1. Search Strategies

- **Breadth-First Search: What is Breadth-First Search? Describe this search strategy and discuss its completeness and complexity.**
- **Uninformed Search: What are non-informed search strategies? Describe the strategies that have been presented during the course.**
- **Iterative Deepening: What are the main features of iterative deepening?**

---

2. Planning

- **Deductive Planning: What are the main approaches of deductive planning? Explain the main differences.**
- **Conditional Planning: What is conditional planning and which are its main features and/or limitations?**
- **Hierarchical Planning: What is hierarchical planning and explain the method presented during the course.**
- **Modal Truth Criterion: What is modal truth criterion and why it has been defined?**

---

3. Local Search & Swarm Intelligence

- **Local Search: What are the main features of a local search algorithm?**
- **Metaheuristics: What are metaheuristics? Describe the main algorithms that have been presented during the course.**
- **Swarm Intelligence: What are the main features of a swarm intelligence algorithm?**
- **Particle Swarm Optimization (PSO): What is Particle Swarm Optimization and which are its main features?**
- **Ant Colony Optimization: What is ant colony optimization?**

---

4. Constraint Satisfaction

- **Arc-Consistency: What is arc-consistency? Describe the algorithm to achieve it. Explain the properties of values that are removed from constraints and of values that are left in the domains.**

Observation: The questions on Swarm Intelligence, Conditional Planning, Modal Truth Criterion, and Deductive Planning appeared most frequently across the exams.

Answers

1. Search Strategies

- **Breadth-First Search: What is Breadth-First Search? Describe this search strategy and discuss its completeness and complexity.**

  Bfs is a type of uniformed search strategy, that use node as states and arc as action to pass from a state to another state. At each step explore-expand the first node in a queue called fringe. Every new node founded in this way is placed at the end of the queue. This process guarantees to find an optimal solution. The search time is O(b^d) where the b is the branching factor and d is the depth of the solution. This algorithm is especially heavy for the memory because it store every node. infact it's memory requirement is O(b^d).

- **Uninformed Search: What are non-informed search strategies? Describe the strategies that have been presented during the course.**

  The uniformed search strategies seen during the course are: BFS, DFS, Uniformed cost Search, Limited DS, Iterative deepening. All these strategies use node as state and arch as actions. DFS for the Fringe use a LIFO stack. So the last found node is the first to be explored. That makes DFS prone to loop, but memory efficient, Limited DS and iterative deepening, are DFS but with limits on the max deep reachable. BFS and UCS use a FIFO queue for the Fringe. That means that the fist found node is the first to be explored. That makes both BSF and UCS Complete and Optimal. The difference between the two is that UCS care of the action cost so the fringe queue has a ascending cost order.

  Non of these strategies has any clue to how to choose the arch, they just follow their fringe data structure.

- **Iterative Deepening: What are the main features of iterative deepening?**

  It is an Uniformed search strategy. It combine the best features of BFS and DFS. Infact it is both complete and optimal, and at the same time use the memory usage b*d of DFS search. To reach this result the Iterative Deepening apply the Limited Deep Search multiple time, and if it does not find the solution incrise the depth limith. In this way explore each level like the BFS but use the DFS memory usage.

---

2. Planning

- **Deductive Planning: What are the main approaches of deductive planning? Explain the main differences.**

  Green formulation and Kovalshi formulation bot use logic resolution for finding the best action to reach the goal. the main different between the two is how the handle the frame problem. In the Green formulation for each axioms will be created a frame axiom, while in the kovalski formulation thanks to Hold() we can define only one frame Axiom for an action

- **Conditional Planning: What is conditional planning and which are its main features and/or limitations?**

  In the real world we don't always had the value of each property. So some value are Unknown, and we get those values with the sensing-action. Conditional Planning is based on the theory that there should be a plan for every possible sensing action. This make the list of possible sub plan exponential in the number of sensing actions. This require a lot of memory, and also is not always possible to know every possible case

- **Hierarchical Planning: What is hierarchical planning and explain the method presented during the course.**

  Hierarchical Planning is based on two different methodologies for handling complex task. In the first method POP-like we use composition to aggregate complex task in Marco-action that can be directly executed, in this way are easier to insert them in the plan. And then use decomposition to define the marco-action with atomic-action. The second method is strips-like, and it is based on the criticality value. Basically the planner try to plan the action in order of their criticality value.

- **Modal Truth Criterion: What is modal truth criterion and why it has been defined?**

  Modal Truth Criterion is a construction process that guarantees palnner completeness. It define 5 rules:

  - Establishment: this rule define three means to reach the gole:
    - Add a new action in the plan
    - Add a new order constraint in the plan
    - Assign a variable
  - Promotion: move the threatening action before the first of the causal link
  - Demotion: mode the threatening action after the second of the causal link
  - White Knight: insert a new action to solve a cross threath
  - Separation: Add a constraint to a variable. This is useful in case we don't want that a variable unify before some other values are instantiated

---

3. Local Search & Swarm Intelligence

- **Local Search: What are the main features of a local search algorithm?**

  The Main characteristic of local search is that instead of reaching a solution from 0, the algorithm start from a solution S and try to permutate-mutate S with the goal of reaching a better solution. So the Local search define things like the Neighborhood of a solution, that are the solutions that can be reach from our current solution. This process can be applied only to problem where it is not important to explain how to reach the solution but need only the final answare. Another characteristic of Local search is that often the solution get stack in local maximum, so we need metha-euristic (Like tabu-list,simulated annealing, iterative search, Population based) in order to skip the local maximum and reach the global maximum

- **Metaheuristics: What are metaheuristics? Describe the main algorithms that have been presented during the course.**

  Are a series of techniques to improve Local search Algorithms not getting stuck at local minimum/maximum:

  -Simulated Annealing: define the probability of moving towards a solution that is worse than our current solution

  -Tabu List: put all the already visited Solution in a List, in this way we can change the neighborhood of a node and try to move into different directions

-Iterated Local search: after finding a solution the algorithm permutate it and start another search iteration. By changing the starting point there is more possibilities of finding the global maximum.

-Population based metauristic: Choose from a list of Solution (Population) the best solution (fittest Parent) and create new solution by mixing them (Offspring creation). The offpring are created by using the crossover of the parents and by instrodusing a random mutation. In the last steps the offspring are inserted into the population or replace the whole population

- **Swarm Intelligence: What are the main features of a swarm intelligence algorithm?**

  the main features of swarm intelligence algorithms are:
  - There is a set of simple individuals with limited capabilities
  - The individual don't have the global view of the problem, individuals try to reach a goal-solution, so individuals performe an semi-indipendent local search.
  - Local comunication: direction or indirect(strigmergic)
  - Distributed Computation: no centralized coordination of indivisuals
  - Robustness
  - adaptability

- **Particle Swarm Optimization (PSO): What is Particle Swarm Optimization and which are its main features?**

  Particle Swarm Optimization, is a type of swarm intelligence and try to imitate bird flocks. Every bird in a flock follow these rule: follow neighbours, stay in the flock and avoid collision. So every particale in PSO knows the Best solution it has found so fare, the global best solution found by the flock, and the particale as also a Velocity that is the direction of his local search. Each individual has is velocity influenced by the Flock best Solution, but it has also some grade of independence into try to find another path

- **Ant Colony Optimization: What is ant colony optimization?**

  Is a type of Swarm Intelligence algorithm. ACO try to imitate the Ants search food, but in our case food is a solution and the ants are agents performing local search. The main feature of this algorithm is that each agents/ant when it move from a solution to another it leaves behind a trace of pheromone. So when an agent must choose between different path his chooise will be influenced both by and euristic,

and by the pheromone level. With the time feromone evaporates (at little bit every iteration). So, in the end only the best trails will remain used by the ants

---

4. Constraint Satisfaction

Arc-Consistency: What is arc-consistency? Describe the algorithm to achieve it. Explain the properties of values that are removed from constraints and of values that are left in the domains

Arc consistency is based on the idea the condition between two variables should be always consistent like arc in a graph. The algorithm is based on an iteration on each arch and a check on the domain of the two Node(variable) connected to the arch. If in the value Di of the node Xi has not a single value supporting the arc-condition in the Domain of the node Xj Di is removed from the domain of Xi. If a value is removed it is always a wrong solution to the problem. But the remaining value can be both good or wrong solution we can’t be sire