Engelbrecht, A. P. (2007). *Computational intelligence: An introduction*. Chichester, England, John Wiley & Sons.

2. De Jong, K. A. (2012). *Evolutionary Computation: A Unified Approach*. New York, USA, Bradford Books.

3. Konar, A. (2005). *Computational Intelligence: Principles, Techniques and Applications*. Secaucus, NJ, USA, Springer-Verlag New York, Inc.

Areas of Focus in MIS501

- Genetic algorithms (GAs)
- Evolutionary programming (EP)
- Evolution strategies (ESs)
- Genetic programming (GP)
- Differential Evolution (DE)
- Cultural Evolution (CE)
- Coevolution (CoE)

Introduction to Evolution

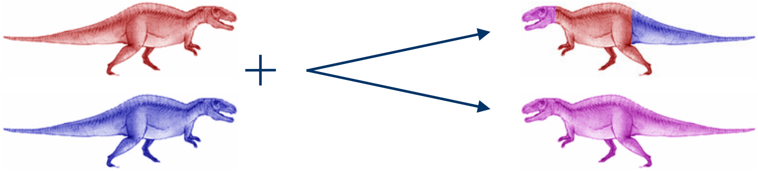
- Evolution is an optimization process that aims to improve the ability of an organism (or a system) to survive in dynamically changing and competitive environments
- Jean-Baptiste Lamarck (1744–1829) was possibly the first to theorize about biological evolution
- Charles Darwin (1809–1882) is widely considered as the founder of both the theory of evolution and the principle of common descent
- Lamarcks theory of evolution was of the inheritance of acquired traits. Individuals adapt during their lifetimes, and transmit their traits to their offspring which continue to adapt
- Lamarckism rests on the concept of use and disuse: over time, individuals lose characteristics they do not require, and develop those which are useful by "exercising" them

Introduction to Evolution

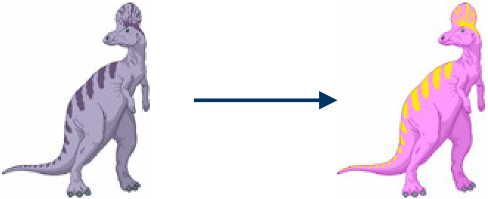
- Darwins theory of natural selection became the foundation of biological evolution
- In a world with limited resources and stable populations, each individual competes with others for survival. Individuals having the "best" characteristics are more likely to survive and to reproduce, and those characteristics will be passed on to their offspring
- The desirable characteristics are inherited by the following generations, and over time, become dominant among the population
- A second part of Darwins theory states that, during production of a child organism, random events cause random changes to the child organisms characteristics
- If the new characteristics are a benefit, then the chances of survival for that organism are increased

Biological Evolution

- Organisms produce a number of offspring similar to themselves but can have variations due to:
  - Sexual reproduction

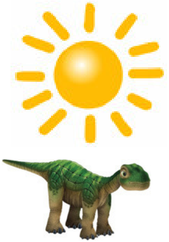


  - Mutations (random changes)



Biological Evolution

- Some offspring survive, and produce next generations, and some do not
- The organisms adapted to the environment better have higher chance to survive
- Over time, the generations become more and more adapted because the fittest organisms survive



Generic Evolutionary Algorithm

- Evolutionary computation refers to computer-based problem solving systems that use computational models of evolutionary processes, such as natural selection, survival of the fittest and reproduction, as the fundamental components of such computational systems

- Evolution via natural selection of a randomly chosen population of individuals can be thought of as a search through the space of possible chromosome values

- An EA is a stochastic search for an optimal solution to a given problem

Generic Evolutionary Algorithm

The evolutionary search process is influenced by the following main components of an EA:

- An encoding of solutions to the problem as a chromosome
- A function to evaluate the fitness, or survival strength of individuals
- Initialization of the initial population
- Selection operators and
- Reproduction operators

Generic Evolutionary Algorithm

Algorithm 2A.1. Generic Evolutionary Algorithm

```
Let t = 0 be the generation counter;
Create and initialize an n_x-dimensional population, C(0), to consist of n_s individuals;
while (stopping condition(s) not true) do
    Evaluate the fitness, f(x_i(t)), of each individual, x_i(t);
    Perform reproduction to create offspring;
    Select the new population, C(t + 1);
    Advance to the new generation t = t + 1;
end while
```

EC Paradigms

- GAs model genetic evolution
- GP is based on genetic algorithms, but individuals are programs represented as trees
- EP is derived from the simulation of adaptive behavior in evolution (*phenotypic* evolution)
- ESs are geared toward modeling the strategic parameters that control variation in evolution(the evolution of evolution)
- DE is similar to genetic algorithms, differing in the reproduction mechanism used
- CE models the evolution of culture of a population and how the culture influences the genetic and phenotypic evolution
- CoE has initially "dumb" individuals that evolve through cooperation, or in competition with one another, acquiring the necessary characteristics to survive

Representation: The Chromosome

- Characteristics that influence an organism's ability to survive and to reproduce are represented by information contained in **chromosomes** found in the cell nuclei
- Each chromosome contains **genes** (unit of heredity)
- Genes determine many aspects of anatomy and physiology
- Each individual has a unique sequence of genes. An alternative form of a gene is referred to as an **allele**
- In EC, each individual represents a candidate solution to an optimization problem. The characteristics of an individual is represented by a chromosome (genome)
- Each variable that needs to be optimized is referred to as a gene
- A genotype describes the genetic composition of an individual. A phenotype describes the behavioral traits of an individual

Representation: An Example

- **Gene:** A single encoding of part of the solution space, (an element of the candidate solution)

1	0	1	0

- **Chromosome:** A string of genes that represents a solution

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

- **Population:** The number of chromosomes available to test

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

Representation: The Chromosome

- An important step in the design of an EA is to find an appropriate representation of candidate solutions (chromosomes)
- Different EAs use different representations. Most EAs represent solutions as vectors of a specific data type
- In GP, individuals are pieces of a program code, represented in a tree format
- The classical representation scheme for GAs is binary vectors of fixed length
- In case of an n_x-dimensional search space, each individual consists of n_x variables, each encoded as a bit string
- In nominal-valued variables, each nominal value can be encoded as an n_d-dimensional bit vector where 2^{n_d} is the total number of discrete nominal values for that variable

Representation: The Chromosome

- In continuous-valued optimization, the continuous search problem can be mapped into a discrete search problem
- Mapping functions are needed to convert the space $\{0,1\}^{n_b}$ to the space $\mathbb{R}^{n_x}$
- For such mapping, each continuous-valued variable is mapped to an n_d-dimensional bit vector, $\phi : \mathbb{R} \to (0,1)^{n_d}$
- The domain of the continuous space needs to be restricted to a finite range, $[\mathbf{x}_{\min}, \mathbf{x}_{\max}]$
- A standard binary encoding scheme can be used to transform the individual $\mathbf{x} = (x_1, \ldots, x_j, \ldots, x_{nx})$, with $x_j \in \mathbb{R}$ to the binary-valued individual, $\mathbf{b} = (\mathbf{b}_1, \ldots, \mathbf{b}_j, \ldots, \mathbf{b}_{nx})$, where $\mathbf{b}_j = (b_{(j-1)}n_d + 1, \cdots, b_{jn_d})$, with $b_l \in \{0,1\}$ and the total number of bits, $n_b = n_x n_d$

Representation: The Chromosome

- Each $\mathbf{b}_j$ can be decoded to a real number using the function $\Phi_j : \{0,1\}^{n_d} \rightarrow [x_{\min,j}, x_{\max,j}]$, where

$$\Phi_j(\mathbf{b}) = x_{\min,j} + \frac{x_{\max,j} - x_{\min,j}}{2^{n_d} - 1} \left(\sum_{l=1}^{n_d - 1} b_{j(n_d - 1)} 2^l \right)$$

- For a conversion form a floating-point value to a bit-string of n_d bits, the maximum attainable accuracy is

$$\frac{x_{\max,j} - x_{min,j}}{2^{n_d} - 1} \quad \text{for } j = 1, \ldots, n_x$$

- Gray coding is an alternative bit representation in which the Hamming distance between the representation of successive numerical values is one

Representation: The Chromosome

- Binary to Gray code conversion rules are:

$$
\begin{aligned}
g_1 &= b_1 \\
g_l &= b_{l-1}\bar{b}_l + \bar{b}_{l-1} b_l
\end{aligned}
$$

- Each $\mathbf{b}_j$ can be decoded to a real number using

$$
\Phi_j(\mathbf{b}) = x_{\min,j} + \frac{x_{\max,j} - x_{\min,j}}{2^{n_d} - 1} \left(\sum_{l=1}^{n_d-1} \left(\sum_{q=1}^{n_d-l} b_{(j-1)n_d+q} \right) \mod 2 \right) 2^l
$$

Initial Population

- EAs are stochastic, population-based search algorithms. The first step is to generate an initial population
- Assign a random value to each gene of each chromosome which ensures that the initial population is a uniform representation of the entire search space
- The size of the initial population has impacts computational complexity and exploration abilities
- Large numbers of individuals increase diversity and improve the exploration abilities leading to higher computational complexity. But, it may take fewer generations to locate an acceptable solution
- A small population results in lower time complexity per generation. But, the it take more generations to converge

Fitness Function

- A mathematical function is used to quantify how good the solution represented by a chromosome is
- The fitness function f maps a chromosome representation into a scalar value, $f : \Gamma^{n_x} \rightarrow \mathbb{R}$, where Γ represents the data type of the elements of an n_x-dimensional chromosome
- The fitness function represents the objective function, Φ, which describes the optimization problem
- The chromosome representation may not correspond to the representation expected by the objective function. In such cases the fitness function representation is $f : \mathcal{S}_C \xrightarrow{\Phi} \mathcal{S}_X \xrightarrow{\Psi} \mathbb{R} \xrightarrow{\Upsilon} \mathbb{R}_+$
- $\mathcal{S}_C$: Search space Φ: Chromosome decoding function
  Ψ: Objective function Υ: Scaling function

f Depends on the Type of Optimization

- **Unconstrained optimization:** $\mathcal{S}_C = \mathcal{S}_X$. The fitness function is same as the objective function
- **Constrained optimization:** The fitness function contains two parts, the original objective function, and the a constraint penalty function
- **Multi-objective optimization:** The fitness function is a weighted sum of all the sub-objectives
- **Dynamic and noisy problems:** Dynamic fitness functions are time-dependent. Noisy functions have additive noise component

Selection

- Selection relates to the concept of survival of the fittest. It ensures better solutions
- A new population can be either from only the offspring, or from both the parents and the offspring
- The selection operator should ensure that good individuals survive to next generations
- Offspring are created using crossover and/or mutation operators. In crossover, "superior" individuals should get more opportunities to reproduce
- Mutation should focus on "weak" individuals because it brings better traits in them and increases their chances of survival

Selective Pressure

- Selective pressure is the speed at which the best solution will occupy the entire population by repeated application of the selection operator alone
- An operator with a high selective pressure decreases diversity in the population, which may lead to premature convergence to suboptimal solutions
- A high selective pressure limits the exploration abilities of the population

Selection Operators: Random Selection

- Each individual has the same probability of $\frac{1}{n_s}$
- The best and the worst individuals have exactly the same probability of surviving to the next generation
- Random selection has the lowest selective pressure

Proportional Selection

- The probability of selection of an individual depends on its fitness
- A probability distribution proportional to the fitness is created, and individuals are selected by sampling the distribution

$$\varphi(\mathbf{x}_i(t)) = \frac{f_\Upsilon(\mathbf{x}_i(t))}{\sum_{l=1}^{n_s} f_\Upsilon(\mathbf{x}_i(t))}$$

- Here, n_s is the population size and $\varphi(\mathbf{x}_i(t))$ is the probability that $\mathbf{x}_i$ will be selected
- **Roulette wheel selection** is a good example of proportional selection

Roulette Wheel Selection

Algorithm 2A.2. Roulette Wheel Selection for maximization

1: Let the chromosome index $i = 1$;
2: Calculate $\varphi(\mathbf{x}_i)$;
3: $sum = \varphi(\mathbf{x}_i(t))$;
4: Choose $r \sim U(0, 1)$
5: **while** $sum < r$ **do**
6: $\quad i = i + 1$ (Go to the next chromosome);
7: $\quad sum = sum + \varphi(\mathbf{x}_i)$;
8: **end while**
9: Return $\mathbf{x}_i$ as the selected individual;

Drawbacks: High variance in the number of offspring created by each individual; The best individual may not get to reproduce

Stochastic Universal Sampling

Algorithm 2A.3. Stochastic Universal Sampling

for $i = 1, \ldots, n_s$ **do**
    $\lambda_i(t) = 0$;
end for
$r \sim U(0, \frac{1}{\lambda})$; ($\lambda$ is the number of offspring)
$sum = 0$;
for $i = 1, \ldots, n_s$ **do**
    $sum = sum + \varphi(\mathbf{x}_i(t))$;
    while r¡sum **do**
        $\lambda_i + +$;
        $r = r + \frac{1}{\lambda}$;
    end while
end for
Return $\lambda = (\lambda_1, \ldots, \lambda_{n_s})$;

Tournament Selection

- A group of n_{ts} individuals is selected randomly from the population, where $n_{ts} < n_s$
- The best individual from this group is selected
- If n_{ts} is not too large, tournament selection prevents the best individual from dominating (low selection pressure)
- If n_{ts} is too small, the chances that bad individuals are selected increase
- Random selection of the individuals for tournament reduces selective pressure compared to proportional selection
- If $n_{ts} = n_s$, the best individual is always selected (very high selective pressure). Random selection is obtained if $n_{ts} = 1$

Rank-Based Selection

- Uses the rank ordering of fitness values to determine the probability of selection, and not the fitness values
- Rank based selection is independent of actual fitness values because the best individual does not dominate in the selection
- Linear sampling selects an individual $_i$ such that $i \sim U(0, U(0, n_s - 1))$, where the individuals are sorted in decreasing order of fitness value. The rank of the best individual is 0, and that of the worst individual is $n_s - 1$
- Linear ranking assumes that the best individual creates $\hat{\lambda}$ offspring, and the worst individual $\tilde{\lambda}$, where $1 \leq \hat{\lambda} \leq 2$ and $\tilde{\lambda} = 2 - \hat{\lambda}$
- Rank-based selection operators may use any sampling method, such as roulette wheel selection or stochastic universal sampling to select individuals

Boltzmann Selection

- Based on the thermodynamical principles of simulated annealing
- Selection probabilities are computed as:

$$\varphi(\mathbf{x}_i(t)) = \frac{1}{1 + e^{f(\mathbf{x}_i(t))/T(t)}}$$

- Here $T(t)$ is the temperature parameter. A temperature schedule is used to reduce $T(t)$ from its initial large value to a small value
- The initial large value ensures that all individuals have an equal probability of being selected. As $T(t)$ becomes smaller, selection focuses more on the good individuals
- Boltzmann selection can also be used to select between two individuals

$(\mu \overset{+}{,} \lambda)$ Selection

- The (μ, λ) and $(\mu + \lambda)$ are deterministic rank-based selection methods used in evolutionary strategies.
- Here, μ indicates the number of parents (size of the population), and λ is the number of offspring produced from each parent
- After production of the λ offspring, (μ, λ)-selection selects the best μ offspring for the next population
- $(\mu + \lambda)$-selection selects the best individuals from both the parents and the offspring

Elitism

- Elitism is the Process of ensuring that the best individuals survive to the next generation
- The best individuals are copied to the new population without mutation
- The more individuals that survive to the next generation, the less the diversity of the new population

Hall of Fame

- The hall of fame is a selection scheme similar to the list of best players of an arcade game
- For each generation, the best individual is inserted into the hall of fame
- The hall of fame contains an archive of the best individuals found from the first generation
- The hall of fame can be used as a parent pool for the crossover operator
- At the last generation, the best individual is selected as the best one in the hall of fame

Reproduction Operators

- **Reproduction:** The process of producing offspring from selected parents by applying crossover and/or mutation operators
- Crossover: The process of creating one or more new individuals through the combination of genetic material randomly selected from two or more parents
- If selection focuses on the most-fit individuals, the selection pressure may cause premature convergence due to reduced diversity of the new population
- **Mutation:** The process of randomly changing the values of genes in a chromosome in order to introduce new genetic material into the population, thereby increasing genetic diversity

Reproduction Operators

- Mutation should not distort the good genetic material in highly fit individuals. For this reason, mutation is usually applied at a low probability
- Alternatively, the mutation probability can be made proportional to the fitness of individuals: the less fit the individual, the more it is mutated
- To promote exploration in the first generations, the mutation probability can be initialized to a large value, which is then reduced over time to allow for exploitation during the final generations
- Reproduction can be applied with replacement. Newly generated individuals replace parents only if the fitness of the former is better than that of the latter

Stopping Conditions

- The evolutionary operators are iteratively applied in an EA until a stopping condition is satisfied. Generally, they limit the number of generations or the number of fitness evaluations
- Another convergence criterion is to detect if the population has converged (stagnant)
- EA can terminate if
  - No improvement is observed
  - There is no change in the population
  - An acceptable solution has been found
  - Objective function slope is approximately zero

  over a number of consecutive generations

EC Versus Classical Optimization

- No-Free-Lunch theorem states that no single algorithm exists for solving all problems that is on average superior to any other algorithm
- Classical optimization: Efficient in linear, quadratic, strongly convex, unimodal and other specialized problems EAs: Efficient for discontinuous non-differentiable, multimodal and noisy problems
- The difference is in their search process and how they use the search surface information

EC Versus Classical Optimization

- **The search process**
  - Classical optimization uses deterministic rules to move from one point in the search space to the next, while EC uses probabilistic transition rules
  - EC applies a parallel search, while classical optimization uses a sequential search
  - EA starts from a set of initial points, while classical optimization starts from one point, successively adjusting this point to move towards the optimum
- **Search surface information**
  - Classical optimization uses first-order or second-order derivative information of the search space to guide the path to the optimum, while EC uses no derivative information (the fitness values of individuals are used to guide the search)

Session Summary

1. EAs are inspired by the natural genetic evolution
2. EAs capture the concepts of *inheritance*, *struggle for existence* and *survival of the fittest*
3. EAs include ECs, GAs, GPs, DEs and CEs and CoE
4. Chromosome are the solution candidates in ECs
5. Initial population, Selection operators, Reproduction operators and Stopping criteria play important roles in the quality of solutions
6. Popular selection operators: Random, Proportional, Roulette Wheel, Tournament, Rank-Based, Boltzmann, Elitism, Hall of Fame
7. Reproduction operators: Crossover and Mutation

Any Questions?

Thank You