Module 2

CHAPTER 2

Intelligent Agents

Syllabus

Agents and Environments, The concept of rationality, The nature of environment, The structure of Agents, Types of Agents, Learning Agent.

Solving problem by Searching : Problem Solving Agent, Formulating Problems, Example Problems.

2.1 INTELLIGENT AGENT

An agent is just something that acts (agent comes from the Latin agree, to do).

In artificial intelligence, an **intelligent agent (IA)** is an autonomous entity which observes through sensors and acts upon an environment using actuators (i.e. it is an agent) and directs its activity towards achieving goals.

- Intelligent agents may also learn or use knowledge to achieve their goals.
- They may be very simple or very complex, **Example :** a reflex machine such as a thermostat is an intelligent agent.
- An agent is anything that can perceive its environment through sensors and acts upon that environment through effectors.

  Agent's structure can be viewed as :

  (1) Agent = Architecture + Agent program.

  (2) Architecture = the machinery that an agent executes on.

  (3) Agent program = an implementation of an agent function.
- IA like Rahul and Gopal are examples of intelligence as they use sensors to perceive a request made by the user and automatically collect the data from the internet without the user's help.
- That can be used to gather information about its perceived environment such as weather and time. Thus, an intelligent agent is an autonomous entity which acts upon an environment using sensors and actuates for achiėving goals. An intelligent agent may learn from the environment to achieve their goals.
- The term '**percept**' means the agent's perceptual inputs at any given instant. An agent's percept sequence is the complete history of everything that the agent has perceived. We illustrate this idea in the Fig. 2.1.1.

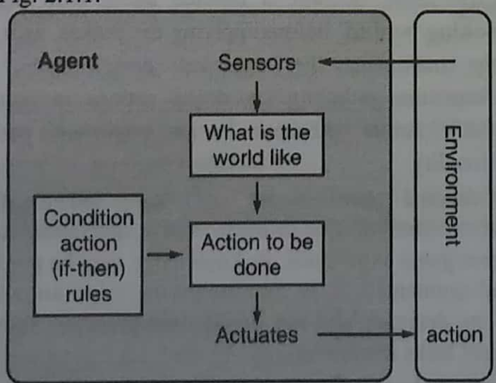


Fig. 2.1.1 : Agent's interact with environments through sensors and actuators

Thus, an agent's behaviour can be described by the **'agent function'** that maps any given percept sequence to an action.

- An intelligent agent is a programme that can make decisions or perform a service based on its environment, user input and experiences.
- These programs can be used autonomously to gather information on a regular, programmed schedule or when prompted by the user in real time.
- IA may be simple or complex - a **thermostat** is considered as an example of an intelligent agent, as is a human being, as is any system that meets the definition, such as firm, a state or total quantity.

Module 2

2.1.1 Characteristics of Intelligent Agent

UQ. Define Intelligent Agent. What are the characteristics of Intelligent Agent ?

(MU - Q. 1(a), May 18, 5 Marks)

1. **Mobility :** Using computer-Network, Intelligent agents engaged in e-commerce gather information until search parameter are complete.
2. **Goal-oriented :** Intelligent agents carry out the particular task provided by **'user statement of goals'**. It moves around from one machine to another and can react in response to their environment and takes initiative to exhibit goal directed behaviour.
3. **Independent :** Intelligent agent is self-dependent, in the sense that it functions an its own without human intervention. It makes decisions on its own and initiate them. It communicates independently with data or information and other agents and achieves the objectives and tasks an behalf of the user.
4. **Intelligent :** Intelligent agent can collect data more intelligently. They can reason out things based on the existing knowledge of its user and environment on past experiences intelligently. To evaluate condition in the external environmental intelligent agents follow present rules.
5. **Reduce net traffic :** Agents communicate and co-operate with other agents quickly. This way, they can perform the tasks, such as information searches quickly and efficiently. And network traffic gets reduce thereby.
6. **Multiple tasks :** Multiple tasks can be performed by an intelligent agent simultaneously. This helps the human from monotonous clerical work.

2.2 RATIONAL AGENT IN AI

> **UQ.** Define Rationality and Rational Agent. Give an example of rational action performed by any intelligent agent.
> **(MU - Q. 1(c), Dec. 2015, 5 Marks)**

Rational agent in AI is an agent which has **clear preference, models uncertainty**, and acts in a way to **maximize its performance measure** with all possible actions. AI is about crating rational agents to use for same theory and decision theory for various real-world phenomenons.

Intelligent agents can be grouped into the following classes based on their **degree of perceived intelligence** and capability.

Agent	Example
Simple reflex agents	The vacuum promises to sense dirt and debris on floors and clean those areas accordingly. This is an example of a simple reflex agent that operates on the condition (dirty floors) to initiate an action (vacuuming).
Model-based agent example	Some examples of items with model-based agents aboard include **Roomba vacuum cleaner** and the **autonomous car** known as **Waymo**. Both interact with their environments by using what they know, an internal model of the world and their on-board sensors as well, to make **moment-to-moment decisions** about their actions.
Goal-based agent example	Google's **Waymo driverless cars** are good examples of a goal-based agent when they are programmed with an end destination, or goal, in mind. The car will then 'think' and make the right-decisions in order to deliver the passenger where they intend to go.

2.2.1 Concept of Rationality

The concept of rationality depends upon the following four things :

(i) The performance measure that defines the criterion of success.

(ii) Prior knowledge of the environment

(iii) Agent can perform the action

(iv) The percept sequence of the agent.

Example of Vacuum cleaner Agent

Let us consider the above example of a vacuum-cleaner agent. It cleans a square if it is dirty and moves to the next square if not;

(i) The performance measure is that it cleans a square if it is dirty, at each time.

(ii) The area of space of the floor or **'geography'** of the environment is known as **'a priori'**. Clean squares stay clean and sucking cleans the current square. The **'Left'** and **'Right'** actions move the agent left and right.

(iii) Left, right and suck are the only available actions of the agent.

(iv) The agent correctly perceives its location, if there is dirt. Under these circumstances, we say that the agent is **'rational'**.

2.2.2 Omniscience and Rationality

(1) An omniscient's agent knows the exact outcome of its actions and act accordingly, but in practice it is impossible. In omniscience there is a total perfection. Rationality is not same as perfection. Rationality tries to maximise the expected performance while perfection gives actual total performance.

(2) Suppose I apply urgent brakes to my BMW car, as I want to take a halt. Car will come to halt instantly but the other car behind me may hit my car.

Does the definition of rationality say that it is OK to apply sudden brakes? First, it is not rational to apply sudden brakes, given the **uninformative percent sequence**; the risk of accident without looking behind is too great. Second, a rational agent will mention **'looking behind'** before applying the brakes, as it may help maximize the expected performance. This information gathering i.e. doing actions in order to modify future precepts is an important part of rationality.

(3) A rational agent is not only supposed to gather information but also to learn what it perceives. As the agent gains experience, its knowledge may be modified and augmented. If the agent relies on 'prior' knowledge of its designer and not on its own precepts, then the agent lacks autonomy.

(4) A rational agent must be autonomous - it is supposed to learn what it can, in order to compensate for partial prior knowledge. For example, a vacuum-cleaning

agent that can learn to locate when and where additional dirt will appear works better than one does not learn. After having sufficient experience of its environment, the behaviour of a rational agent can become effectively independent of its prior knowledge.

2.3 STRUCTURE OF INTELLIGENT AGENTS

(1) The IA structure consists of three main parts: architecture, agent function and agent programme. Architecture refers to machinery or devices that consist of **actuators and sensors**. The IA executes on this machinery. The tool allows the adjusting of image details and clarifies. Using this tool we can obtain great detail or get a smoother picture with less detail. This is the main tool to increase the contrast of the image and visualize more details of the image.

(2) A software agent has file contents, received network packages which act as sensors and display on the screen, files, sent network packets acting as actuators. A human agent has eyes, ears and other organs which act as sensors, and hands, legs, mouth and other body parts acting as actuators.

(3) The learning element is responsible for improvements. This can make a change on any of the knowledge components in the agent. One way of learning is to observe parts of successive steps in the percept sequence; from this the agent can learn how the world evolves.

(4) Agency is the capacity of individuals to act independently and to make their own free choices. The structure versus agency debate may be understood as an issue of socialization against autonomy in determining whether an individual acts as a free agent or in a manner dictated by social structure.

So far we have discussed the behaviour of the agent, i.e. the **action that is performed after a given sequence of precepts**. Now we begin to study how the inside of the agent works. The job of 'Artificial intelligence' is to design an agent program that designs the agent function-the function from precepts to action. This is 'architecture' or 'structure' of intelligent agent.

We mention four basic kinds of agent programs that cover the principles of almost all intelligent systems.

They are :

(1) Simple reflex agents
(2) Model-based reflex agents
(3) Goal-based agents; and
(4) Utility-based agents

Each of these programs combine particular component in particular ways to generate actions. Then we shall see how to convert all these agents into learning agents. This will improve the performance of their components so as to have better actions. Then we shall also describe variety of ways in which the components themselves can be represented within the agent.

2.3.1 Simple Reflex Agent

Module 2

(1) In artificial intelligence, a simple reflex agent is a **'type of intelligent agent'** that performs actions based solely on the current situation, with an intelligent agent generally being one that perceives its environment and then acts. The agent cannot learn or take past percepts into account to modify its behavior.

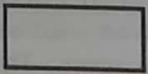
(2) The 'simple reflex agent' works on **condition-action rule**, which means it **maps the current state to action**. Since 'simple reflex agent' is based on the present condition so it is called as condition-action rule. Problems with simple reflex agents are: very limited intelligence. No knowledge of non-perceptual parts of the state is required.

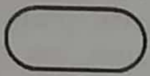
(3) Usually too big to generate and store. The simple reflex based agent is designed only to respond to the currently occurring problem. If the knowledge of the entire environment is given, then simple reflex agent is perfectly rational.

(4) This agent selects actions based on the agent's current perception, and not based on past perceptions. **For example**, if a mars lander found a rock in a specific place it needed to collect then it would collect it, **if** it was a 'simple reflex agent' **then** if it found the same rock in a different place, it would still pick it up, as it does not take into account that it has already picked it up.

(5) This is useful when a **quick automated response** is needed. Humans have a very similar reaction to fire for example; our brain pulls our hand away without thinking about any possibility that there could be danger in the path of your arm this is called as a **reflex action**.

(6) This kind of connection-action rule, written as **if** hand is in fire, **then** pulls it away. The 'simple reflex agent' has a library of such rules so that if a certain situation should arise, it is in the set of condition-action rules, the agent will know how to react with **minimal reasoning**. These agents are simple to work with but have very limited intelligence, such as picking up 2 rock samples. Refer Fig. 2.3.1.

 (rectangles) : To represent the current internal state of the agent's decision process.

 (ovals) : To represent the background information used in the process.

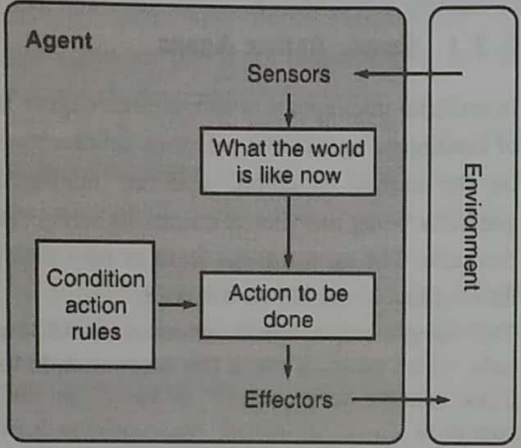


Fig. 2.3.1 : Simple Reflex Agent in AI

Function : SIMPLE–REFLEX–AGENT (percent) returns on action. **Static :** Rules, a set of condition – action rules. State ← INTERPRET – INPUT (percept) Rule ← RULE – MATCH (State, rules) Action ← RULE – ACTION (rule) Return Action

2.3.2 Model-based Reflex Agent

UQ. Explain Model based Reflex agent with block diagram.
(MU - Q. 2(a), Dec. 19, 10 Marks, Q. 2(c), Dec. 16, 5 Marks, Q. 5(b), May 16, 5 Marks)

(1) A model-based reflex agent needs **memory for storing the precept history**; it uses the percept-history to help to reveal the current unobservable aspects of the environment. An example of this IA class is the **self-screening mobile-vision**, where it is necessary to check the percept history to fully understand how the world is evolving.

(2) Model-based reflex agents are made to deal with **partial accessibility**; they do this by keeping track of the world it can see now. It does this by keeping an internal state that depends on what it has seen before, so it holds information on the unobserved aspects of the current state.

(3) A simple reflex agent selects actions based on the **agent's current perception of the world and not based on past perception**. A model based reflex agent is designed to deal with partial accessibility. They do this by keeping track of the part of the world it can see now.

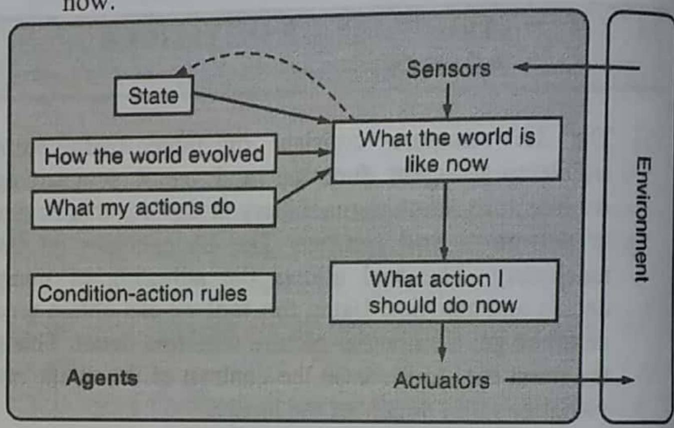


Fig. 2.3.2 : Model-based reflex agent in AI

Function REFLEX-AGENT WITH-STATE (percept) Returns an action. Static : State, a description of the current world state rules, a set condition-action rules. action, the most recent action, initially none. State ← UPDATE – STATE (State, action, percept) Rule ← RULE – MATCH (State, rules) Action ← RULE – ACTION (rule) Return Action

2.3.3 A Goal-based Reflex Agent

UQ. Explain Goal Based with block diagram.
(MU - Q. 2(b), Dec. 18, Q. 2(B), Dec. 17, 10 Marks, Q. 1(d), May 17, 4 Marks)

(1) A goal-based reflex agent has a goal and has a strategy to reach that goal. All actions are taken to reach this goal.

(2) More precisely, from a set of possible actions, it selects the one that improves the progress towards the goal (not necessarily the best one).

(3) A goal-based agent has an agenda. Unlike a simple reflex agent that makes decisions based solely on the current environment, a goal-based agent is capable of thinking beyond the present moment to decide the best actions to take in order to achieve its goal.

(4) Goal-based agents expand the capabilities of the model-based agent by having the **'goal'** information. They choose an action, so that they can achieve the goal.

(5) These agents may have to consider a long sequence of possible actions, before deciding whether the goal is achieved or not.

(6) A goal-based algorithm uses **searching and planning** to act in the most-efficient solution to achieve the goal.

Remark : Conclusion is a statement to Goal-based agent, but is not considered as goal-based agent.

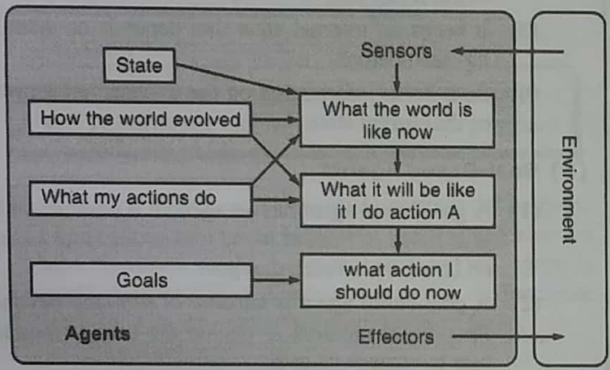


Fig. 2.3.3 : Goal-based agent diagram

2.3.4 An Utility-based Reflex Agent

UQ. Explain Utility based agent with block diagram. **(MU - Q. 2(a), Dec. 19, 10 Marks, Q. 2(c), Dec. 16, 5 Marks, Q. 2(b), Dec. 18, Q. 2(B), Dec. 17, 10 Marks, Q. 1(d), May 17, 4 Marks, Q. 2(a), Dec. 19, 10 Marks, Q. 5(b), May 16, 5 Marks)**

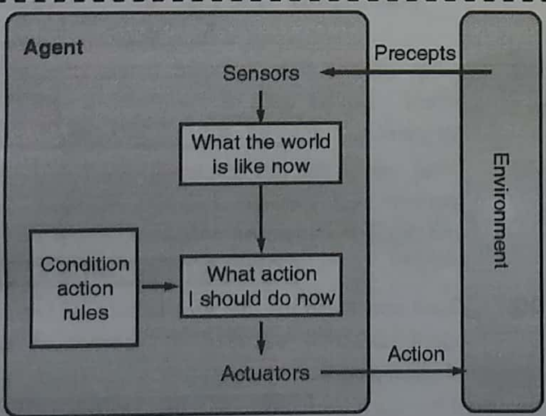


Fig. 2.3.4

(1) An utility-based reflex agent is like the goal-based agent but with measure of **'how much happy'** an action would make it rather than the **goal-based binary feedback [happy, unhappy].**

(2) This kind of agents provides the best solution. An example is the route recommendation system which solves the 'best' route to reach a destination.

(3) The agents which are developed having their end-uses as building blocks are called **utility-based agents**. When there are multiple possible alternatives, then to decide which one is best, utility-based agents are used. They choose actions based on a preference (utility) for each state.

(4) Goal-based agents are important as they are used to expand the capabilities of the model-based agent by having the 'goal' information.

(5) They choose an action, in order that they will achieve the goal. Utility based agent-act based not only goals but also the samples, thanks to achieving the goal.

(6) The utility-based agent is useful when there are multiple possible alternatives, and an agent has to choose in order to perform the best action. The utility function maps each state to a real number to check how efficiently each action achieves the goals.

(7) In artificial intelligence, utility function **assigns values to certain actions** that the AI can take. An AI agent's preferences over all possible outcomes can be captured by a function that maps the outcomes to a utility value, the higher the number, the more that agent likes that outcome.

(8) In Economics, the utility function **measures the welfare or satisfaction of a consumer as a function of the consumption of real goods**, such as food or clothing. Utility function is widely used in rational choice theory to analyze human behavior.

(9) Utility theory bases its beliefs upon individual's preferences. It is a theory postulated in economics to explain behavior of individuals based on premise; people can consistently rank order their choices depending upon their preferences. We can state that individual's preferences are intrinsic.

Module 2

2.3.5 Comparison of Model Based Agent and Utility Based Agent

UQ. Compare of Model based Agent and Utility based Agent. **(MU - Q. 4(b), May 17, 4 Marks)**

Sr. No.	Model based Agent	Utility based Agent.
1.	Goal-based agents are very important as they are used to expand the model-based agent by having the goal information.	Utility based agent-act based not only goals but also the simplest thanks to achieving the goal.
2.	They choose an action in order that they will achieve the goal.	A utility based agent makes decisions based on the maximum utility of its choices.
3.	A model based reflex agent that uses percept history and internal memory to make decisions about the 'model' of the world around it.	It's usefulness (utility) of the agent that makes itself distinct from its counterparts.
4.	Internal memory allows these agents to store some of their navigation history to help understand things about their current environment even when everything that they need to know cannot be directly observed.	A goal-based agent makes decisions based simply on achieving a set goal. Suppose you want to travel from Pune to Mumbai. Mumbai is the goal and the goal based agent will get you there.
5.	Model-based agent uses GPS to understand its location and predict upcoming drivers.	But if you come across a closed road, the **utility-based** agent will analyse other routes to get you there. And it will select the best option for maximum utility. Hence, the utility-based agent is a step above the goal-based agent.
6.	Model-based reflex agents are made to deal with partial accessibility. It does this by keeping an internal state that depends on what it has seen before. So it holds information on the unobserved aspects of the current state.	Utility based agent is more agile and sophisticated since it has some decision making capabilities.

2.3.6 Comparison Model Based Agent with Goal Based Agent

UQ. Compare Model Based Agent with Goal Based Agent. **(MU - Q. 1(a), May 19, 5 Marks)**

(A) Model Based Agents

(1) Model based reflex agents are made to deal with partial accessibility;

(2) They do this by keeping track of the part of the world it can see;

(3) It keeps an internal state that depends on what it has seen before.

(4) So it holds information on the unobserved aspects of the current state.

(B) Goal-Based Agents

(1) A goal-based agent has an agenda. It operates on a goal based in front of it and makes decisions based on how best to reach that goal.

(2) A goal-based agent is capable of thinking beyond the present moment to decide the best actions to take to achieve its goal.

(3) A goal based agent operates as a **search and planning** function.

(4) It targets the goal ahead and finds the right action in order to reach it.

2.4 LEARNING AGENTS

UQ. Explain Learning agent with block diagram. **(MU - Q. 4(c), May 19, 5 Marks, Q. 1(b), May 18, Q. 1(D), Dec. 17, Q. 4(b), May 17, 10 Marks, Q. 5(b), May 16, 5 Marks)**

UQ. What are the basic building blocks of Learning Agent ? Explain each of them with a neat block diagram. **(MU - Q. 6(b), May 16, 10 Marks)**

UQ. What will be the job of each of the components (performance element, Learning element, Critic and problem generator) of learning agent ? **(MU - Q. 1(c), Dec. 18, 5 Marks)**

UQ. Draw and illustrate the Architecture of Learning agent. Describe each of its component w.r.t. Medical diagnosis system. **(MU - Q. 2(a), May 19, 10 Marks)**

(1) A learning agent is an agent **capable of learning from experience.** It has the capability of **automatic acquisition and integration** into the system. Any agent designed and expected to be successful in an uncertain environment is considered to be a learning agent.

(2) A learning agent in AI is the type of agent that can learn from its past experiences or it has learning capabilities. It starts to act with basic knowledge and then is able to act and adapt automatically through learning. The learning algorithm gains **feedback from the critic** on how well the agent is doing and determines how the performance element should be modified it at all to improve the agent.

The Building blocks of learning agent are :

1. Critic	2. Learning Agent
3. Performance element	4. Problem generator

1. Critic : The critic element determines the outcome of the action and gives feedback. The learning element uses feedback from the critic on how the agent is performing and then determines how the performance element should be modified to do better in future.

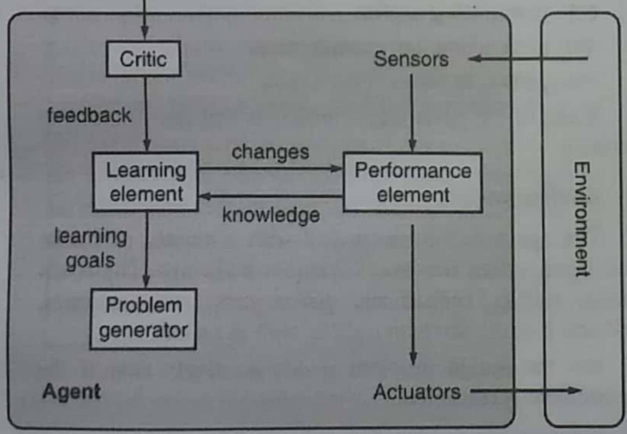


Fig. 2.4.1 : Block Diagram of General Learning Agent

2. Learning Agent : The learning agent takes the feedback from the critic element and figures out how to make the action better next time. Thus, it is responsible for making improvements.

3. Performance element : The performance element is responsible for selecting external actions. The performance element takes in percepts and decides on actions; it is the entire agent.

The performance element chooses the action and later it shifts to a new action based on feedback and suggestions for improvements. Learning is a process of modification of each component of the agent with the available feedback of information, thereby improving the overall performance of the agent.

4. Problem generator : The generator is tasked with developing new experiences for the learning agent to try; this is the piece that helps the agent continue to help.

Thus we observe that agents have a variety of components, and hence there is a great variety among learning methods. In short, we can say: Learning in intelligent agent is a process of modification of each component of the agent with the available feedback of information, thereby improving the overall performance of the agent.

2.4.1 Differentiate between S-R Agent Model and Goal-Based Agent

Simple Reflex Agents

(1) They choose actions only based on the current percept.

(2) They are rational only if a correct decision is made on the basis of current precept.

(3) Their environment is completely observable.

- **Condition-action rule :** It is a rule that maps a state (condition) to an action.
- **Goal based agents :** They choose their actions in order to achieve goals. Goal-based approach is more flexible than reflex agent since the knowledge supporting a decision is explicitly modelled, thereby allowing for modifications.
- **Goal :** It is the description of desirable situations.

2.5 ENVIRONMENT

The natural environment or natural world encompasses all living and non-living things occurring naturally, meaning that they are not artificial.

2.5.1 Definition of Environment

1. The circumstances, objects or conditions by which one is surrounded.
2. The complex of physical, chemical or biotic factors (**such as climate, soil and living things**) that act upon an organism or an ecological community and ultimately determine its form and survival.

Module 2

3. The sum of the total of the elements, factors and conditions in the surroundings which may have an impact on the development, action or survival of an organism or group of organisms.

2.5.2 Three Types of Environment

The three types of environment are as follows :

(I) The physical environment
(II) Social environment
(III) Cultural environment.

(I) Physical Environment

(1) It consists of all components provided by nature and hence can be called as the natural environment. It is also referred to as physical environment, as it pertains to the physical requirements of life.

(2) These physical conditions are not dependent on the existence of humans. It includes natural resourses, the earth's surface, mountains, plains, land, water, deserts, storms, cyclones, volcanoes, oceans, climatic factors, and so on.

(3) It also refers to biological situations such as complexities associated with plants and animals.

(II) Social Environment

(1) Social environment is the immediate environment or surroundings associated with an individual.

(2) Social environment includes all the social practices, government, occupational structure, family, friends, acquaintances, religioius beliefs. It also refers to trends and developments in changes in attitudes, behaviour, and values in society.

(3) It is closely related to population, life-style, culture, tastes, customs and traditions. Social environments allow us to receive feedback, comments, and suggestions in real time concerning our developmental goals and the observations of our behaviours.

(4) Timely feedback is a critical component for personal development and provides for assessing developmental gaps and progress.

(III) Cultural Environment

(1) Cultural environments are environments shaped by human activities, such as cultural landscapes in the countryside, forests, urban areas and cities, fixed archaeological structures on land or water, constructions and built environments from different ages, along with bridges, roads and power lines.

(2) Cultural environment influences the personality traits of people in the community as well as their ideologies.

(3) The factors of cultural environment are language, social norms, religion, ethics, socio-economics, traditions, social regulations, nationalism aethetics, material culture, attitudes, values and social organisation.

Now we consider the types based on the way the environment appears to the agent.

2.5.3 The Task Environment

In task environment, we consider PEAS (Performance, Environment, Actuators, Sensors) descriptions. In designing an agent, the first step must be to specify the task environment.

Let us consider a problem : an automated taxi-driver. We mention the PEAS-description for the taxi's task environment.

1. Performance Measure

The desirable qualities are :

(i) correct destination
(ii) minimising violations of traffic laws,
(iii) minimising disturbances to other drivers,
(iv) maximising safety and passenger comfort,
(v) maximising profits,
(vi) minimising fuel consumption,
(vii) getting to correct destination.

Some of the goals may conflict, in that case tradeoff is required.

2. Environment

The agent will have to deal with a variety of roads; rural lanes, urban free ways, express highways. The roads contain traffic, pedestrians, police cars, stray animals, potholes. It could drive on right as well as on left.

But the design problem is comparatively easy if the environment is restricted.

3. Actuators

The actuators include control over the engine through steering, accelerator and braking. Also output to a display screen or voice synthesizer to communicate with other vehicles and also passengers and also to communicate in rough language, if so required.

4. Sensors

Sensors include to detect distances to other cars and obstacles, to control the vehicle on curves. It will also need engine, fuel and electrical system sensors.

2.5.4 Properties of Task-Environment

UQ. Write short notes on: Properties of Agent Task Environment.

(MU - Q. 6(a), May 19, 5 Marks, Dec. 15, 3 Marks, Q. 5(b), Dec. 15, 5 Marks)

The task-environment in AI is vast. So, we categorise the task-environments in a small number of dimensions. First we enlist the dimensions :

(I) Fully observable versus partially observable
(II) Single agent Vs. Multiagent
(III) Static Vs-Dynamic
(IV) Deterministic Vs. Stochastic
(V) Episodic Vs. Sequential
(VI) Discrete Vs Continuous
(VII) Known Vs Unknown
(VIII) Accessible Vs Inaccessible

(I) Fully observable versus partially observable

(MU - May 19, Dec. 15)

- If the agent sensors describes the environment at each point in detail, then the environment is fully observable. Fully observable environments are convenient because agent need not maintain track of the world.
- An environment may be partially observable because an automated taxi cannot see what other drivers are thinking. A fully observable environment is also termed as accessible environment, while partially observable environment is termed as maccessible environment.

(II) Single agent Vs. Multiagent

(MU - May 19, Dec 15)

- An agent playing on a violine is a single-agent environment while an agent playing a chess is in a two-agent environment.The agent design problems in the multi-agent environment are different from single agent environment.
- Playing tennis against the ball is a single agent environment. The vacuum cleaning environment is single agent environment while chess is a two-agent environment.
- The expert agent where an agent acts like an expert assistant to a user attempting to fulfil some tasks on a computer is multiple-agent environment.

(III) Static Vs-Dynamic

(MU - May 19, Dec 15)

- If the environment can change itself while an agent is deliberating then such environment is called a dynamic environment, else it is called a static environment.
- Static environment are easy to deal because an agent does not need to continue looking at the world while deciding for an action.
- But for dynamic environment, agents need to keep looking at the world at each action.
- Taxi driving is an example of a dynamic environment whereas playing on Piano is an example of a static environment. If the environment does not change with the passage of time, but the agent performance changes by time, then it is semi-dynamic.

Playing card-games is a perfect example of partially observable environment where a player is not aware of the card in the opponent's hand.

(IV) Deterministic Vs. Stochastic

(MU - May 19, Dec. 15)

- If an agent's current state and selected action can completely determine the next state of the environment, then such environment is called a deterministic environment.
- A stochastic environment is random in nature and cannot be determined completely by an agent.
- In a deterministic, fully observable environment, agent does not need to worry about uncertainty.

For a given current state and action executed by agent, the next state or outcome cannot be exactly determined, e.g., if agent kicks the ball in a particular direction, then the ball may or may not be stopped by other players.

Chess : the board is fully observable, so are the opponent's moves.

Driving : the environment is partially observable because what is around the corner is not known.

(V) Episodic Vs. Sequential

(MU - May 19, Dec. 15)

- In an episodic environment, there is a series of one-shot actions, and only the current percept is required for the action. But in sequential environment, an agent requires memory of past actions to determine the next best actions.
- Episodic is an environment where each state is independent of each other. The action on a state has nothing to do with the next state.

- The sequential environment is an environment where the next state is dependent on the current action. So agent current action can change all of the future states of the environment.
- An AI that looks at radiology images to determine if there is a sickness, is an example of an episodic environment. One image has nothing to do with the next state. Chess with a clock is fully sequential.

(VI) Discrete Vs Continuous

(MU - May 19, Dec. 15)

- If in an environment there are a finite number of percepts and actions that can be performed within it, then such an environment is called a discrete environment else it is called continuous environment.
- A chess game comes under discrete environment as there is a finite number of moves that can be performed.
- A self-driving car is an example of a continuous environment. Input from digital cameras is discrete.

(VII) Known Vs Unknown

(MU - May 19, Dec. 15)

- Known and unknown are not actually a feature of an environment, but it is an agent's state of knowledge to perform an action.
- In a known environment, the results for all actions are known to the agent.
- While in unknown environment, agent needs to learn how it works in order to perform an action.
- It is quite possible that a known environment to be partially observable

  For example, in Rummy card game I know the rules of the game but unable to see the card that is not turned over. An unknown environment is fully observable in a video game, the entire game is visible on the screen, but I do not know what the buttons **do till I try them.**

(VIII) Accessible Vs Inaccessible

(MU - May 19, Dec. 15)

- If an agent can obtain complete and accurate information about the state's environment then such an environment is called an Accessible environment else it is called inaccessible.
- An empty room whose state can be defined by its temperature is an example of an accessible environment.

Now ; we enlist the properties of a number of familiar environments :

2.5.5 Examples of Task Environments and their Characteristics

Task environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Chess with a clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continuous
Image analysis	Fully	Single	Deterministic	Episodic	Semi-dynamic	Continuous
Chess without clock	Fully	Multi	Stochastic	Sequential	Static	Discrete

2.6 SOLVING PROBLEMS BY SEARCHING/STATE SPACE SEARCH

UQ. Explain any AI problem with suitable example. (MU - Q. 1(d), Dec. 18, 4 Marks)

1. Defining problem and search

(i) Problem (or problem space) refers to the entire range of components that exist in the process of finding a solution to a problem.

(ii) It is complete set of states including start and goal states, where the answer of the problem is to be searched.

(iii) Problem : It is a question to be solved.

(iv) A problem has four components : initial state, goal test, set of actions, path cost.

(v) A set of rules which describe the actions available.

If we are solving a problem, we are usually looking for some solution which will be the best among others. The space of all feasible solutions (i.e., the set of solutions among which the desired solution exists) is called search (or search space or state space). Each point in the search space represents one possible solution.

2. Define State and State Space

(i) A state space is the set of all possible configurations of a system.

(ii) It is a useful abstraction for reasoning about the behaviour of a given system.

(iii) It is widely used in the fields of artificial intelligence and game theory.

(iv) In the theory of dynamical systems, a discrete system defined by a function **f**, the state space of the system can be modelled as a directed graph where each possible state of a dynamical system is represented by a vertex, and there is a directed edge from a to b if and only if **f(a) = b**. This is known as a state diagram.

(v) For a continuous dynamical system defined by a function f, the state space of the system is the image of f.

(vi) State spaces are useful in computer science as a simple model of machines.

(vii) Formally, a state space can be defined as tuple [N, A, S, G] where, N is a set of states

A is a set of arcs connecting the states

S is a nonempty subset of N that contains start stater.

G is a non-empty subset of N that contains the goal states.

State spaces can be either infinite or finite, and discrete or continuous.

3. Defining the problem as state space search

(i) State space search is a process used in the field of computer science, including artificial intelligence (AI).

(ii) In which successive 'configurations' or 'states' of an instance are considered, with the intention of finding a goal state with the desired property.

(iii) The set of states forms a graph where two states are connected it one state can be transformed to another state by the given operation an states.

(iv) State space search differs from traditional computer science search methods because the state space is implicit.

(v) The typical state space graph is much too large to generate.

(vi) Instead nodes are generated as they are explored, and then discarded.

(vii) A solution may consist of the goal state itself, or of a path from some initial state to the goal state.

(viii) A state space problem

P = (S, A, S, T) consists of a set of states S, an initial state $s \in S$, a set of goal states $T \subseteq S$, and a finite set of actions

A : $\{a_1, a_2, ..., a_{n3}\}$ where each

$a_i : S \rightarrow S$ transforms a state into another state.

2.7 PROBLEM SOLVING AGENTS

1. The problem - solving agent performs by defining problems and several solutions. Problem solving agent employees a number of techniques such as a tree, heuristic algorithms to solve a problem.
2. The reflex agent of AI directly maps states into action.

   When the state of mapping is too large, then the problem gets dissolved and sent to a problem - solving domain. It breaks the large stored problem into smaller storage area and resolves one by one. The final action will be the desired outcome.
3. Intelligent agent organises finite number of steps to formulate a target or goal which require some action to achieve the goal.
4. Problem formulation is one of the most important steps of problem-solving. Agent has to decide what action to be taken to achieve the goal.

   Components to formulate the associated problem :

   (i) **Initial state :** In this state, new methods also initialise problem solving by a specific class.

   (ii) **Action :** Here the agent performs all possible actions with a specific class taken from the initial state

   (iii) **Transition :** Here the agent integrates the actual action by the previous action stage and collets the final stage to forward to the next stage.

   (iv) **Goal test :** When the goal is achieved, action steps and then forwards into the next stage to determine the cost to achieve the goal.

   (v) **Path costing :** Here the agent calculates all hardware, software and human working cost.

Thus, a problem - solving agent is a goal - driven agent and focuses on satisfying the goal.

2.7.1 Simple Problem-Solving Agent

Function SIMPLE-PROBLEM-SOLVING- AGENT (percept) returns an action **Persistent:**

seq. an action sequence, initially empty,

state, some description of the current world state,

goal; a goal initially null,

problem, a problem formulation,

state ← UPDATE-STATE(state percept)

If seq. is empty then

goal ← FORMULATE-GOAL(State)

problem ← FORMULATE-PROBLEM (State, goal)

seq. ← SEARCH(problem}

If seq. = failure then return a null action action

← FIRST(seq.)

seq. ← REST(seq.) return action

Method

(i) Formulate a problem and a goal.

(ii) Formulate a sequence of actions and execute the sequence of action one - by - one.

(iii) When this is complete, form another goal and proceed.

(iv) Agent uses search-procedure to arrive at a goal.

(v) Agent then uses the solution to guide its actions. Then agent formulates a **new goal**.

Consider an example of a car from Pune to Surat (Fig. 2.7.1)

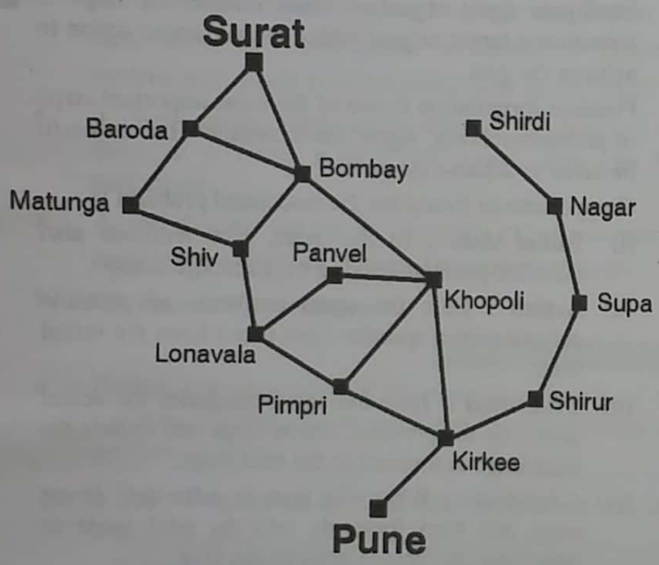


Fig. 2.7.1

2.7.2 Well - Defined Problems and Solutions

Problem - solving consists of using generic or ad-hoc methods in an orderly manner.

Problems can be classified into two different types : ill-defined and well-defined.

1. **Ill defined problems** are those that do not have clear goals, solution paths, or expected Solutions.
2. **Well - defined problems** have specific goals, clearly defined solution paths, and clear expected solutions.

These problems also allow for more initial planning than ill-defined problems.

2.7.2(A) A Problem can be Defined by Five Components

(i) The initial state

It is the state where the agent starts in For example, the initial state for agent is Pune, can be written as In (Pune)

(ii) Action

A description of the possible actions available to the agent.

Given a particular state p, ACTIONS (p) returns the set of actions that can be executed in p. For example, from the state (Pune), the actions could be

{Go (Kirkee), Go (Lonavala), Go (Panvel)}

(iii) Transition model

The formal name for the description of each action is 'transition model'. It is specified by Result (p, a), that results the state from doing action a in p.

We use the term successor to refer to any state reachable from a given state by a single action.

For example, Result {In (Pune), Go (Kirkee)} = In (Kirkee). The initial state, actions and transition model define the **state space** of the problem. It is the set of all states reachable from the initial state by any sequence of actions.

The state space forms a **directed network** or **graph** in which the **nodes** are **states** and **links** between **nodes** are **actions**. A path in the **state space** is a sequence of states connected by a sequence of actions.

(iv) The goal test

It determines whether a given state is a goal-test. There may be an explicit set of possible goal - states, and the test checks whether the given state is one of them.

Sometimes the goal is specified by an abstract property. For example, in chess, the goal is to reach a state called "**checkmate**", where the opponent's king is under attack and cannot escape.

(v) Path cost

It is a function which assigns numeric value or cost to each path. For the agent going to Surat, time is the essence, so the cost of path may be its length in kilometres. Here the cost of the path is the sum of the costs of individual actions along the path. The **step cost** of taking action a in state p to reach state p′ is denoted by c(p, a, p′). The step costs are non-negative.

2.7.3 Formulating Problems

(i) In the previous article, we made a formulation of the problem of getting to Surat in terms of **initial state, action, transition model, goal test** and **path cost**. This formulation makes sense, but it is still an abstract mathematical description and far away from the reality. For example, consider the simple state description, In (Pune).

Where the state of the world includes so many things, the travelling companions, the scenery, condition of the road, the weather and so on.

All these considerations we did not take into our discussion, because they are irrelevant to the problem of reaching to Surat. The process of deleting these details from the representation is called **'abstraction'**

(ii) Now, we take note of abstracting the actions themselves. There are many that we omit altogether, looking out of the window, turning on the radio slowing down for the police-officers, and so on. Also besides changing the location of the vehicle and its occupants it takes up time, consumes fuel, generates pollution and so on.

Our formulation takes into account only the change in location.

(iii) Now we consider a solution of the abstract problem : In the previously mentioned example, the path from Pune to Kirkee to Lonavala to Panvel to Bombay to Surat.

This abstract solution corresponds, to a large number of details. For example, we can drive from Lonavala to Panel, windows open and the shut the windows for the rest of the trip.

Thus the good abstraction is that which involves removing as much details as possible, assuming that abstract actions are easy to carry out.

2.8 EXAMPLE PROBLEMS

We enlist some of the best known problems, distinguishing between **toy** and **real-world** problems.

1. Toy Problem

A toy problem is intended to illustrate various problem-solving methods. Here we mention exact description of the problem and hence is usable to compare the performance of algorithm.

2. Real-world problem

The solution of these problems are useful and people actually care about these problems.

These problems cannot have a unique solution but we can give the general flavour of their formulations.

2.8.1 Toy-problem Example

(a) Vacuum-cleaner World

- This world-problem is very simple so that we can invent many variations in it.
- This world has just two locations : squares A and B. The vacuum agent perceives which square it is in and whether there is dirt in the square. It can move left, move right, suck up the dirt, or do nothing.
- One very simple agent function is : if the current square is dirty, then suck; otherwise, move to the other square.

Module 2

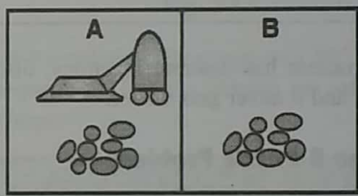


Fig. 2.8.1

We formulate the problem as follows :

1. **States :** The state is determined by the agent location and the dirt locations. The agent is in one of the two locations. And each of them may or may not contain dirt.

   Thus, there are $2 \cdot 2^2 = 8$ possible world states.

   For n locations, it will have $n \cdot 2^n$ states
2. **Initial state :** Any state can be taken as the initial state
3. **Actions :** Each state has three actions : **Left, Right, Suck**
4. **Transition Model :** The actions have their expected effects, except that moving left in the leftmost square, moving right in the rightmost square and sucking in a clean square have no effect.
5. **Goal Test :** It checks whether all squares are clean.
6. **Path cost :** Each step costs 1, so the path cost is the number of steps in the path.

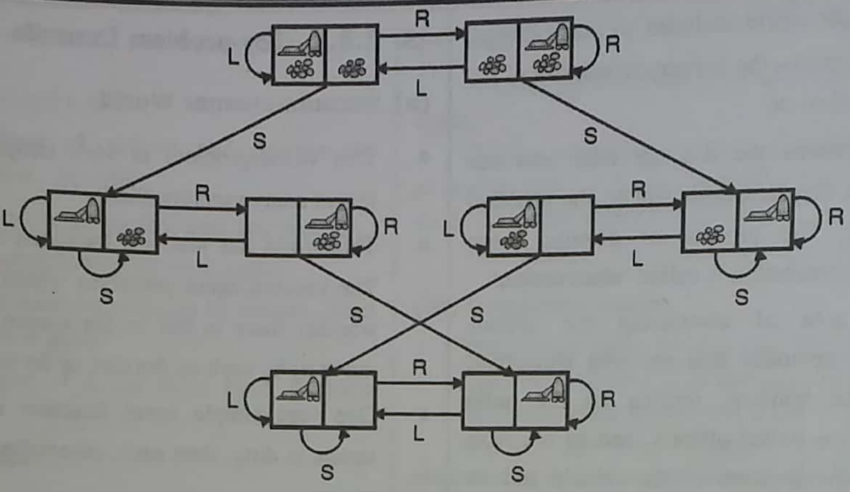


Fig. 2.8.2

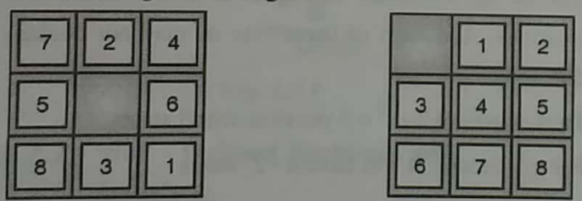
This toy problem has discrete locations, discrete dirt, reliable cleaning and it never gets dirtier.

2.8.2 The 8 Puzzle Problem

It consists of a 3 × 3 board with eight numbered tiles and a blank space.

A tile adjacent to the blank space can slide into the space.

The object is to reach a specified goal state, such as shown on the right of the figure.



(a) Start state (b) Goal state

Fig. 2.8.3

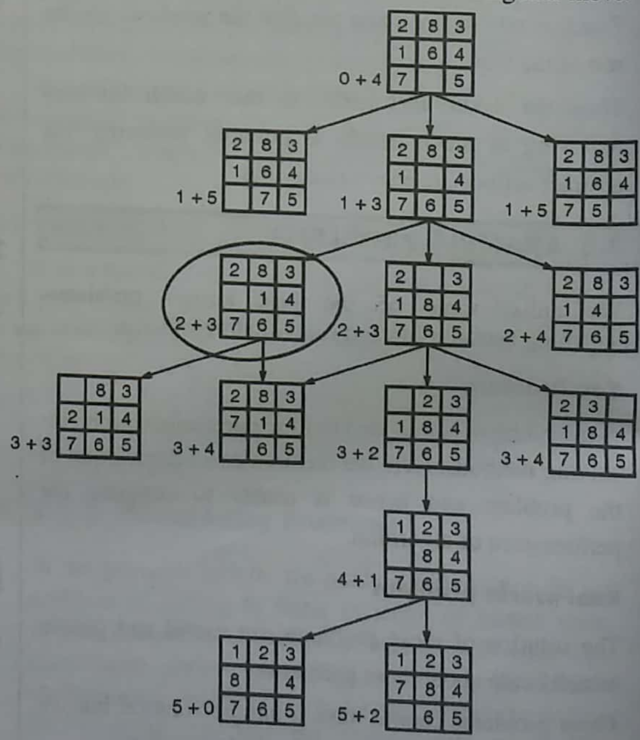
We form the standard **formulations.**

1. **States :** The location of each of the eight tiles and the blank in one of the nine squares is described by a state.
2. **Initial state :** Any state can be taken as the initial state.
3. **Actions :** The actions as movements of the blank space Left, Right, Up or Down are defined by the simplest formulation.
4. **Transition Model :** Given a state and action, this returns the resulting state. For example, if we apply **right** to the start state then the resulting state has 6 and the blank state is switched.
5. **Goal Test :** The test checks whether the state matches the goal configuration.
6. **Path cost :** Each step cost 1, so the path cost is the number of steps in the path.

Here we are interested in the description of the rules of the puzzle, avoiding all the details of physical manipulation.

The 8th puzzle problem

The puzzle can be solved by moving the tiles one by one in the single empty space and achieving the goal state. One can take only one step at a time and no diagonal move.



2.8.3 8-Queens Problem

The goal of 8-Queens Problem is to place eight queens on a chessboard such that no queen attacks any other. A queen can attack any piece in the same row, column or diagonal.

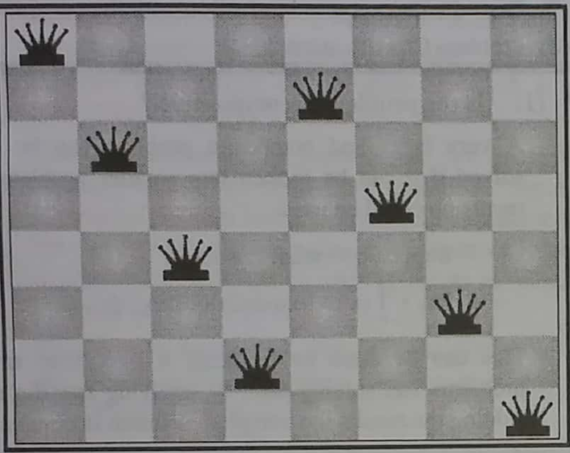


Fig. 2.8.4

In Fig. 2.8.4 the queen in the rightmost column is attacked by the queen at the top left.
(This attempted solution is a failure.)
Here there are two main kinds of formulations.

(1) An incremental formulations

This involves operators that **augment** -the state description, beginning with an empty state ; for this problem, it means that each action adds a queen to the state.

(2) A complete state formulation

This starts with all 8 queens on the board and moves them around. In either case, the path cost is of no interest because only the final state counts. We form the first incremental formulation as follows :

(i) **States :** Any arrangement of 0 to 8 queens on the board is a state.

(ii) **Initial state :** No queens on the board.

(iii) **Actions :** Add a queen to any empty square.

(iv) **Transition model :** Returns the board with a queen added to the specified square.

(v) **Goal test :** 8 Queens are on the board, no one is attacked.

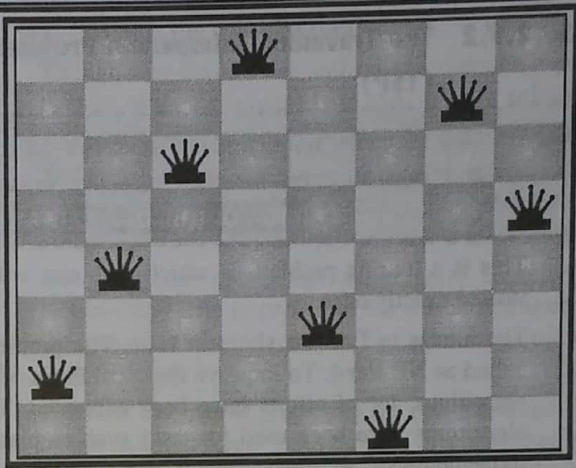


Fig. 2.8.5

In this formulation, there are $64 \cdot 63 \quad 57 = 1.8 \times 10^{14}$ possible sequences. Here the Queens are safe; no queen can attack the other.

2.9 EXAMPLES OF REAL WORLD PROBLEM

2.9.1 Route-Finding Problem

Route-finding problem is defined in terms of specified locations and transitions along links between them. In a variety of applications, route finding algorithms are used.

It is used in Web-sites and in car systems that provide driving directions. Also in routing video-streams in computer networks, military operations planning etc.

We consider the airline travel problem that can be solved by a travel - planning web sites.

(i) **States :** Each state includes a location i.e. an airport and the current time. The state has to record extra information such as domestic or international flight, their fare bases etc.

(ii) **Initial state :** This depends upon users's query.

(iii) **Actions :** Take any flight from the current location, leaving after the current time, leaving enough time for within airport transfer if needed.

(iv) **Transition Model :** Taking a flight will have the flight's destination as the current location and the flights arrival time as the current time.

(v) **Goal Test :** Final destination mentioned by the users.

(vi) **Path cost :** This depends upon monetary cost, waiting time, flight time, customs and immigration procedures, seat quality, time of day, type of airplane and so on.

2.9.2 The Travelling Salesperson Problem (TSP)

UQ. Give the initial state, goal test, successor function, and cost function for the Travelling salesman problem **(MU - Q. 1(b), Dec. 17, 5 Marks)**

- This is a touring problem in which each city is to be visited exactly once.
- The aim is to find the **shortest tour.** The problem is called as **NP-hard**. To improve the capabilities of TSP algorithms, lot of efforts have been expended. These algorithms have been used for tasks such as planning movements of automatic circuit board drills and of stocking machines on shop floors.

2.9.3 A VLSI Layout

This problem requires vast number of components and connections on a chip to minimise area, minimise stray capacitance, minimise circuit delays and maximising manufacturing yield.

The layout problem is basically logical design phase. It is usually split into two parts. They are (i) cell layout and (ii) channel routing.

(i) **Cell Layout :** In cell layout, the primitive components of the circuit are grouped into cells, each of which performs some recognised function. Each cell has a fixed footprint (size and shape) and a certain number of connections to each of the other cells. Cells are placed on the chip so that they do not overlap and connecting wires can be placed between the cells.

(ii) **Channel Routing :** Channel routing finds a specific route for each wire through the gaps between the cells. These problems even though they are complex but worth solving.

2.10 PROBLEM CHARACTERISTICS OF IA

GQ. Discuss with examples : AI Problem Characteristic.

GQ. Explain how AI techniques improve real-world problem solving.

It includes a variety of techniques. In order to choose an appropriate method, it is necessary to analyze the problem with respect to the following considerations :

(1) Is the problem decomposable ?
(2) Can solution steps be ignored or undone?
(3) Is the Universal Predictable ?
(4) Is good solution absolute or relative ?
(5) The knowledge base consistent ?
(6) What is the role of Knowledge?
(7) Does the task require interaction with the person.
(8) Problem Classification

(1) Is the problem decomposable ?

- A very large and composite problem can be easily solved if it can be broken into smaller problems and recursion could be used.

Suppose we want to solve.

Ex. : $\int (x^2 + 3x+\sin 2x \cos 2x)\, dx$

- This can be done by breaking it into three smaller problems and solving each by applying specific rules. Adding the results the complete solution is obtained.

(2) Can solution steps be ignored or undone?

- Problem fall under three classes ignorable, recoverable and irrecoverable. This classification is with reference to the steps of the solution to a problem. Consider thermo proving. We may later find that it is of no help. We can still proceed further, since nothing is lost by this redundant step. This is an example of ignorable solutions steps.
- Now consider the 8 puzzle problem tray and arranged in specified order. While moving from the start state towards goal state, we may make some stupid move and consider theorem proving.
- We may proceed by first proving lemma. But we may backtrack and undo the unwanted move. This only involves additional steps and the solution steps are recoverable.
- Lastly consider the game of chess. If a wrong move is made, it can neither be ignored nor be recovered. The thing to do is to make the best use of current situation and proceed. This is an example of an irrecoverable solution steps.
  1. Ignorable problems Ex:- theorem proving
     In which solution steps can be ignored.
  2. Recoverable problems Ex:- 8 puzzle
     In which solution steps can be undone
  3. Irrecoverable problems Ex:- Chess
     In which solution steps can't be undone
- A knowledge of these will help in determining the control structure.

(3) Is the Universal Predictable ?

- Problems can be classified into those with certain outcome (eight puzzle and water jug problems) and those with uncertain outcome (playing cards) in certain – outcome problems, planning could be done to generate a sequence of operators that guarantees to a lead to a solution.
- Planning helps to avoid unwanted solution steps. For uncertain outcome problems, planning can at best generate a sequence of operators that has a good probability of leading to a solution.
- The uncertain outcome problems do not guarantee a solution and it is often very expensive since the number of solution and it is often very expensive since the number of solution paths to be explored increases exponentially with the number of points at which the outcome cannot be predicted.
- Thus one of the hardest types of problems to solve is the irrecoverable, uncertain – outcome problems (Ex:- Playing cards).

(4) Is good solution absolute or relative ? (Is the solution a state or a path ?)

- There are two categories of problems. In one, like the water jug and 8 puzzle problems, we are satisfied with the solution, unmindful of the solution path taken, whereas in the other category not just any solution is acceptable.
- We want the best, like that of traveling salesman problem, where it is the shortest path. In any – path problems, by heuristic methods we obtain a solution and we do not explore alternatives.
- For the best-path problems all possible paths are explored using an exhaustive search until the best path is obtained.

(5) The knowledge base consistent ?

- In some problems the knowledge base is consistent and in some it is not. For example, consider the case when a Boolean expression is evaluated.
- The knowledge base now contains theorems and laws of Boolean Algebra which are always true. On the contrary consider a knowledge base that contains facts about production and cost.
- These keep varying with time. Hence many reasoning schemes that work well in consistent domains are not appropriate in inconsistent domains. **Ex.** Boolean expression evaluation.

(6) What is the role of Knowledge ?

- Though one could have unlimited computing power, the size of the knowledge base available for solving the problem does matter in arriving at a good solution.
- Take for example the game of playing chess, just the rules for determining legal moves and some simple control mechanism is sufficient to arrive at a solution.
- But additional knowledge about good strategy and tactics could help to constrain the search and speed up the execution of the program. The solution would then be realistic.
- Consider the case of predicting the political trend. This would require an enormous amount of knowledge even to be able to recognize a solution , leave alone the best. **Ex :** 1. Playing chess 2. News paper understanding.

(7) Does the task require interaction with the person

The problems can again be categorized under two heads.

(1) **Solitary** in which the computer will be given a problem description and will produce an answer, with no intermediate communication and with the demand for an explanation of the reasoning process. Simple theorem proving falls under this category. given the basic rules and laws, the theorem could be proved, if one exists.

Ex. : theorem proving (give basic rules and laws to computer)

(2) **Conversational,** in which there will be intermediate communication between a person and the computer, wither to provide additional assistance to the computer or to provide additional informed information to the user, or both problems such as medical diagnosis fall under this category, where people will be unwilling to accept the verdict of the program, if they cannot follow its reasoning. **Ex :** Problems such as medical diagnosis.

(8) Problem Classification

Actual problems are examined from the point of view, the task here is examine an input and decide which of a set of known classes. **Ex. :** Problems such as medical diagnosis , engineering design.

In Short AI has following Characteristics

(i) They have some level of autonomy that allows them to perform certain tasks on their own.

(ii) They have a learning ability that enables them to learn even as tasks are carried out.

(iii) They can interact with other entities such as agents, humans and system.

Chapter Ends...

□□□