**1. Explain the importance of genetic algorithms in Artificial Intelligence.**

1. Genetic algorithms have the ability to deliver a “good-enough” solution “fast enough”.
2. Evolution is known to be successful robust method for adaptation with biological systems.

Solving Difficult Problems:

1. GAs prove to be an efficient tool to provide usable near-optimal solutions in a short amount of time.
2. Faster and more efficient as compared to the traditional methods.
3. Optimizes both continuous and discrete functions and also multi-objective problems.
4. Provides a list of “good” solutions and not just a single solution. Always gets an answer which gets better over the time.
5. Useful when the search space is very large and there are a large number of parameters involved.

**Genetic algorithms have many applications, some of them are –**

Recurrent Neural Network

Mutation testing

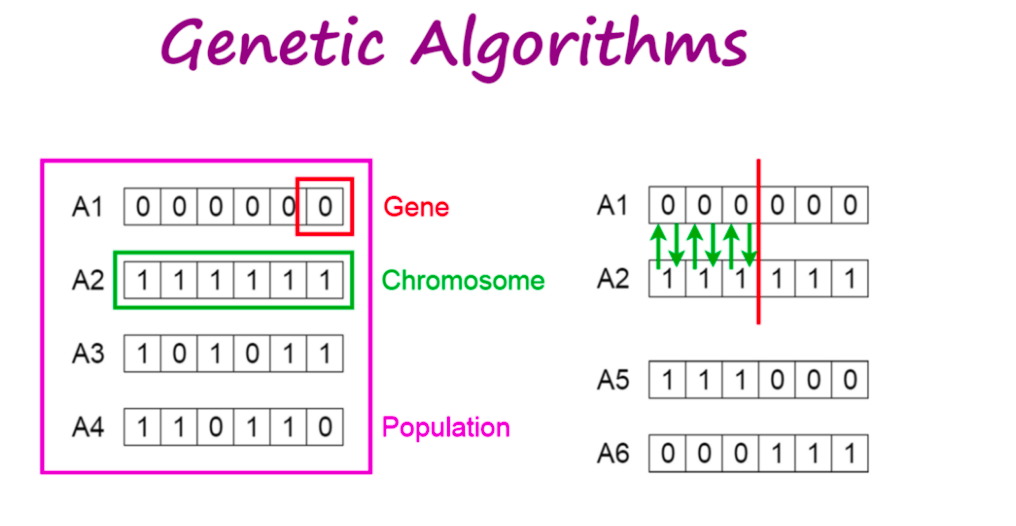
Code breaking

Filtering and signal processing

Learning fuzzy rule base etc

**2. Write a note on genetic algorithms ( 5 marks)** [**link**](https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_introduction.htm)

A genetic algorithm is a search heuristic that is inspired by Charles Darwin’s theory of natural evolution. This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation.



The process of natural selection starts with the selection of fittest individuals from a population. They produce offspring which inherit the characteristics of the parents and will be added to the next generation. If parents have better fitness, their offspring will be better than parents and have a better chance at surviving. This process keeps on iterating and at the end, a generation with the fittest individuals will be found.

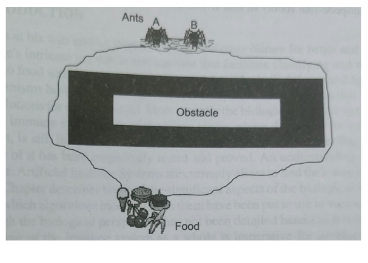
This notion can be applied for a search problem. We consider a set of solutions for a problem and select the set of best ones out of them.

**Five phases are considered in a genetic algorithm.**

1. Initial population
2. Fitness function
3. Selection
4. Crossover
5. Mutation

**2.Explain the Ant’s algorithm and list at least two applications of the same.**

**(Sure shot Question)**

****

Ants are capable of navigating complex terrains in search of food. They also find their way back to the nest.Over a period of time a colony of ants are able to find the best or shortest path between the food source and the nest. So how do they achieve this?

As they navigate they keep laying a chemical called pheromones which tend to modify their environment and serve as a means for communication amongst them in the colony.

• Pheromones are chemicals that are volatile and give way over a period of time.

• All ants choose to move over tracks of high pheromone concentration.

• In the beginning each ant goes in search of food and as they move the pheromone is laid along the path. When an ant finds the food source it starts its return journey along the same path and adds the pheromone concentration along it.

• Since the colony comprises a large number of ants, a parallel search ensues.

• Chances are that several of them discover the food source through different paths.

• Naturally the ant that found the closest path would over a period of time shuttle up and down more times than its counterparts.

• This increases the pheromone concentration of the shortest path and forces other ants too choose it.

• Over a period of time only the shortest path exists while other paths fade away due to pheromone volatility.

• Figure depicts two ants A and B using this technique.

• Ant A will definitely reach back to the starting point faster than ant B, thereby depositing more pheromone over a period of time forcing the other to also follow the shortest route discovered.

**Applications of Ant’s algorithm**

• Traveling Salesman problem.

• Image Processing

**3. Formulate any two tasks of computing which would be possibly solved by applying the philosophy of biological ants and explain the same.**

1. Solving Traveling salesman problem (TSP) using Ant algorithm

In the TSP a set of locations (e.g. cities) and the distances between them are given. The problem consists of finding a closed tour of minimal length that visits each city once and only once.

To apply **ant colony optimization** (ACO) to the TSP, we consider the graph defined by associating the set of cities with the set of vertices of the graph. This graph is called a construction graph. Since in the TSP it is possible to move from any given city to any other city, the construction graph is fully connected and the number of vertices is equal to the number of cities. We set the lengths of the edges between the vertices to be proportional to the distances between the cities represented by these vertices and we associate pheromone values and heuristic values with the edges of the graph. Pheromone values are modified at runtime and represent the accumulated experience of the ant colony, while heuristic values are problem dependent values that, in the case of the TSP, are set to be the inverse of the lengths of the edges.

The ants construct the solutions as follows. Each ant starts from a randomly selected city (vertex of the construction graph). Then, at each construction step it moves along the edges of the graph. Each ant keeps a memory of its path, and in subsequent steps it chooses among the edges that do not lead to vertices that it has already visited. An ant has constructed a solution once it has visited all the vertices of the graph. At each construction step, an ant probabilistically chooses the edge to follow among those that lead to yet unvisited vertices. The probabilistic rule is biased by pheromone values and heuristic information: the higher the pheromone and the heuristic value associated to an edge, the higher the probability an ant will choose that particular edge. Once all the ants have completed their tour, the pheromone on the edges is updated. Each of the pheromone values is initially decreased by a certain percentage. Each edge then receives an amount of additional pheromone proportional to the quality of the solutions to which it belongs (there is one solution per ant).

This procedure is repeatedly applied until a termination criterion is satisfied.

1. Image processing:

The ACO algorithm is used in image processing for image edge detection and edge linking.

Edge detection:

The graph here is the 2-D image and the ants traverse from one pixel depositing pheromone. The movement of ants from one pixel to another is directed by the local variation of the image's intensity values. This movement causes the highest density of the pheromone to be deposited at the edges.

*Step 1: Initialization.* Randomly place

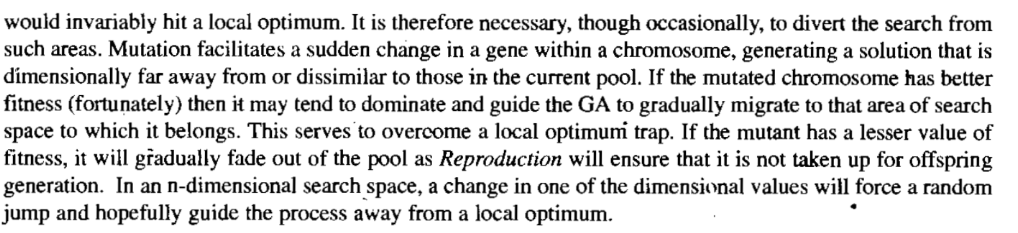
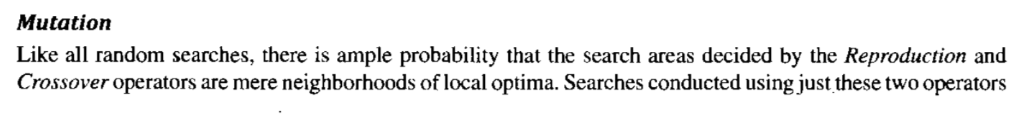
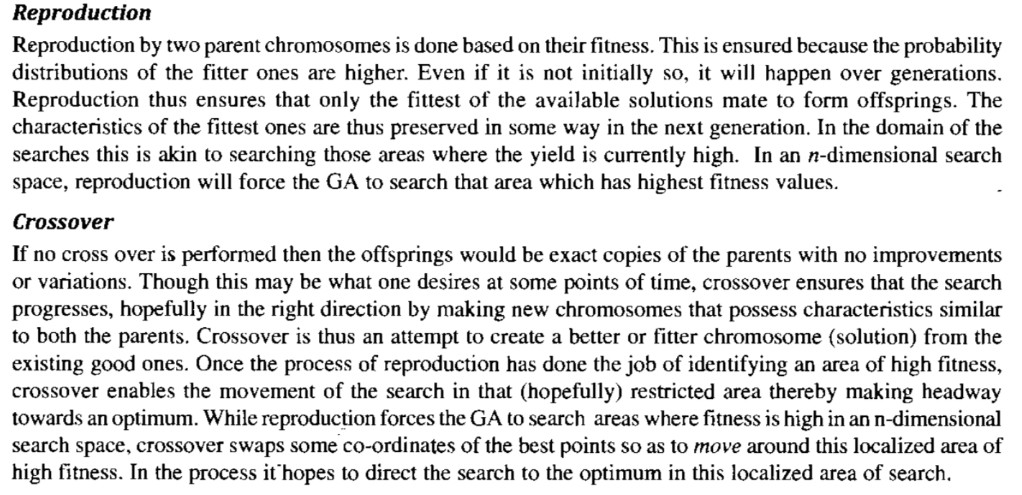
*Step 2: Construction process.* The ant's movement is based on [4-connected](https://en.wikipedia.org/wiki/4-connected_neighborhood) [pixels](https://en.wikipedia.org/wiki/Pixel) or [8-connected](https://en.wikipedia.org/wiki/8-connected) [pixels](https://en.wikipedia.org/wiki/Pixel).

*Step 3 and Step 4 : Update process.* The pheromone matrix is updated twice. in step 3 the trail of the ant is updated where as in step 4 the evaporation rate of the trail is updated

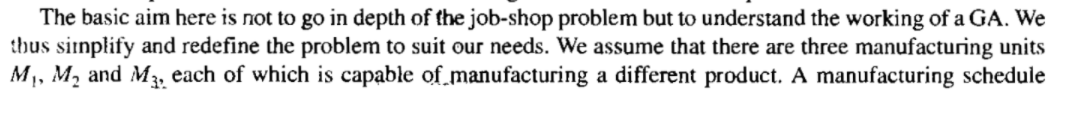
*Step 5: Decision process.* Once the K ants have moved a fixed distance L for N iteration, the decision whether it is an edge or not is based on the threshold T on the pheromone matrixτ.

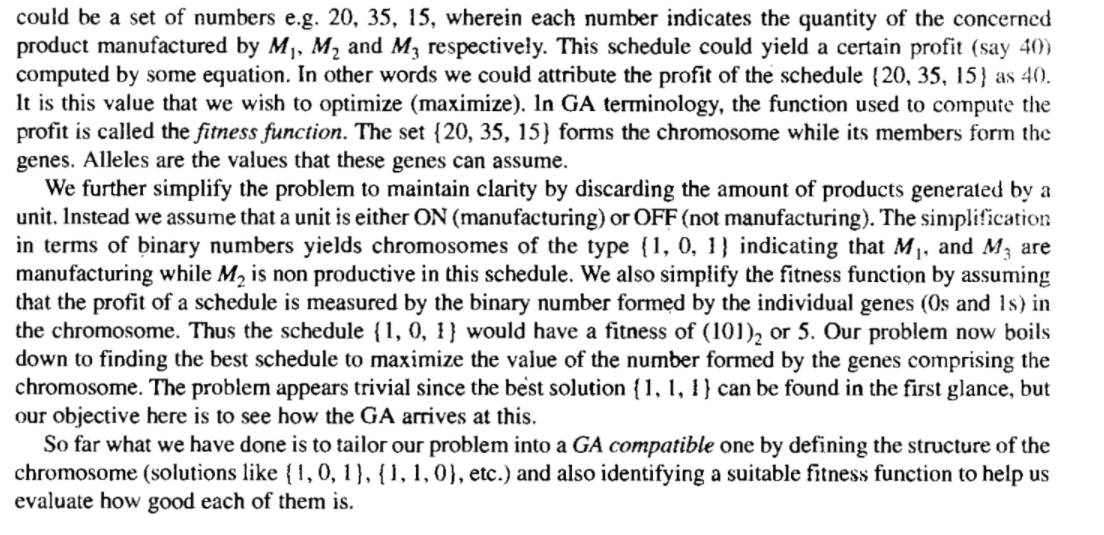
**4. Discuss the Significance of genetic operators / operations of genetic algorithms.**

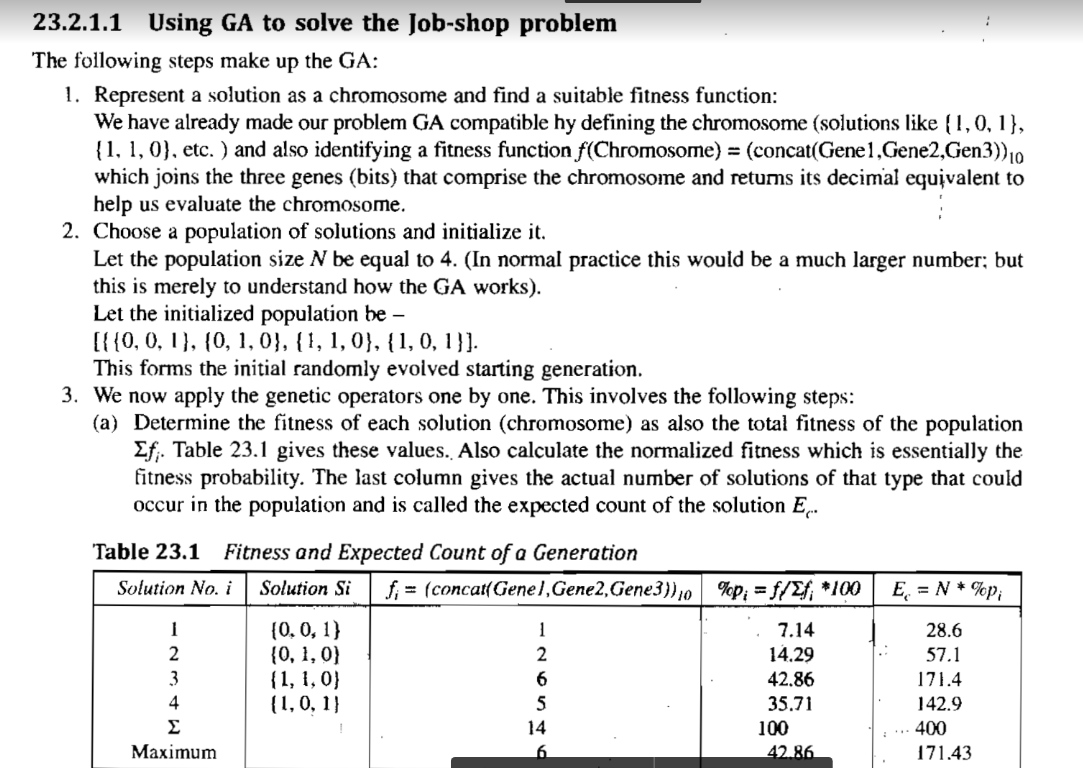
**(V imp)**

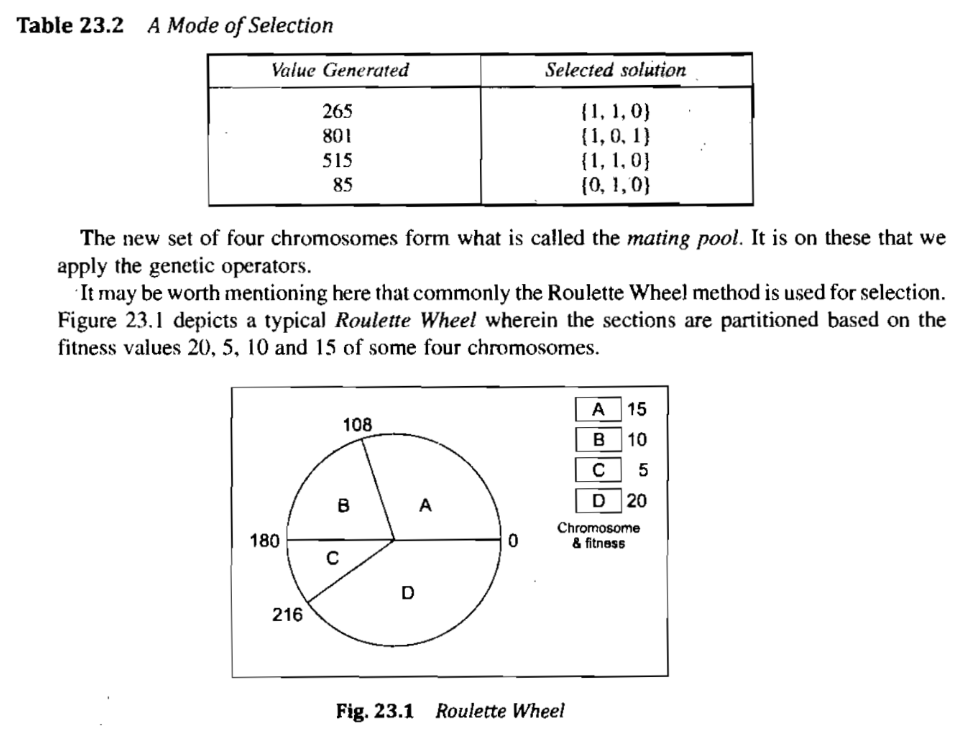
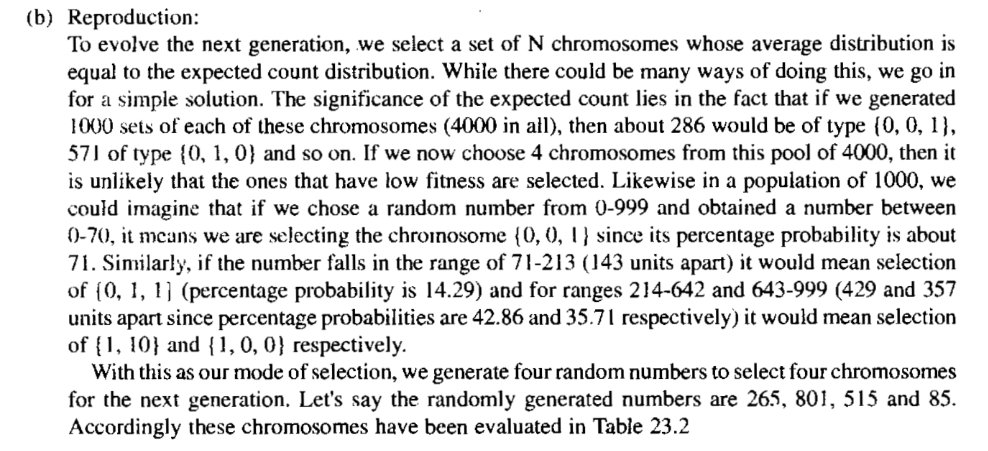
****

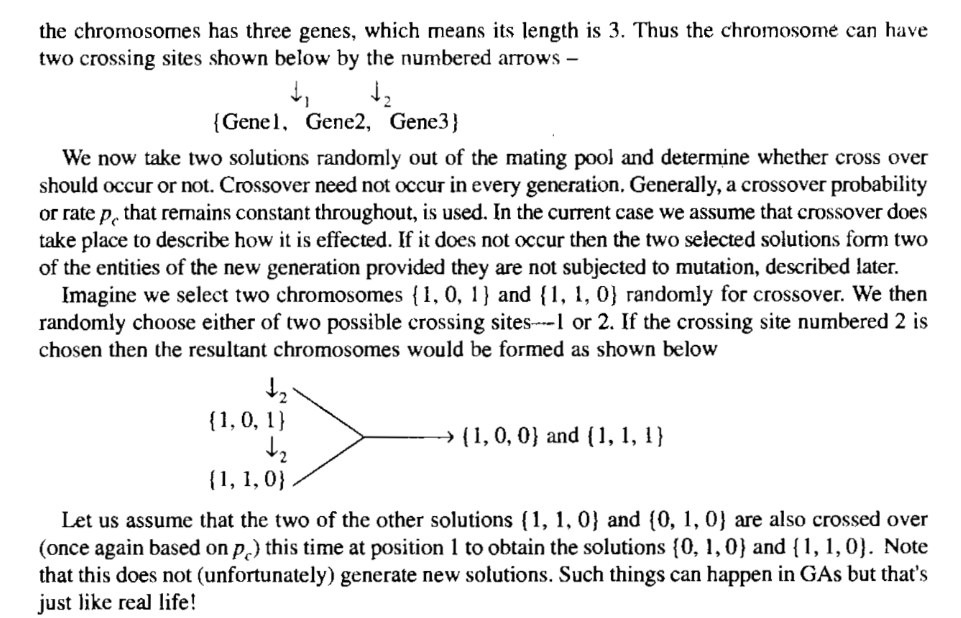
**5.Elaborate on the application of Genetic Algorithm on optimization problem of Job-Shop scheduling. (Ma’am notes** [**https://drive.google.com/file/d/1ERitdc7vtrDZN9rP2zQn8kHPvGK6sze\_/view**](https://drive.google.com/file/d/1ERitdc7vtrDZN9rP2zQn8kHPvGK6sze_/view) **)**

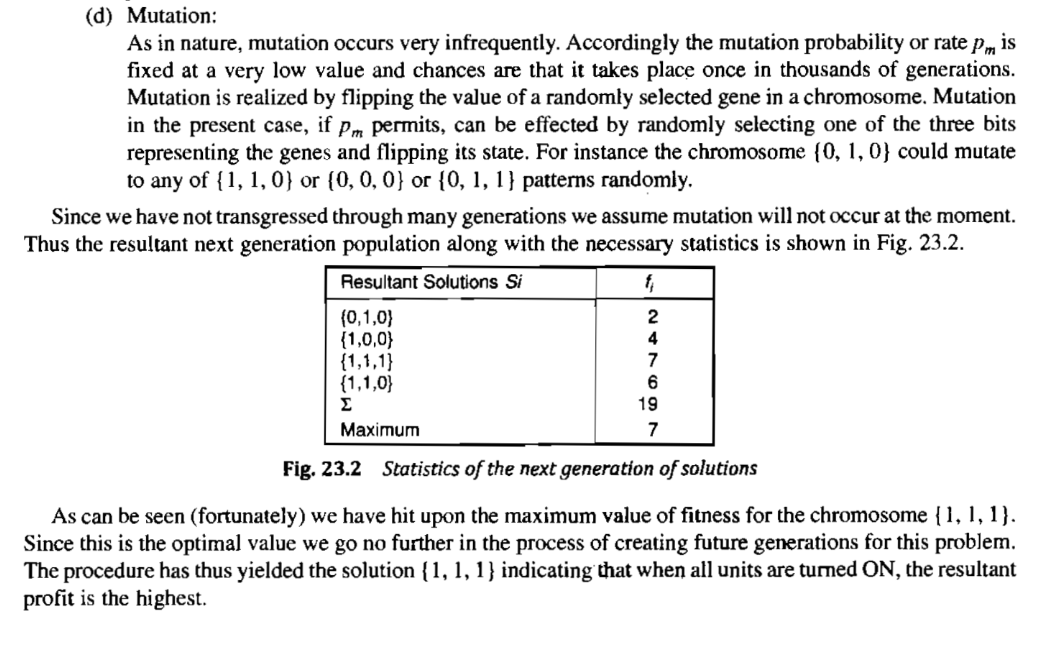




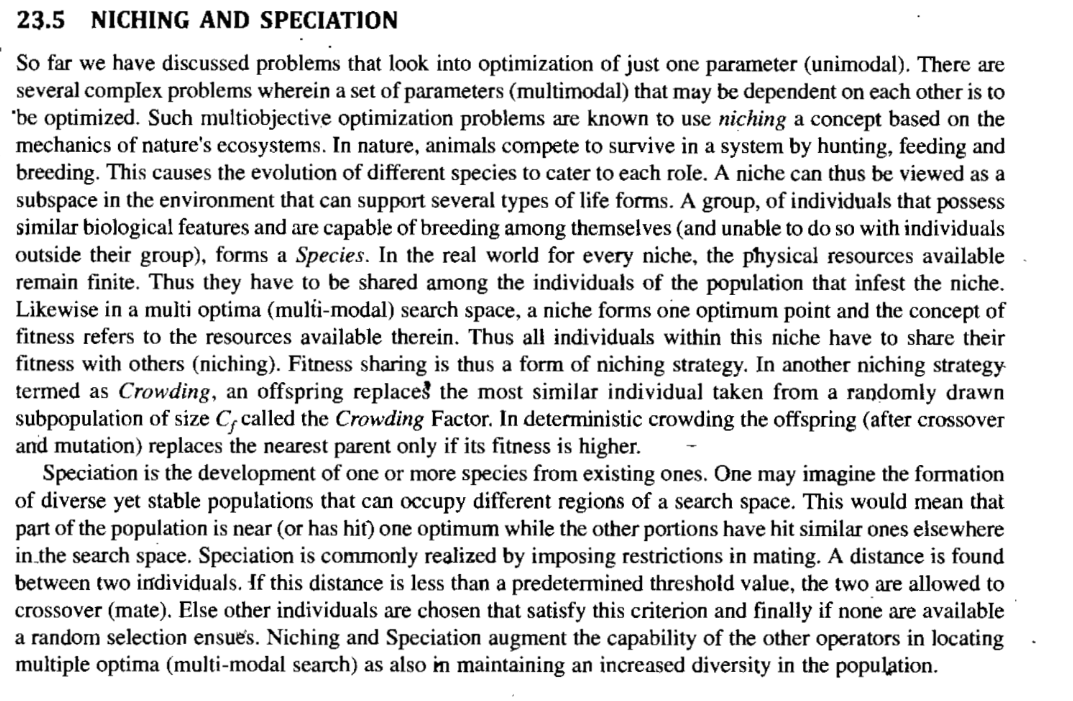
****

****

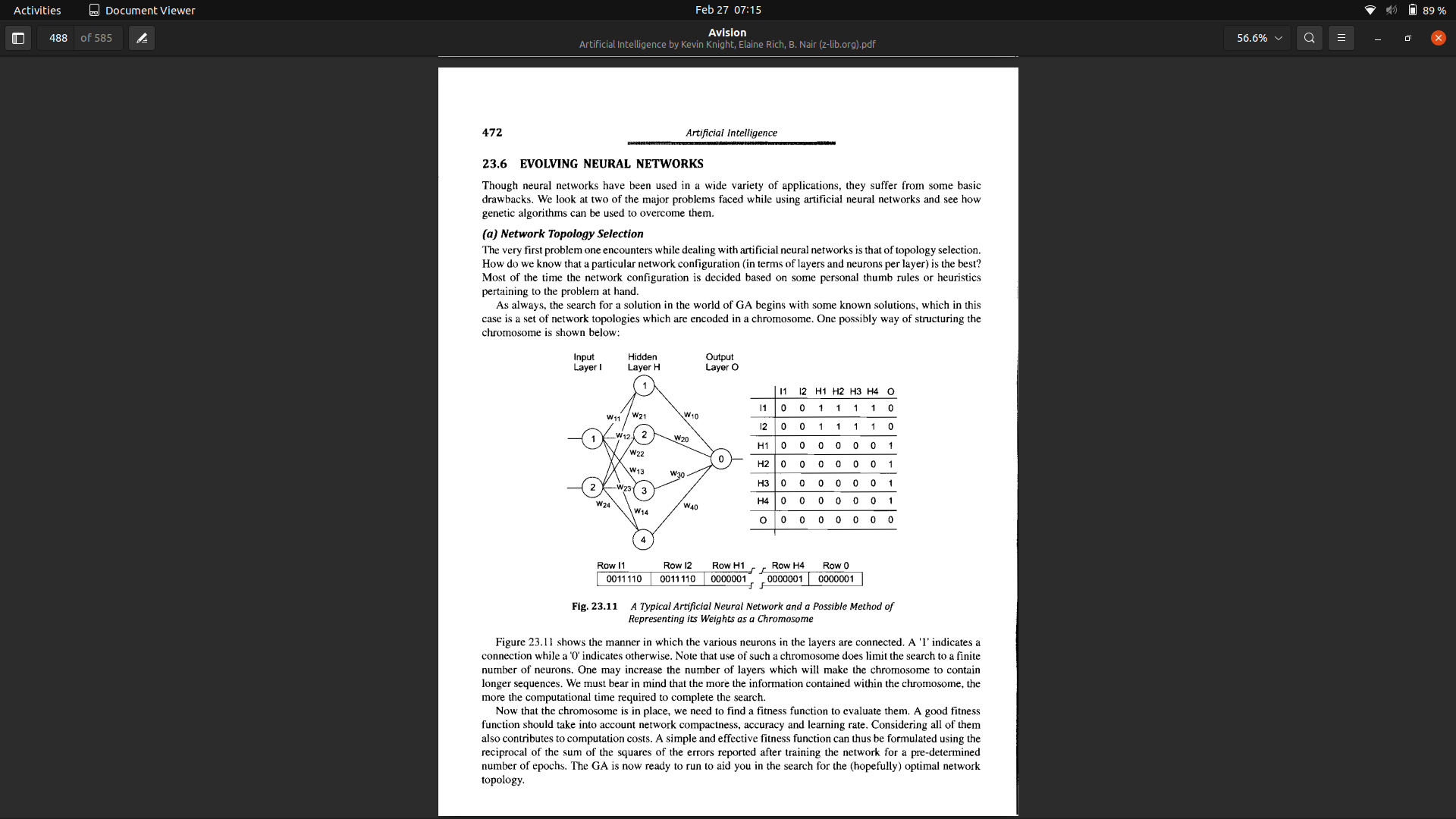
****

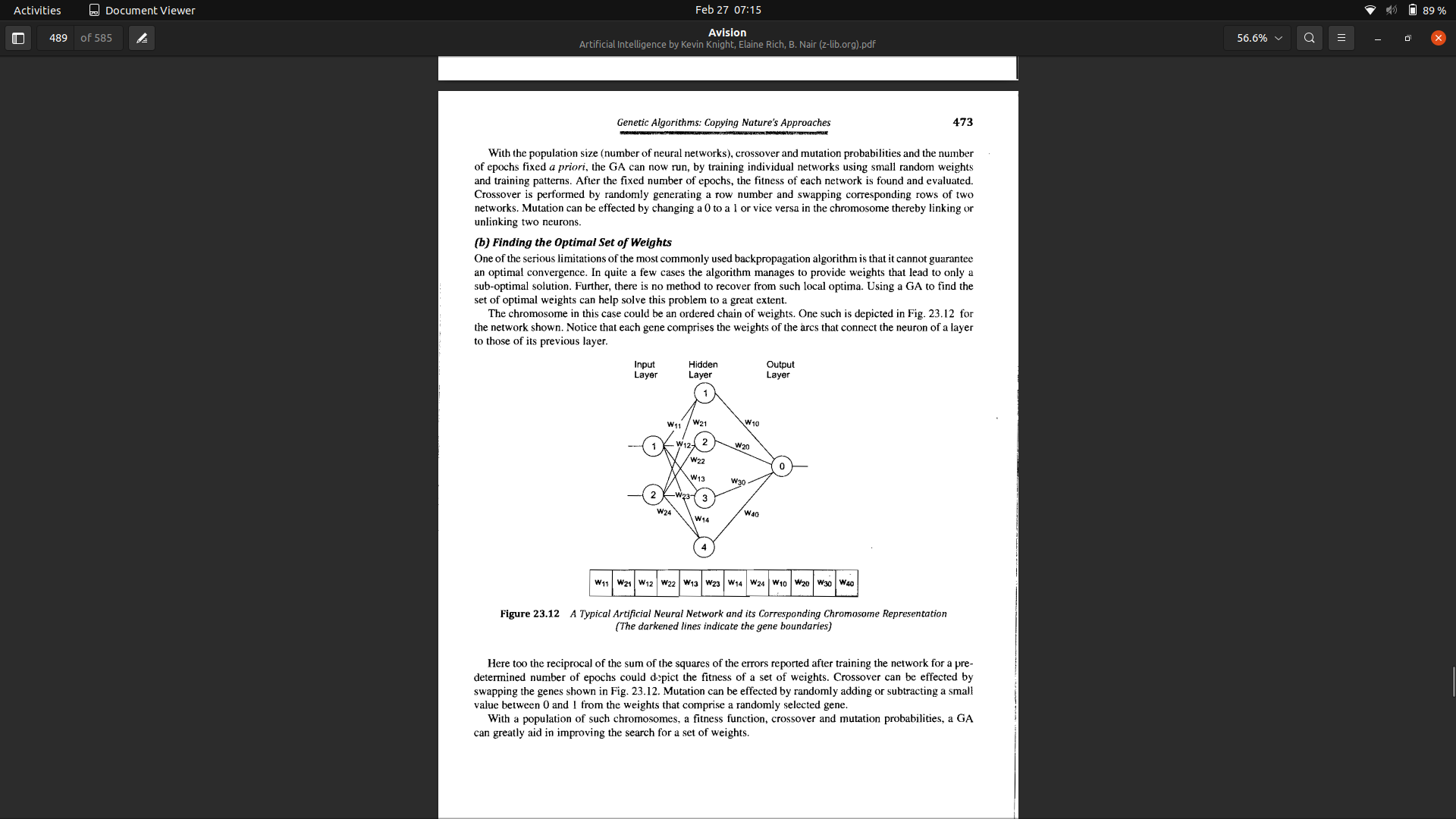
****

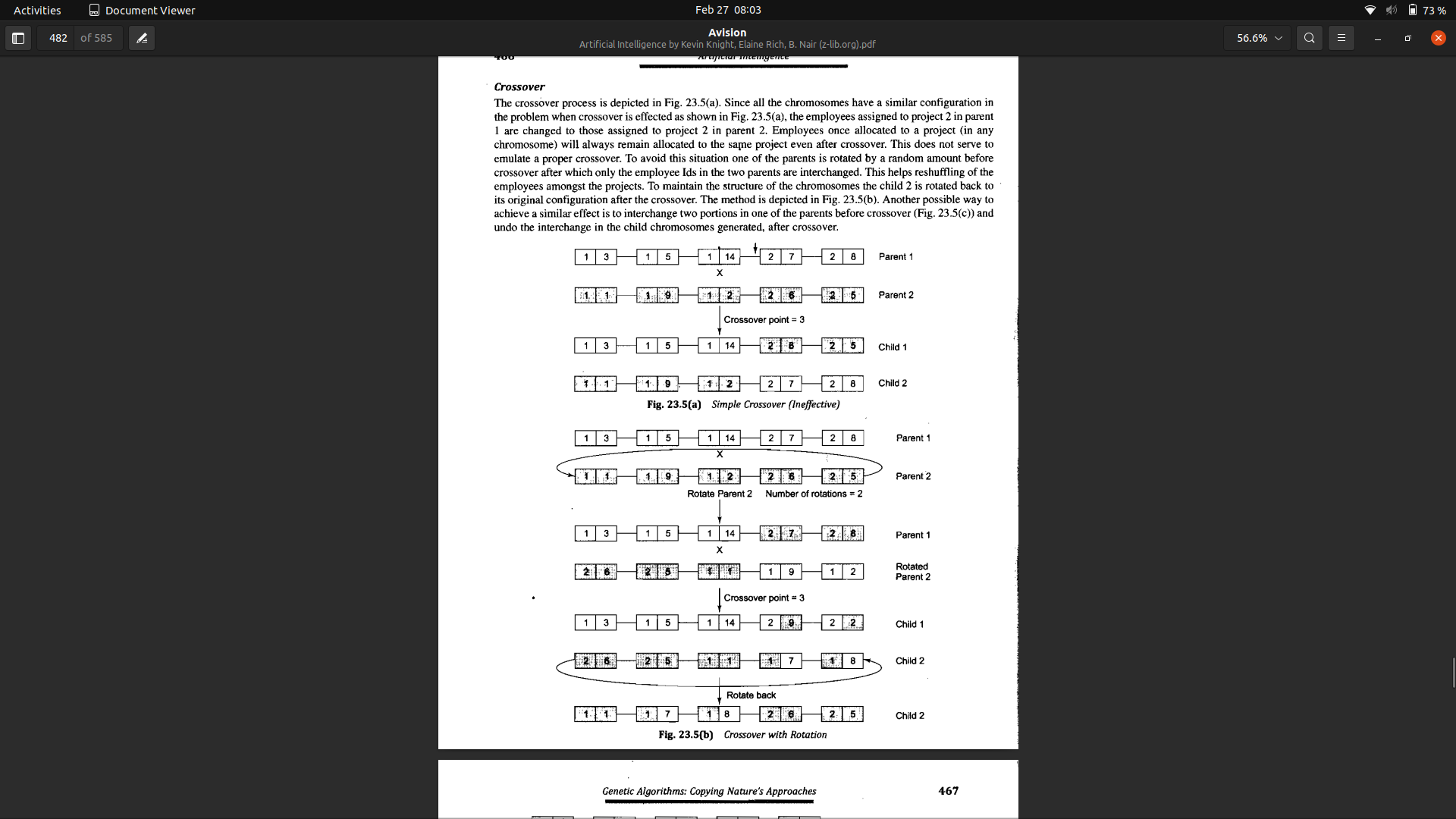
**Difference between Niching and Speciation methods in Genetic Algorithms.**

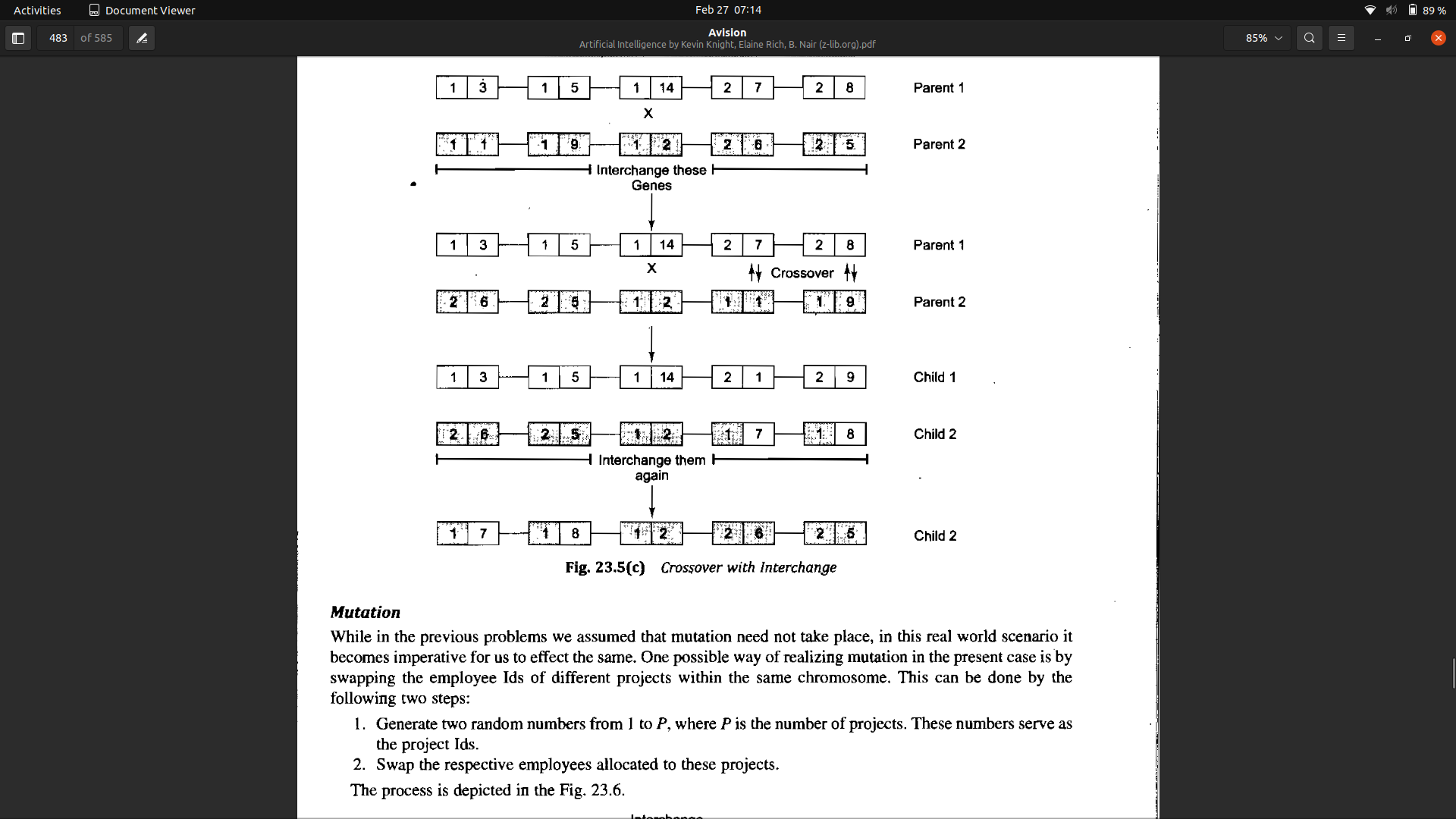
****

**6. Compare the advantages of using genetic algorithm approach over artificial Neural Networks to solve the network topology selection problem and finding the optimal set of weights problem.**

****

**7. Describe the different types of crossover process in genetic algorithms with an example.**

****

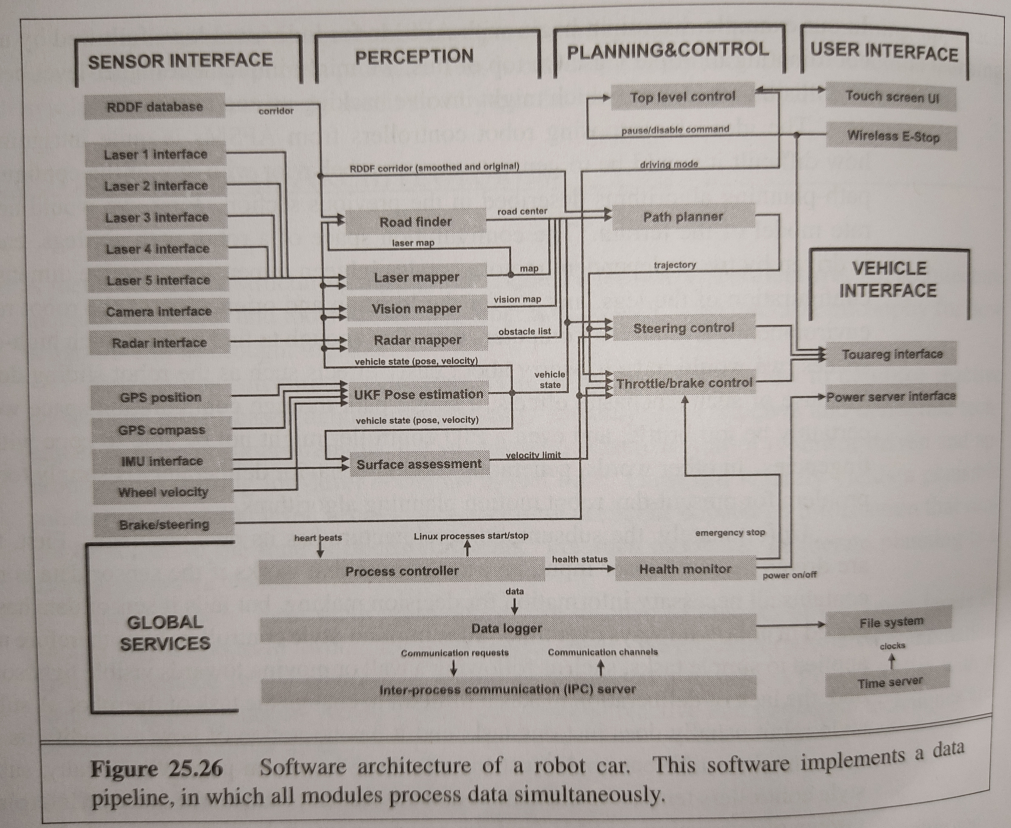
****

**8. Discuss how you apply the three basic genetic operators on the case study of “Optimization with constraints: skill based employee allocation”.**

**(Too lengthy)**

**9. Discuss in detail the subsumption and pipeline robot architectures.**

1. **Subsumption architecture**
   1. Task achieving behaviors are represented in separate layers
   2. Individual layers work on individual goals concurrently and asynchronously
   3. No global memory, bus or clock
   4. Lowest level description of a behavior is an Augmented Finite State machine
   5. A (purely reactive) behavior-based method
   6. Sound-bites
      1. The world is its own best model
      2. No central world model or global sensor representations
      3. All onboard computation is important
      4. Systems should be built incrementally
      5. No representation. No calibration, no complex computation, no high bandwidth computation
2. **Pipeline architecture**
   1. Just like the subsumption architecture, the pipeline architecture executes multiple processes in parallel.
   2. However, the specific modules in this architecture resemble those in the three-layer architecture.
   3. Figure shows the example pipeline architecture, which is used to control an autonomous car.
   4. Data enters this pipeline at the sensor interface layer. The perception layer then updates the robot’s internal models of the environment based on this data.
   5. Next these models are handed to the planning and control layer, which adjusts the robot’s internal plans and turns them into actual controls for the robot. Those are then communicated back to the vehicle through the vehicle interface layer.



**10)Categorize Robots based on their types, Manipulation, environment and the device it’s made of. Give examples for each type.**

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. Present day robotics employs a diverse set of sensors, including cameras and lasers to measure the environment, and gyroscopes and accelerometers to measure the robot’s own motion.

Most of today’s robots fall into one of three primary categories. **Manipulators**, or robot arms are physically anchored to their workplace, for example in a factory assembly line or on the International Space Station. Manipulator motion usually involves a chain of controllable joints, enabling such robots to place their effectors in any position within the workplace. Manipulators are by far the most common type of industrial robots,with approximately one million units installed worldwide. Some mobile manipulators are used in hospitals to assist surgeons. Few car manufacturers could survive without robotic manipulators, and some manipulators have even been used to generate original artwork.

The second category is the **mobile robot**. Mobile robots move about their environment using wheels, legs, or similar mechanisms. They have been put to use delivering food in hospitals, moving containers at loading docks, and similar tasks. Unmanned ground vehicles, or UGVs, drive autonomously on streets, highways, and off-road. Other types of mobile robots UAV include unmanned air vehicles (UAVs), commonly used for surveillance, crop-spraying, and military operations.Autonomous underwater vehicles (AUVs) are used in deep sea exploration. Mobile robots deliver packages in the workplace and vacuum the floors at home.

The third type of robot combines **mobility with manipulation**, and is often called a mobile anipulator. Humanoid robots mimic the human torso.HUMANOID ROBOT early humanoid robots, both manufactured by Honda Corp. in Japan. Mobile manipulators can apply their effectors further afield than anchored manipulators can, but their task is made harder because they don’t have the rigidity that the anchor provides.

The field of robotics also includes prosthetic devices (artificial limbs, ears, and eyes for humans), intelligent environments (such as an entire house that is equipped with sensors and effectors), and multibody systems, wherein robotic action is achieved through swarms of small cooperating robots.

**11)Describe two main approaches that are involved in path planning for robot movement. (10 marks)**

The path planning problem is to find a path from one configuration to another in configuration space. There are two main approaches: cell decomposition and skeletonization.

**Cell decomposition**

it decomposes the free space into a finite number of contiguous regions, called cells. These regions have the

important property that the path-planning problem within a single region can be solved by

simple means (e.g., moving along a straight line). The path-planning problem then becomes

a discrete graph-search problem,

**skeletonization.**

These algorithms reduce the robot’s free space to a one-dimensional representation, for which the planning problem is easier. This lower-dimensional representation is called a skeleton of the configuration space.

**(more content for 10 marks)**

**Cell decomposition**

The first approach to path planning uses cell decomposition—that is, it decomposes the free space into a finite number of contiguous regions, called cells. These regions have the

important property that the path-planning problem within a single region can be solved by

simple means (e.g., moving along a straight line). The path-planning problem then becomes

a discrete graph-search problem.

The simplest cell decomposition consists of a regularly spaced grid. Figure 25.16(a)

shows a square grid decomposition of the space and a solution path that is optimal for this

grid size. Grayscale shading indicates the value of each free-space grid cell—i.e., the cost of

the shortest path from that cell to the goal.

Such a decomposition has the advantage that it is extremely simple to implement, but it also suffers from three limitations. First, it is workable only for low-dimensional configuration spaces, because the number of grid cells increases exponentially with d, the number of dimensions. Sounds familiar? This is the curse!dimensionality@of dimensionality. Second, there is the problem of what to do with cells that are “mixed”—that is, neither entirely within free space nor entirely within occupied space. A solution path that includes such a cell may not be a real solution, because there may be no way to cross the cell in the desired direction in a straight line. This would make the path planner unsound. On the other hand, if we insist that only completely free cells may be used, the planner will be incomplete, because it might be the case that the only paths to the goal go through mixed cells—especially if the cell size is comparable to that of the passageways and clearances in the space. And third, any path through a discretized state space will not be smooth. It is generally difficult to guarantee that a smooth solution exists near the discrete path. So a robot may not be able to execute the solution found through this decomposition.

Cell decomposition methods can be improved in a number of ways, to alleviate some of these problems. The first approach allows further subdivision of the mixed cells—perhaps using cells of half the original size. This can be continued recursively until a path is found that lies entirely within free cells. (Of course, the method only works if there is a way to decide if a given cell is a mixed cell, which is easy only if the configuration space boundaries have relatively simple mathematical descriptions.) This method is complete provided there is a bound on the smallest passageway through which a solution must pass. Although it focuses most of the computational effort on the tricky areas within the configuration space, it still fails to scale well to high-dimensional problems because each recursive splitting of a cell creates 2d smaller cells. A second way to obtain a complete algorithm is to insist on an exact cell decomposition of the free space. This method must allow cells to be irregularly shaped where they meet the boundaries of free space, but the shapes must still be “simple” in the sense that it should be easy to compute a traversal of any free cell.

**skeletonization**

The second major family of path-planning algorithms is based on the idea of skeletonization.

These algorithms reduce the robot’s free space to a one-dimensional representation, for which

the planning problem is easier. This lower-dimensional representation is called a skeleton of

the configuration space.it is a Voronoi graph of the free space—the set of all points that are equidistant to two or more obstacles. To do path plan-

ning with a Voronoi graph, the robot first changes its present configuration to a point on the

Voronoi graph. It is easy to show that this can always be achieved by a straight-line motion

in configuration space. Second, the robot follows the Voronoi graph until it reaches the point

nearest to the target configuration. Finally, the robot leaves the Voronoi graph and moves to

the target. Again, this final step involves straight-line motion in configuration space.

Following the Voronoi graph may not give us the shortest path, but the resulting paths tend to maximize clearance. Disadvantages of Voronoi graph techniques are that they are difficult to apply to higher-dimensional configuration spaces, and that they tend to induce unnecessarily large detours when the configuration space is wide open. Furthermore, computing the Voronoi graph can be difficult, especially in configuration space, where the shapes of obstacles can be complex.

**12)Justify how uncertainty is handled in Robot motion. (Not there in ppt , but present in Prev SEE)**

In robotics, uncertainty arises from partial observability of the environment and from the stochastic (or unmodeled) effects of the robot’s actions.

**Explain how robots are used in Industry, transportation, hazardous environments, healthcare and human augmentation.**

1. Industry and Agriculture
   1. Assembly lines
   2. Harvest, Mine
   3. Excavate earth
2. Transportation
   1. Autonomous helicopters
   2. Automatic wheelchairs
   3. Transport food in hospitals
3. Hazardous environments
   1. Cleaning up nuclear waste
   2. Collapse of World Trade Center
   3. Transport bombs
4. Exploration
   1. Surface of Mars
   2. Under the sea
   3. Military activities
5. Health Care (surgery)
6. Personal Services
7. Entertainment
8. Dog-like robots
9. Human Augmentation

**List the potential threats from AI Technology to society. Also discuss how these threats can be combated.**

**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.

**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.

**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.

**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.