RAMAIAH
Institute of Technology

M.S. Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)
Department of Computer Science and Engineering

Course Name: Artificial Intelligence(AI)

Course Code: CS53

Credits: 3:0:1

UNIT - 5

Term: October 2021 – Feb 2022

RAMAIAH
Institute of Technology

Outline

Genetic Algorithms

1. Genetic Algorithms Introduction
2. Significance of Genetic Operators
3. Termination Parameters
4. Niching and Speciation
5. Evolving Neural Networks
6. Theoretical Grounding
7. Ant Algorithms

RAMAIAH
Institute of Technology

Genetic Algorithm

- Genetic Algorithm (GA) is a search-based optimization technique based on the principles of **Genetics and Natural Selection**.
- It is frequently used to find optimal or near-optimal solutions to difficult problems which otherwise would take a lifetime to solve.
- It is frequently used to solve optimization problems, in research, and in machine learning.

RAMAIAH
Institute of Technology

Genetic Algorithm

- Optimization is the process of **making something better**. In any process, we have a set of inputs and a set of outputs as shown in the following figure.
- Optimization refers to finding the values of inputs in such a way that we get the “best” output values.
- The definition of “best” varies from problem to problem, but in mathematical terms, it refers to maximizing or minimizing one or more objective functions, by varying the input parameters.

Genetic Algorithm

What are Genetic Algorithms?

Nature has always been a great source of inspiration to all mankind. Genetic Algorithms (GAs) are search based algorithms based on the concepts of natural selection and genetics.

GAs are a subset of a much larger branch of computation known as **Evolutionary Computation**.

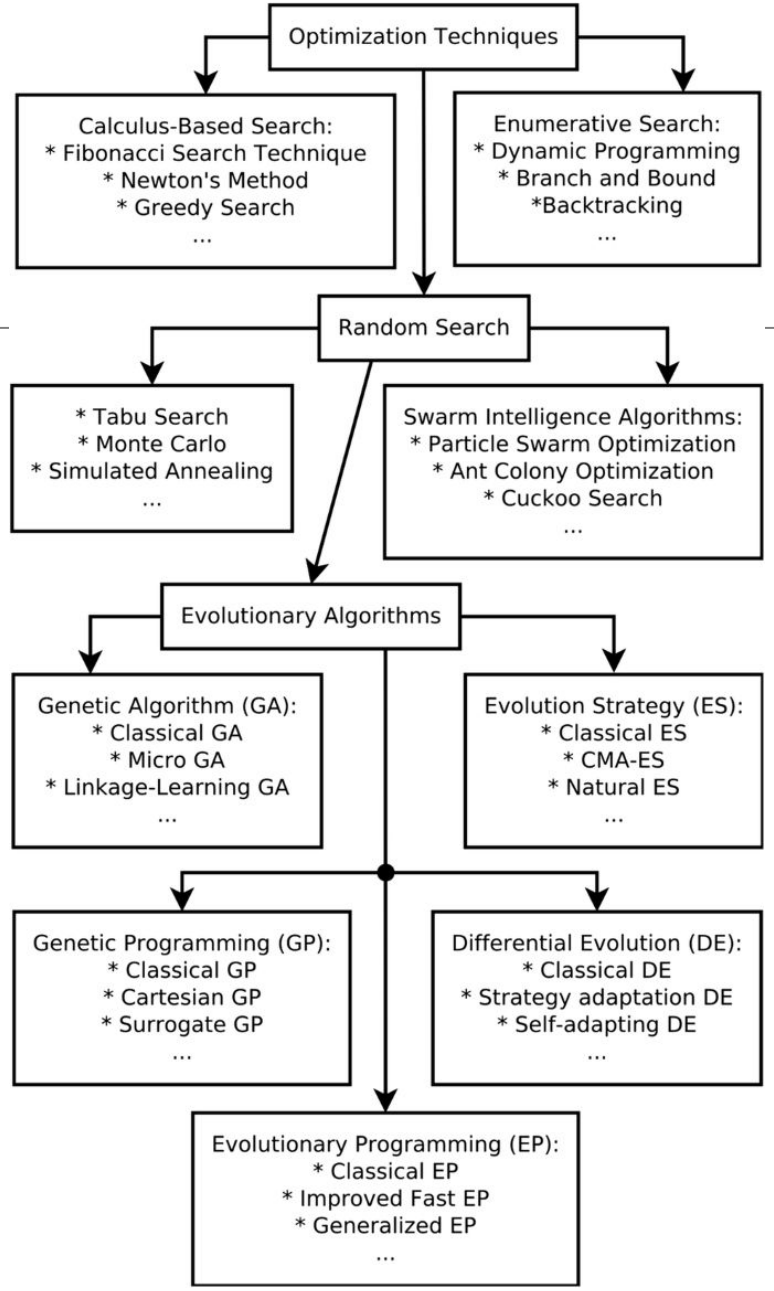
GAs were developed by John Holland and his students and colleagues at the University of Michigan, most notably David E. Goldberg and has since been tried on various optimization problems with a high degree of success.

Genetic Algorithm

- In GAs, we have a **pool or a population of possible solutions** to the given problem.
- These solutions then undergo recombination and mutation (like in natural genetics), producing new children, and the process is repeated over various generations.
- Each individual (or candidate solution) is assigned a fitness value (based on its objective function value) and the fitter individuals are given a higher chance to mate and yield more “fitter” individuals. This is in line with the Darwinian Theory of “Survival of the Fittest”.
- In this way we keep “evolving” better individuals or solutions over generations, till we reach a stopping criterion.
- Genetic Algorithms are sufficiently randomized in nature, but they perform much better than random local search (in which we just try various random solutions, keeping track of the best so far), as they exploit historical information as well.

RAMAIAH
Institute of Technology

Applications



GA Motivation

Genetic Algorithms have the ability to deliver a “good-enough” solution “fast-enough”. This makes genetic algorithms attractive for use in solving optimization problems. The reasons why GAs are needed are as follows –

Solving Difficult Problems

1. In computer science, there is a large set of problems, which are **NP-Hard**. What this essentially means is that, even the most powerful computing systems take a very long time (even years!) to solve that problem.
2. In such a scenario, GAs prove to be an efficient tool to provide **usable near-optimal solutions** in a short amount of time.

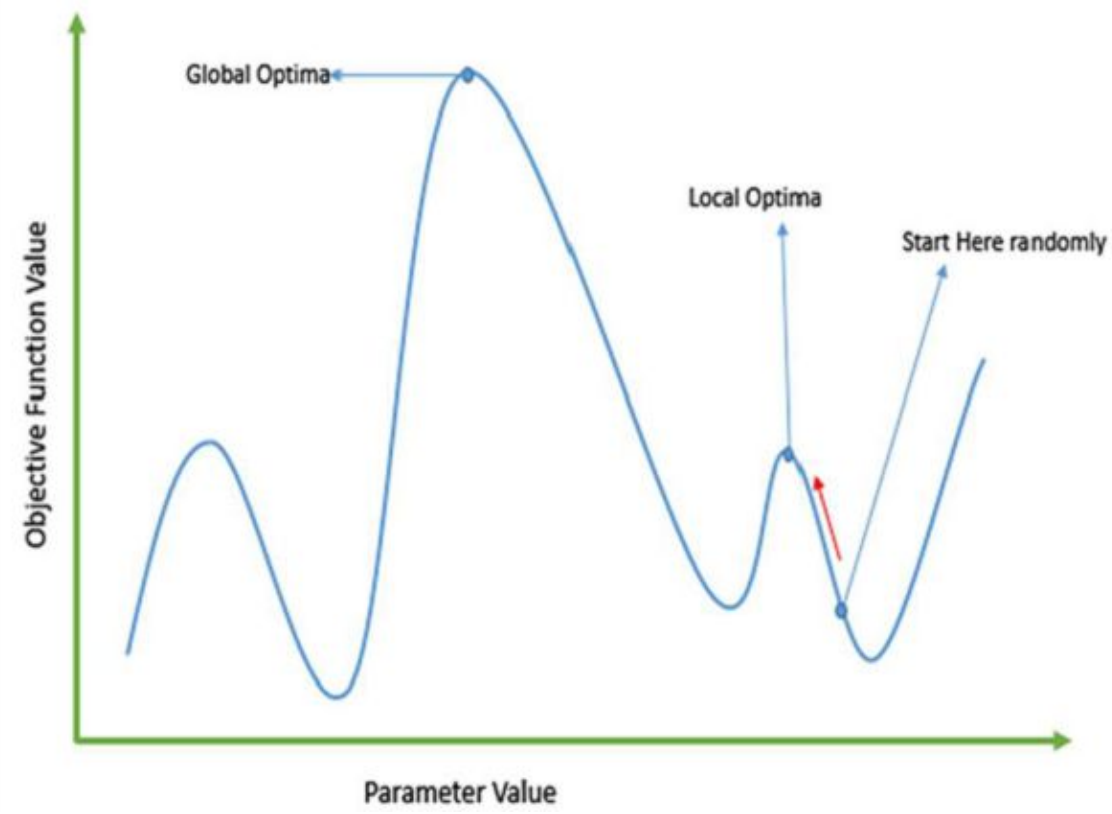
GA Motivation

Failure of Gradient Based Methods

1. Traditional calculus based methods work by starting at a random point and by moving in the direction of the gradient, till we reach the top of the hill. This technique is efficient and works very well for single-peaked objective functions like the cost function in linear regression.
2. But, in most real-world situations, we have a very complex problem called as landscapes, which are made of many peaks and many valleys, which causes such methods to fail, as they suffer from an inherent tendency of getting stuck at the local optima as shown in the following figure.

GA Motivation

Failure of Gradient Based Methods



GA Motivation

Getting a Good Solution Fast

1. Some difficult problems like the Travelling Salesperson Problem (TSP), have real-world applications like path finding and VLSI Design.
2. Now imagine that you are using your GPS Navigation system, and it takes a few minutes (or even a few hours) to compute the “optimal” path from the source to destination.
3. Delay in such real world applications is not acceptable and therefore a “good-enough” solution, which is delivered “fast” is what is required.

RAMAIAH
Institute of Technology

Basic terminology of GA

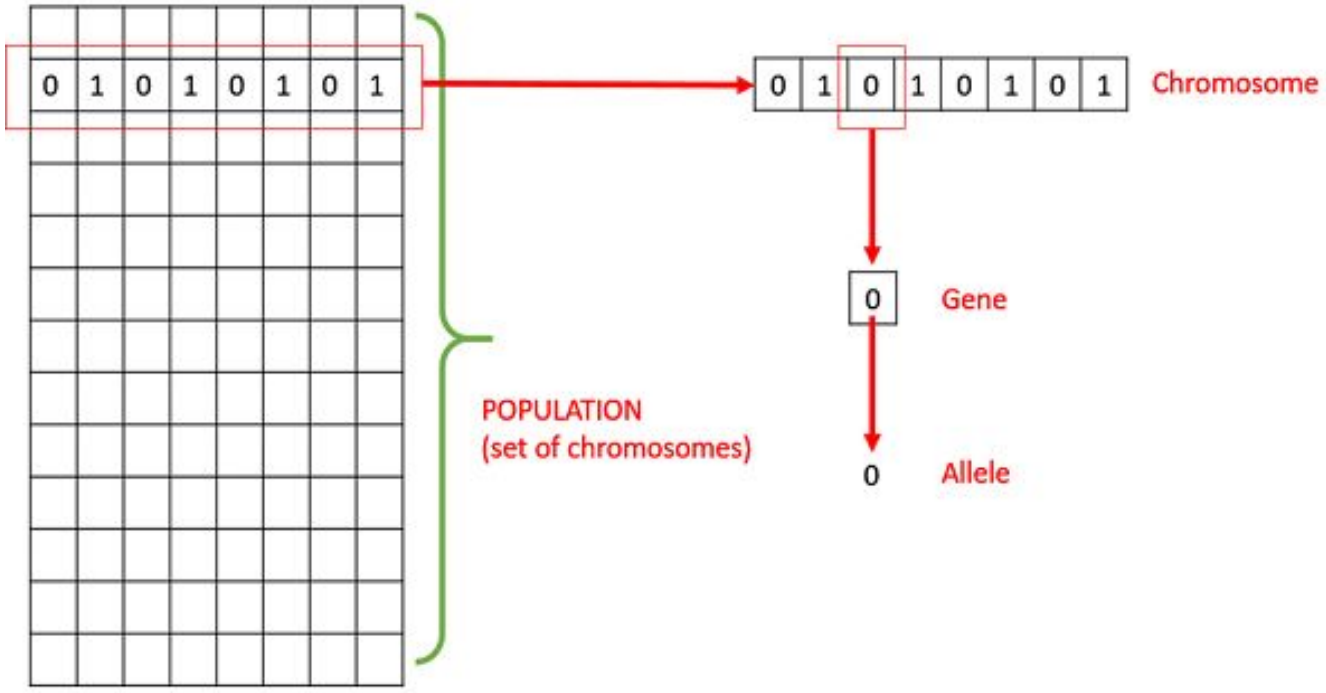
Before beginning a discussion on Genetic Algorithms, it is essential to be familiar with some basic terminology which will be used.

- **Population** – It is a subset of all the possible (encoded) solutions to the given problem. The population for a GA is analogous to the population for human beings except that instead of human beings, we have Candidate Solutions representing human beings.
- **Chromosomes** – A chromosome is one such solution to the given problem.
- **Gene** – A gene is one element position of a chromosome.
- **Allele** – It is the value a gene takes for a particular chromosome.

RAMAIAH
Institute of Technology

Basic terminology of GA

Before beginning a discussion on Genetic Algorithms, it is essential to be familiar with some basic terminology which will be used



RAMAIAH
Institute of Technology

Basic terminology of GA

Genotype – Genotype is the population in the computation space. In the computation space, the solutions are represented in a way which can be easily understood and manipulated using a computing system.

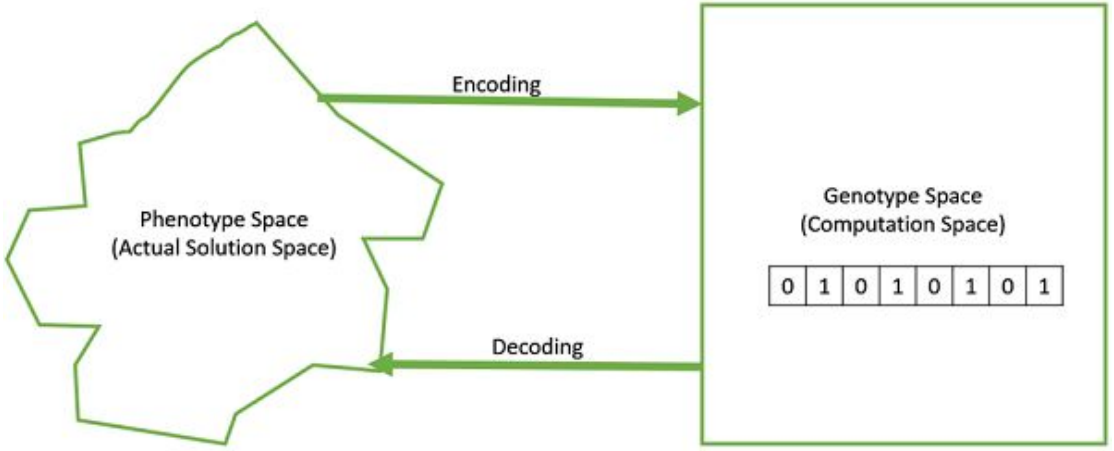
Phenotype – Phenotype is the population in the actual real world solution space in which solutions are represented in a way they are represented in real world situations.

Decoding and Encoding – For simple problems, the **phenotype and genotype** spaces are the same. However, in most of the cases, the phenotype and genotype spaces are different. Decoding is a process of transforming a solution from the genotype to the phenotype space, while encoding is a process of transforming from the phenotype to genotype space. Decoding should be fast as it is carried out repeatedly in a GA during the fitness value calculation.

Basic terminology of GA

For example, consider the 0/1 Knapsack Problem. The Phenotype space consists of solutions which just contain the item numbers of the items to be picked.

However, in the genotype space it can be represented as a binary string of length n (where n is the number of items). A **0 at position x** represents that **x^{th}** item is picked while a 1 represents the reverse. This is a case where genotype and phenotype spaces are different.



Basic terminology of GA

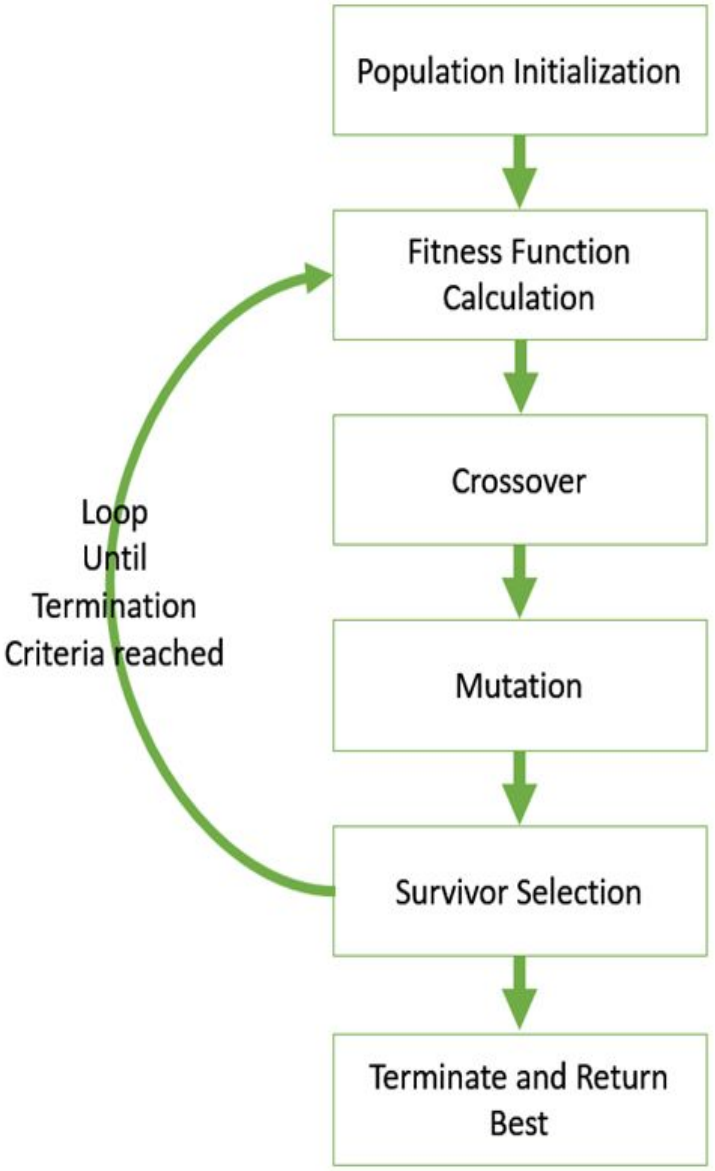
Fitness Function – A fitness function simply defined is a function which takes the solution as input and produces the suitability of the solution as the output. In some cases, the fitness function and the objective function may be the same, while in others it might be different based on the problem.

Genetic Operators – These alter the genetic composition of the offspring. These include crossover, mutation, selection, etc.

RAMAIAH
Institute of Technology

Basic Structure of GA

- We start with an initial population (which may be generated at random or seeded by other heuristics),
- select parents from this population for mating.
- Apply crossover and mutation operators on the parents to generate new off-springs.
- And finally these off-springs replace the existing individuals in the population and the process repeats.
- In this way genetic algorithms actually try to mimic the human evolution to some extent.

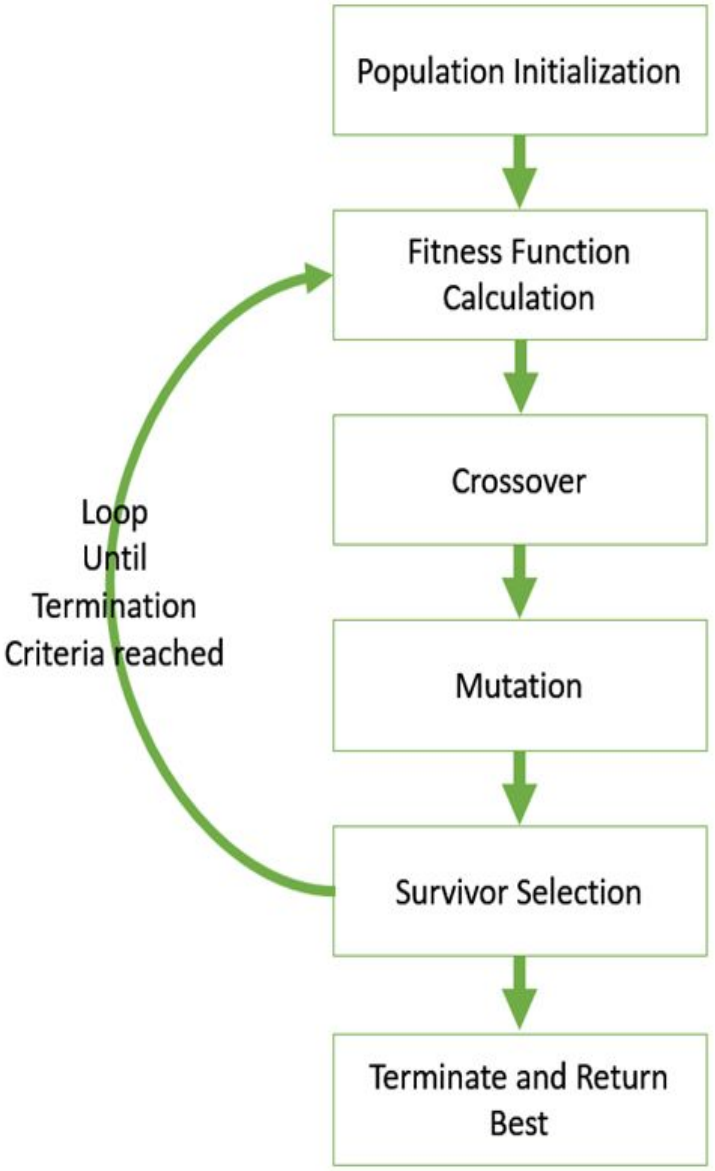


RAMAIAH
Institute of Technology

Basic Structure of GA

A generalized pseudo-code for a GA :

```
GA()
    initialize population
    find fitness of population
 while (termination criteria is reached) do
    parent selection
    crossover with probability pc
    mutation with probability pm
    decode and fitness calculation
    survivor selection
    find best
 return best
```



Genotype representation

- One of the most important decisions to make while implementing a genetic algorithm is deciding the representation that we will use to represent our solutions. It has been observed that improper representation can lead to poor performance of the GA.
- Therefore, choosing a proper representation, having a proper definition of the mappings between the phenotype and genotype spaces is essential for the success of a GA.
- In this section, we present some of the most commonly used representations for genetic algorithms. However, representation is highly problem specific and the reader might find that another representation or a mix of the representations mentioned here might suit his/her problem better.

RAMAIAH
Institute of Technology

Genotype representation

Binary Representation

This is one of the simplest and most widely used representation in GAs. In this type of representation the genotype consists of bit strings.

For some problems when the solution space consists of Boolean decision variables – yes or no, the binary representation is natural. Take for example the 0/1 Knapsack Problem. If there are n items, we can represent a solution by a binary string of n elements, where the x^{th} element tells whether the item x is picked (1) or not (0).

RAMAIAH
Institute of Technology

Genotype representation

Real Valued Representation

For problems where we want to define the genes using continuous rather than discrete variables, the real valued representation is the most natural. The precision of these real valued or floating point numbers is however limited to the computer.

0.5	0.2	0.6	0.8	0.7	0.4	0.3	0.2	0.1	0.9

Genotype representation

Permutation Representation

In many problems, the solution is represented by an order of elements. In such cases permutation representation is the most suited.

A classic example of this representation is the travelling salesman problem (TSP). In this the salesman has to take a tour of all the cities, visiting each city exactly once and come back to the starting city. The total distance of the tour has to be minimized. The solution to this TSP is naturally an ordering or permutation of all the cities and therefore using a permutation representation makes sense for this problem.

RAMAIAH
Institute of Technology

Advantages of Genetic Algorithm

1. Does not require any derivative information (which may not be available for many real-world problems).
2. Is faster and more efficient as compared to the traditional methods.
3. Has very good parallel capabilities.
4. Optimizes both continuous and discrete functions and also multi-objective problems.
5. Provides a list of “good” solutions and not just a single solution.
6. Always gets an answer to the problem, which gets better over the time.
7. Useful when the search space is very large and there are a large number of parameters involved.

Limitations of Genetic Algorithm

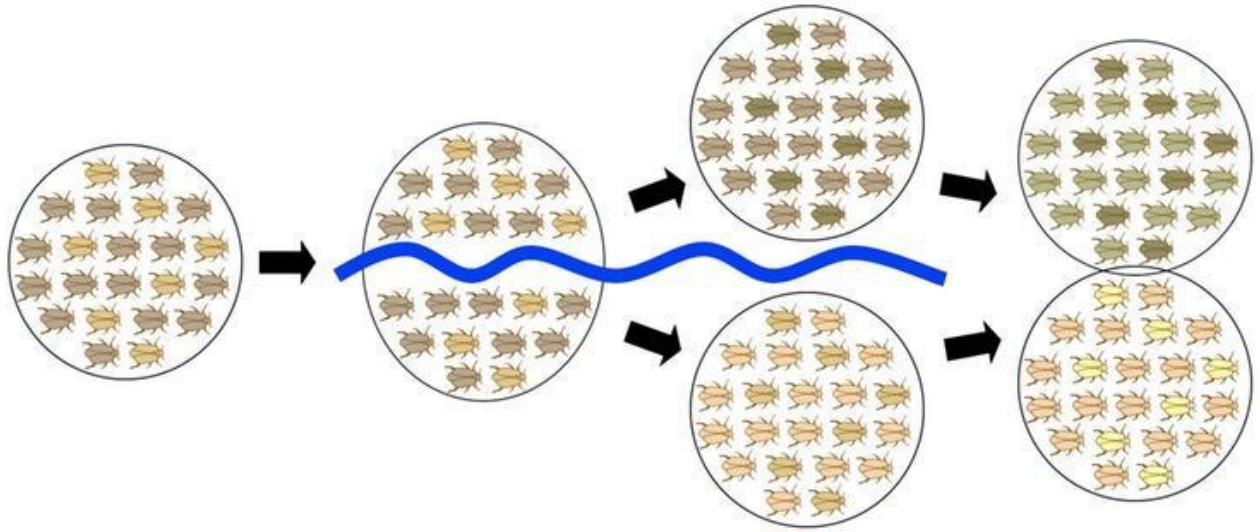
1. GAs are not suited for all problems, especially problems which are simple and for which derivative information is available.
2. Fitness value is calculated repeatedly which might be computationally expensive for some problems.
3. Being stochastic, there are no guarantees on the optimality or the quality of the solution.
4. If not implemented properly, the GA may not converge to the optimal solution.

Niching

- So far we have discussed that **look into optimization of just one parameter (unimodal)**.
- There are several complex problems where in a set of parameters (multimodal) that may be **dependent on each other** is to be optimized.
- Such are known to use niching. **Multi-objective optimization problems.**
- Niching techniques are **the extension of standard evolutionary algorithms (EAs) to multi-modal domains**, in scenarios where the location of multiple optima is targeted and where EAs tend to lose population diversity and converge to a solitary basin of attraction.

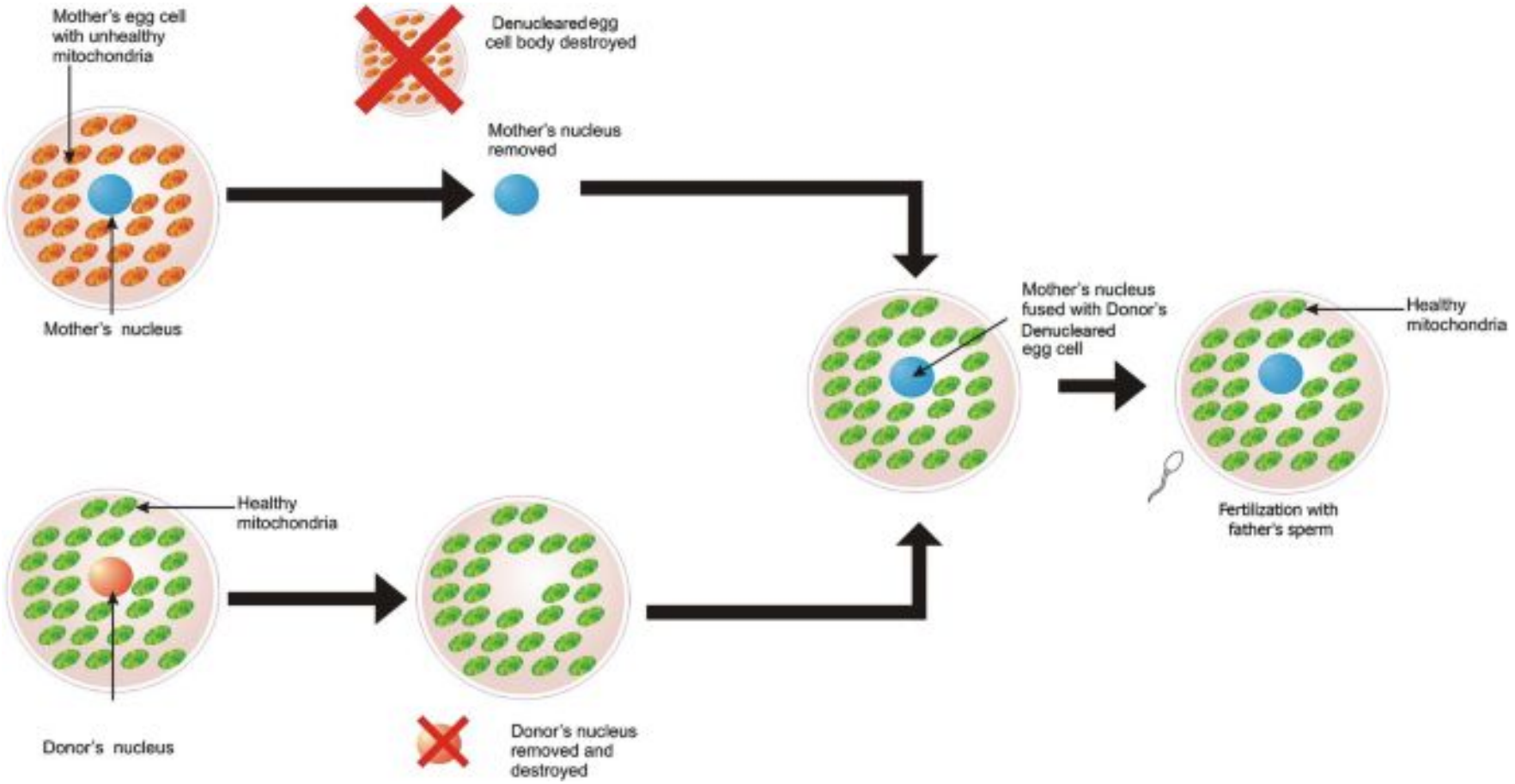
Speciation

- Speciation is the **development of one or more species** from **existing ones**.
- Basically, a genetic algorithm (that implements speciation) will **divide the population of candidates into a number of species**. Essentially, each species will be a group of solutions that are allowed to crossover with each other. Candidates belonging to different species rarely mate.



RAMAIAH
Institute of Technology

Speciation



Evolving Neural Networks

- Though neural networks have been used in a wide variety of applications, they suffer from some basic drawbacks.
- We look at two of the major problems faced while using ANN and see how genetic algorithms can be used to overcome them.

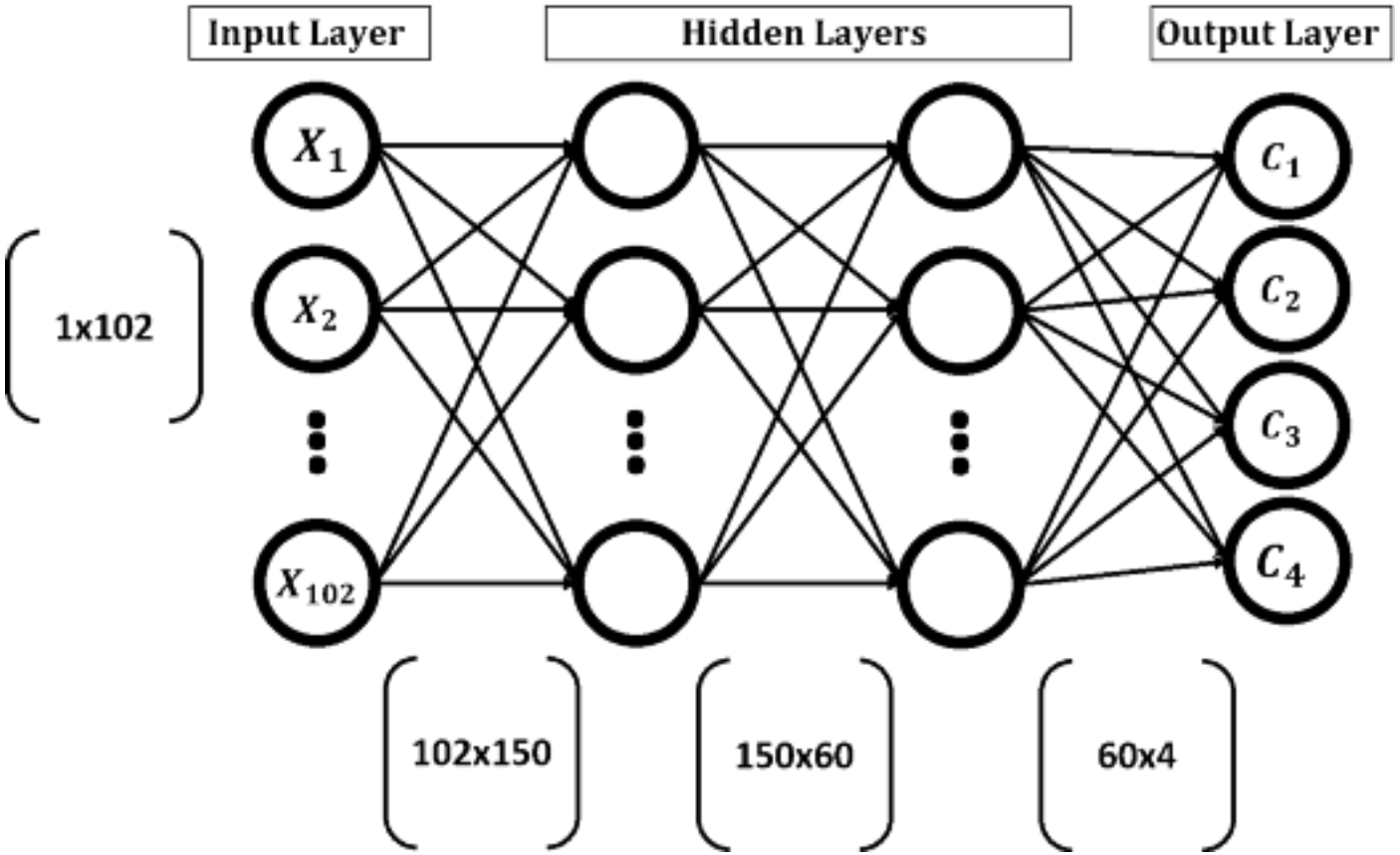
RAMAIAH
Institute of Technology

Evolving Neural Networks

- GA creates multiple solutions to a given problem and evolves them through a number of generations. Each solution holds all parameters that might help to enhance the results.
- For ANN, weights in all layers help achieve high accuracy. Thus, a single solution in GA will contain all weights in the ANN. According to the network structure discussed in the previous tutorial and given in the figure below, the ANN has 4 layers (1 input, 2 hidden, and 1 output).
- Any weight in any layer will be part of the same solution. A single solution to such network will contain a total number of weights equal to 102x150+150x60+60x4=24,540. If the population has 8 solutions with 24,540 parameters per solution, then the total number of parameters in the entire population is 24,540x8=196,320.

RAMAIAH
Institute of Technology

Evolving Neural Networks



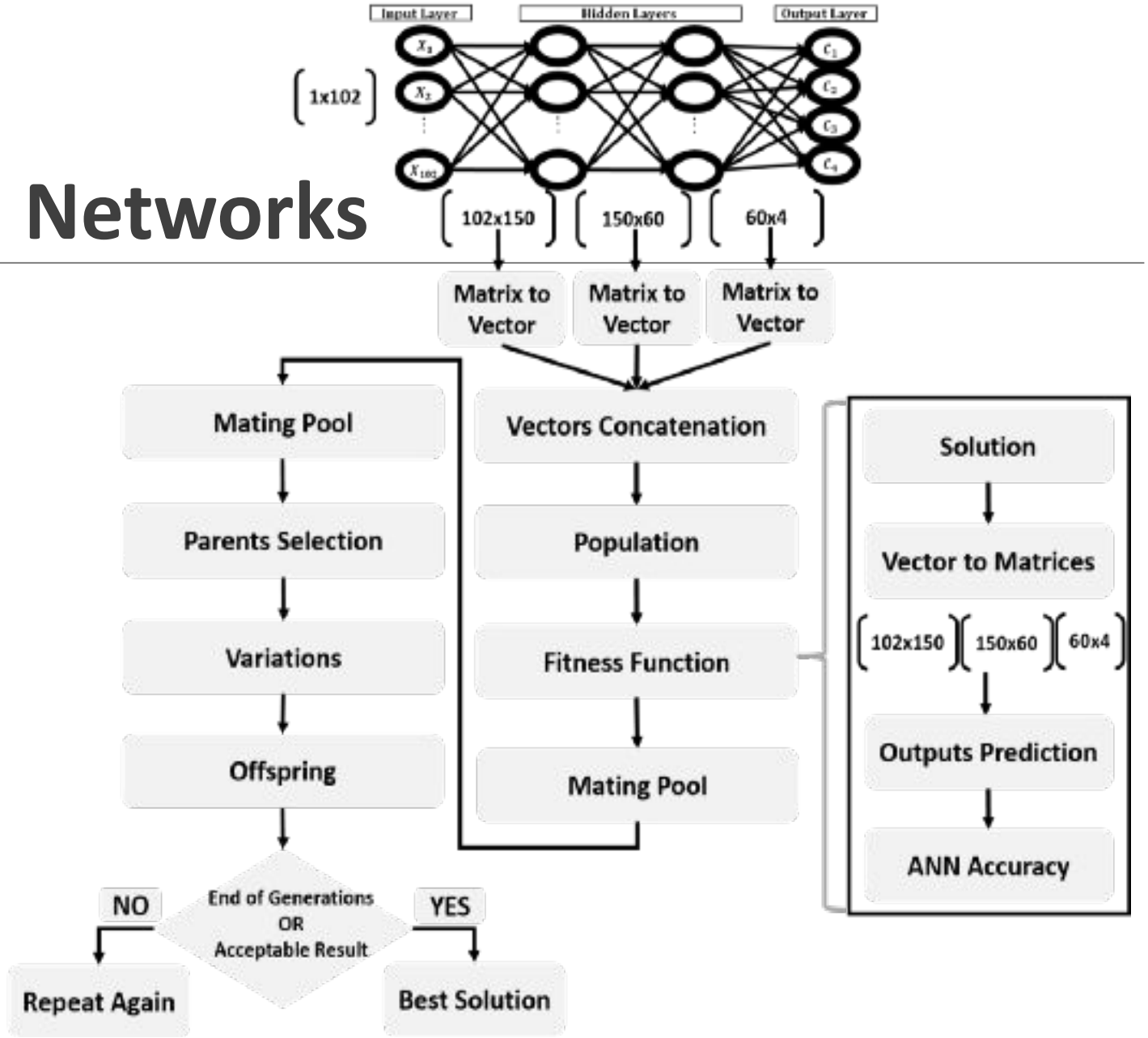
RAMAIAH
Institute of Technology

Evolving Neural Networks

- Looking at the above figure, the parameters of the network are in matrix form because this makes calculations of ANN much easier. For each layer, there is an associated weights matrix.
- Just multiply the inputs matrix by the parameters matrix of a given layer to return the outputs in such layer. Chromosomes in GA are 1D vectors and thus we have to convert the weights matrices into 1D vectors.
- Because matrix multiplication is a good option to work with ANN, we will still represent the ANN parameters in the matrix form when using the ANN. Thus, matrix form is used when working with ANN and vector form is used when working with GA.
- This makes us need to convert the matrix to vector and vice versa. The next figure summarizes the steps of using GA with ANN. This figure is referred to as the **main figure**.

RAMAIAH
Institute of Technology

Evolving Neural Networks



RAMAIAH
Institute of Technology

Ant algorithm

- In computer science and operations research, the **ant colony optimization** algorithm (**ACO**) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs.
- Artificial ants stand for multi-agent methods inspired by the behavior of real ants. The pheromone-based communication of biological ants is often the predominant paradigm used.
- Combinations of artificial ants and local search algorithms have become a method of choice for numerous optimization tasks involving some sort of graph, e.g., vehicle routing and internet routing.

RAMAIAH
Institute of Technology

Ant algorithm

- As an example, ant colony optimization is a class of optimization algorithms modeled on the actions of an ant colony.
- Artificial 'ants' (e.g. simulation agents) locate optimal solutions by moving through a parameter space representing all possible solutions.
- Real ants lay down pheromones directing each other to resources while exploring their environment. The simulated 'ants' similarly record their positions and the quality of their solutions, so that in later simulation iterations more ants locate better solutions.
- One variation on this approach is the bees algorithm, which is more analogous to the foraging patterns of the honey bee, another social insect.

RAMAIAH
Institute of Technology

Outline

Robotics

- Introduction
- Hardware Perception
- Planning to Move
- Planning Uncertain Movement
- Moving, Robotic Software Architecture
- Application Domains

RAMAIAH
Institute of Technology

Introduction to Robotics

- Robots are physical agents that perform tasks by manipulating the physical world.
- To do so, they are equipped with effectors such as legs, wheels, joints, and grippers.
- Effectors have a single purpose: to assert physical forces on the environment.
- Robots are also equipped with sensors, which allow them to perceive their environment.

Effectors	Actuators
Assert a force on the environment	Communicates a command to an effector

RAMAIAH
Institute of Technology

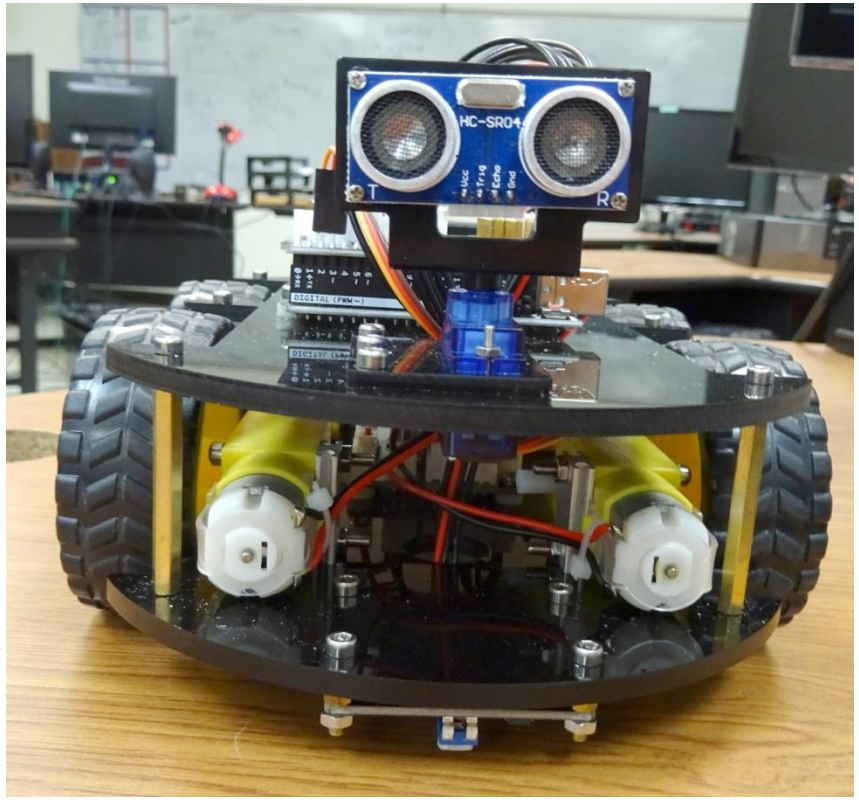
Types of Robots

1. Manipulators

- Anchored to the workplace.
- Common industrial robots.

2. Mobile Robots

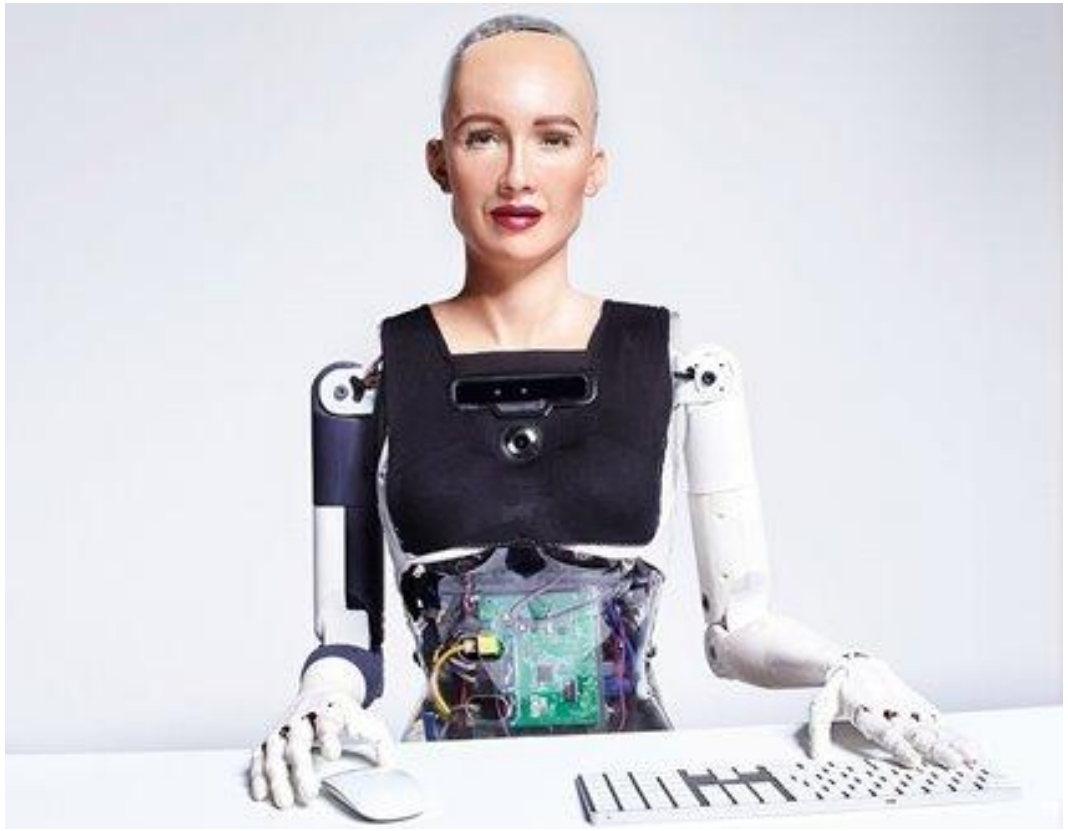
- Move using wheels, legs, etc.
- Examples: delivering food in hospitals, autonomous navigation, surveillance, etc.



RAMAIAH
Institute of Technology

Types of Robots

3. Hybrid (mobile with manipulators)
   Examples: humanoid robot
   (physical design mimics human torso)
   Made by Honda Corp. in Japan.



RAMAIAH
Institute of Technology

Robot Hardware

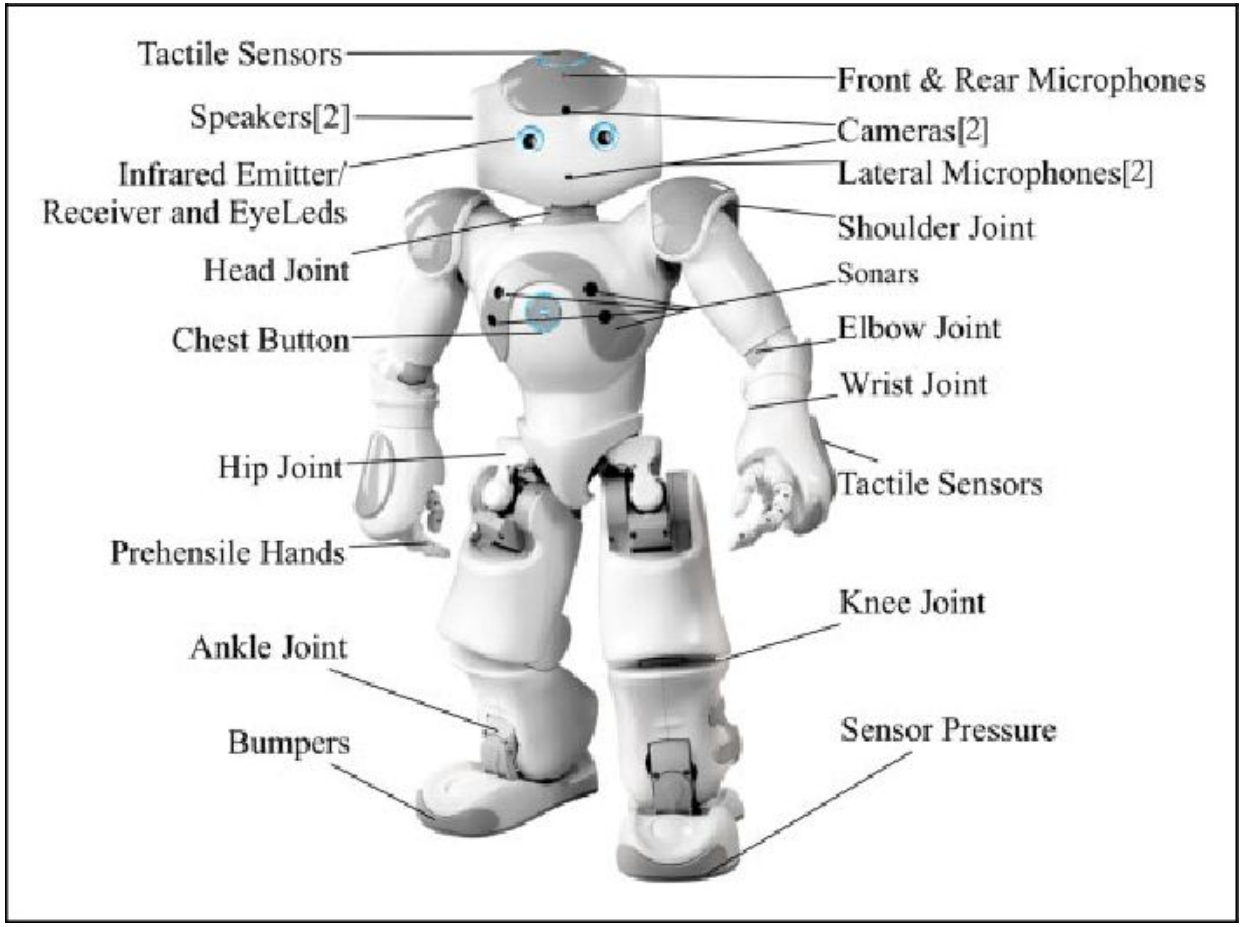
Sensors:

a. Passive sensors.

True observers such as cameras.

b. Active sensors

Send energy into the environment, like sonars.



Sensors

Examples of sensors:

- Tactile sensors (whiskers, bump panels)
- Global Positioning System
- Imaging sensors
- Odometry (distance travelled)

RAMAIAH
Institute of Technology

Planning to Move

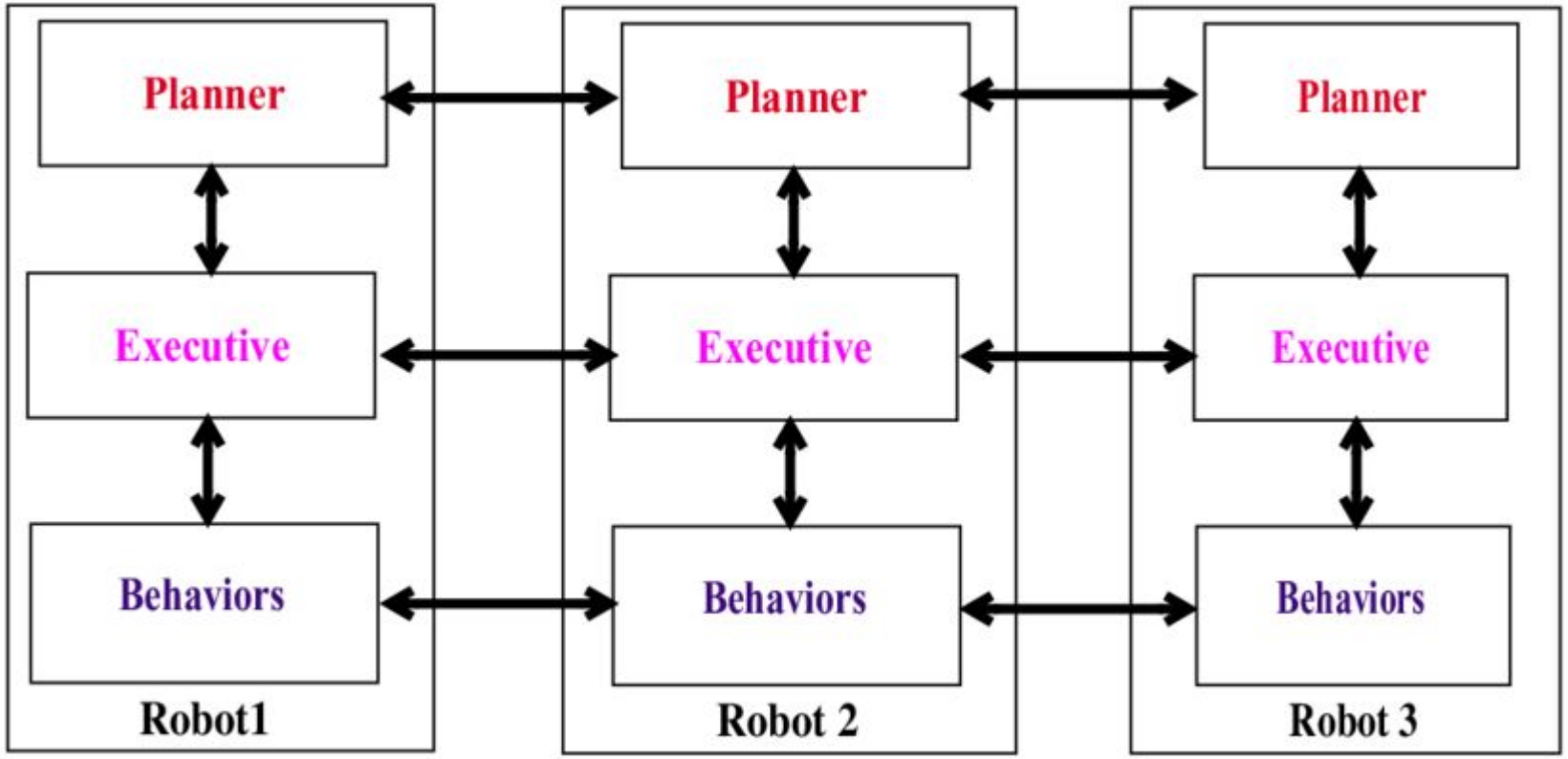
Types of motion:

a.Point-to-Point.
Deliver robot to target location.
b.Compliant motion.
Move while in contact to an obstacle
(robot pushing a box).

Dynamics and Control

- Keeping a robot on track is not easy.
- Use a controller to keep the robot on track.
- Controllers that provide a force in negative proportion to the observed error are known as P controllers.

Robotic Software



RAMAIAH
Institute of Technology

Robotic Software

- The reactive layer provides low-level control to the robot. It is characterized by tight sensor-action loop. Its decision cycle is often on the order of milliseconds.
- The executive layer serves as the glue between reactive layer and the deliberative layer. It accepts directives by the deliberative layer, and sequences them for the reactive layer.
- The deliberative layer generates global solutions to complex tasks using planning. Because of the computational complexity involved in generating such solutions, its decision cycle is often in the order of minutes. The deliberative layer uses models for decision making. Those models might be either learned from data or may utilize state information gathered at the executive layer.

Applications

- Industry and Agriculture
  - Assembly lines
  - Harvest, Mine
  - Excavate earth
- Transportation
  - Autonomous helicopters
  - Automatic wheelchairs
  - Transport food in hospitals

RAMAIAH
Institute of Technology

Applications

- Hazardous environments
  - Cleaning up nuclear waste
  - Collapse of World Trade Center
  - Transport bombs
- Exploration
  - Surface of Mars
  - Under the sea
  - Military activities
- Health Care (surgery)
- Personal Services

RAMAIAH
Institute of Technology

Applications

- Entertainment
  Dog-like robots
- Human Augmentation

RAMAIAH
Institute of Technology

Outline

- **Philosophical Foundations:**
  - Weak and Strong AI
  - The Ethics and Risks of Developing AI
- **AI: The Present and Future**

RAMAIAH
Institute of Technology

Philosophical Foundations: Weak and Strong AI

The Problem:

- ❑ Human intelligence is currently not fully understood
- ❑ There is no method of determining when and how much machine is actually intelligent

The Divisions:

- ❑ AI's definition leads to divisions what AI refers to

Two General Types:

Weak AI

Strong AI

RAMAIAH
Institute of Technology

Weak AI

- Weak AI refers to AI that only simulates human thoughts and actions
- Actions, decisions, and ideas are programmed into it
- All current forms of AI are Weak AI

Strong AI

- Weak AI refers to AI that only simulates human thoughts and actions
- Strong AI refers to AI that matcher or exceeds human intelligence
- Example: The robots form the movies Matrix, Terminator, I Robot, etc
- Also called “True AI”, as they are truly intelligent
- They don’t just simulate humans, they are intelligent on their own

RAMAIAH
Institute of Technology

The Ethics and Risks of Developing AI

1. Unemployment. What happens after the end of jobs?

The hierarchy of labor is concerned primarily with automation. As we've invented ways to automate jobs, we could create room for people to assume more complex roles, moving from the physical work that dominated the pre-industrial globe to the cognitive labor that characterizes strategic and administrative work in our globalized society.

2. Inequality. How do we distribute the wealth created by machines?

Our economic system is based on compensation for contribution to the economy, often assessed using an hourly wage. The majority of companies are still dependent on hourly work when it comes to products and services. But by using artificial intelligence, a company can drastically cut down on relying on the human workforce, and this means that revenues will go to fewer people. Consequently, individuals who have ownership in AI-driven companies will make all the money.

RAMAIAH
Institute of Technology

The Ethics and Risks of Developing AI

3. Humanity. How do machines affect our behavior and interaction?

In this challenge, human raters used text input to chat with an unknown entity, then guessed whether they had been chatting with a human or a machine.

4. Artificial stupidity. How can we guard against mistakes?

Intelligence comes from learning, whether you’re human or machine. Systems usually have a training phase in which they "learn" to detect the right patterns and act according to their input. Once a system is fully trained, it can then go into test phase, where it is hit with more examples and we see how it performs.

RAMAIAH
Institute of Technology

The Ethics and Risks of Developing AI

5. Racist robots. How do we eliminate AI bias?

Though artificial intelligence is capable of a speed and capacity of processing that's far beyond that of humans, it cannot always be trusted to be fair and neutral. Google and its parent company Alphabet are one of the leaders when it comes to artificial intelligence, as seen in Google's Photos service, where AI is used to identify people, objects and scenes. But it can go wrong, such as when a camera missed the mark on racial sensitivity, or when a software used to predict future criminals showed bias against black people.

RAMAIAH
Institute of Technology

The Ethics and Risks of Developing AI

6. Security. How do we keep AI safe from adversaries?

The more powerful a technology becomes, the more can it be used for nefarious reasons as well as good. This applies not only to robots produced to replace human soldiers, or autonomous weapons, but to AI systems that can cause damage if used maliciously. Because these fights won't be fought on the battleground only, cybersecurity will become even more important. After all, we're dealing with a system that is faster and more capable than us by orders of magnitude.

RAMAIAH
Institute of Technology

AI: The Present and Future

The Present:

- In recent years, we have seen a significant surge in Artificial Intelligence technologies, such as robotics being used for surgeries and the third generation of **AI – The Era of Deep Learning**. Therefore, we can see innovation happening at a fast pace, with leaders such as Google using deep learning algorithms to create predictive models.

Furthermore, AI is already being used many scale applications. The most common examples are:

- Vacuum cleaners
- Smart cars
- Self-driving cars
- Personal assistant robots

RAMAIAH
Institute of Technology

AI: The Present and Future

The Present:

- These are representative of the uses of AI today, and we are regularly seeing new milestones in AI technology.
- AI is transforming many aspects of business functions, and software development is no exception to this. Traditionally, developing a computer program requires you to specify exactly what you want the system to do and then engineer all of the features yourself. However, I'm sure you know that some tasks are too difficult to teach computers to do. Enter AI techniques such as **Machine Learning and Deep Learning**. A machine learning model can deduce what features and patterns are important, without a human encoding this knowledge. The outputs of some ML models can even surprise humans and highlight details that we hadn't thought about.

RAMAIAH
Institute of Technology

AI: The Present and Future

The Future:

- In the future, AI will continue to grow in a way that will enable new business models and functionalities.
- It will also play a key role in improving quality of life through allowing humans to complete tasks more efficiently and in less time.
- Although we cannot be sure of the exact future, it is quite evident that interacting with AI will soon become an everyday activity. These interactions could help our society evolve, particularly in regards to automated transportation, solving climate change, and improving the care of the elderly. Beyond these few impacts, there are even more ways that AI technology can influence our future.

- **Are there any other ways that you think Artificial Intelligence will change or shape the future?**
- Comment below and tell us your thoughts!

RAMAIAH
Institute of Technology

References

1. https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3
2. https://analyticsindiamag.com/10-real-life-applications-of-genetic-optimization/

RAMAIAH
Institute of Technology

Thank you