1) What is hierarchical planning and explain the method presented during the course.

Hierarchical planners are search algorithms that manage the creation of complex

plans at different levels of abstraction, by considering the simplest details only

after finding a solution for the most difficult ones.

1) STRIPS-Like:

Hierarchical planner who enhances the Strips definition of actions with a criticality

value (proportional to the complexity of its achievement) to each precondition.

The planning algorithm proceeds at different levels of abstraction spaces. At each

level, lower level preconditions are ignored.

2) Partial-Order-Like:

Contains two type of operators. Atomic and Macro.

Atomics are elementary operators, executable by an executor. Macro operators in turn represent a set of elementary actions and should be decomposed before the execution (precompiled or from plan).

At every step we should choose between

1) Reach an open goal with an operator (including macro operators)

2) Expand a macro step of the plan

2) What is Particle Swarm Optimization and which are its main features?

Particle Swarm Optimization PSO (Kennedy, Eberhart 1995)

– Algorithm based on the observation of bird flocks or fish shouls. Stigmergy as communication and imitation of neighborhoods.

- The research activity starts from the analysis of interaction mechanisms between individuals that compose the swarm
- Particularly interesting when the whole swarm has a common goal such as food search.
- The observation of rules that guide the bird flock moves show that each individual entity has three driving trends

– Follow neighbours

– Stay in the flock

– Avoid collisions

• With these rules it is possible to describe and model the collective move of a flock with NO COMMON OBJECTIVE. PSO adds a common objective: food search

• With a common objective, a single individuals that finds a food source has two alternatives:

– Move away from the group to reach the food (individualistic choice)

– Stay in the group (social choice)

If more than one individual entity moves toward the food other flock members do the same. Gradually the whole group changes direction toward promising areas. The information propagates to all members.

• With PSO we can solve optimization problems with the following analogy:

– individuals: tentative configurations that move and sample the solution Ndimensional space.

– Social interaction: each individual agent takes advantage from other searches moving toward promising regions (best solution globally found)

• Search strategy can be found as a balance between exploration and exploitation:

– exploration: individual agents that search for a solution

– Exploitation: social behaviour which is the exploitation of other individual successful behaviour

3) What are the main features of a swarm intelligence algorithm?

Some features of swarm intelligence systems are the following:

• These systems are usually composed of simple individuals with limited capacities.

• The individuals are not aware of the environment in which they live in.

• There exist local communication patterns between individuals. An example of this is stigmergy, a form of indirect communication, in which an individual modifies the environment and the others. react to this modification.

• There exists distributed computation, i.e. there exist no centralized coordination of individuals activities.

• These systems are robust and adaptive.

4) What is conditional planning and which are its main features?

A conditional planner is a search algorithm that generates various alternative plans for each source of uncertainty of the plan.

Main features??

A conditional plan it is therefore constituted by:

➢causal actions

➢Sensing actions for retrieving unknown information

➢Several alternative partial plans of which only one will be

executed depending on the results of the observations.

5) What is modal truth criterion and why it has been defined?

The Modal Truth Criterion is a construction process that guarantees planner Completeness. The MTC provides five plan refinement methods (one for the open goal achievement and 4 for threat protection) that ensure the completeness of the planner.

(Promotion and demotion alone are not enough to ensure the completeness of the planner. A planner is complete if it always finds a solution if a solution exists)

6) What are the main approaches of deductive planning. Explain the main differences.

Deductive planning uses logics for representing states, goals and actions and generates a plan as a theorem proof.

Green and Kowalsky formulations shall we write Differences between them??

7) What are metaheuristics? Describe the main algorithms that have been presented during the course.

A metaheuristic is a high-level procedure used in optimization problems used to find a sufficiently good solution for a problem.

During the course we have distinguished two metaheuristic algo:

1. Population based metaheuristic

Genetic algorithm

It is based on natural selection and provides a good approximation if not the actual solution to an optimization problem. It needs few requirements:

1. A fitness function, to evaluate the goodness of a solution
2. A way to generate a population of solutions of a given problem (**genotypes**)
    a. Each **genotype** is formed by some **genes**, that could be natural numbers, floating point numbers, plans, games, ecc.
3. A way to sort the population
4. A way to create a new population based on the best elements of the previous one:
    a. Genetic operators are typically used:
        i. Crossover
        ii. Mutations
        iii. Proportional selection
        iv. Genetic substitution

The algorithm itself works as it follows:

1. It generates a random population
2. It evaluates the population
3. It applies genetic operator on the current generation to create a newer one, choosing the best element
4. It repeats from the step 2 with the new generation
5. When a good solution is found (using some criteria), or when enough generation have been evaluated, the algorithm stops

Ant colony Algorithm

Honey-bee Colony

Particle swarm Optimization

2. Local search metaheuristic algorithms

Simulated annealing

Tabu search

Iterated local search

8) 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 a property of constraint satisfaction problems (CSPs). A variable `Xi` is arc-consistent with respect to another variable `Xj` if and only if for every value `vi` in the domain of `Xi`, there is a value `vj` in the domain of `Xj` such that `(vi, vj)` satisfies the constraint between `Xi` and `Xj`. In other words, for each value in one variable, there exists a value in the other variable such that the constraint between the variables is satisfied[14].

Properties of Values Removed and Left in Domains

- Removed Values: Arc consistency eliminates values of each variable domain that can never satisfy a particular constraint (an arc). If a value doesn't have a support (i.e., it's not allowed by the constraint), then we know it's impossible for the variable to have that value, as it would violate the constraint[9]. When a value is removed from the domain of a variable, all arcs of constraints pointing to that pruned variable (except the arc of the current constraint) are added to the collection to be checked[6].

- Left Values: After enforcing arc consistency, there can be three possible outcomes:

1. One domain is empty → no solution.

2. Each domain has a single value → unique solution.

3. Some domains have more than one value → zero or more solutions. In this case, arc consistency isn't enough to solve the problem, and further search is needed.

9) What is conditional planning and what are its main limitations?

A conditional planner is a search algorithm that generates various alternative plans for each source of uncertainty of the plan.

LIMITATIONS:

- Combinatorial explosion of the search tree with high numbers of alternative contexts.
- A comprehensive plan which takes account every possible contingency might require a lot of memory.
- Not always all alternative contexts are known in advance
- Often conditional planners are associated with probabilistic planners that plan only for the most probable contexts.

10) What is ant colony optimization?

Ant Colony Optimization ACO (Dorigo, 1992): Algorithm based on the behaviour of ants. Positive feedback based on pheromone trails. Positive feedback based on pheromon trails that reinforce components that contribute to the problem solution

11) What are the main features of iterative deepening?

- Iterative deepening search avoids the problem of choosing the maximum depth limit by trying all possible depth limits.

  – Start with 0, then 1, then 2 etc ...

- It combines the advantages of depth and breadth-first strategies. It is complete and explores a single branch at a time.
- Many states are expanded multiple times, but this does not worsen considerably the execution time.
- In general it is the favourite search strategy when the search space is very large.
- It can emulate the breadth-first search through repeated applications of the depth first search with an increasing depth limit.

12) What are non-informed search strategies? Describe the strategies that have been presented during the course.

Non informed strategies do not use any domain knowledge and apply rules arbitrarily and do an exhaustive search strategy. They are all special cases of a more general algorithm “general-search” (actually in Russel and Norvig they call it “best-first search” paragraph 3.3.1) which expands nodes according to an evaluation function f(n) in increasing order.

A) BREADTH-FIRST SEARCH

- always EXPANDS LESS DEEP tree nodes.
- It uses as evaluation function f(n) the depth so first closest and then deepest nodes.

- It is complete (by design)
- The main disadvantage is the excessive memory footprint which is exponentially growing as O(b^d) with “b” being the branching factor and “d” the depth.
- always finds the min-cost path if the cost equals the depth (otherwise we should use another strategy that always expands the least cost node -> uniform cost strategy).

B) DEPTH FIRST SEARCH

- Depth-first EXPANDS deepest nodes first.
- It uses as evaluation function f(n) the negative of the depth so first deepest and then closest nodes.
- requires a modest memory occupation but not exponentially in the depth so O(b*d)
- It is not complete as it might get stuck in infinite loop.
- It is EFFICIENT from an implementation point of view.
-

C) LIMITED DEPTH SEARCH

- It is a depth-first variant
- It includes a MAXIMUM DEPTH parameter
- When you reach the maximum depth or a failure, it explores alternative paths (if they exist).
- it does not necessarily solve the problem of completeness
- Avoids infinite branches

D) ITERATIVE DEEPENING

- Iterative deepening search avoids the problem of choosing the maximum depth limit by trying all possible depth limits.

  – Start with 0, then 1, then 2 etc ...

- It combines the advantages of depth and breadth-first strategies. It is complete and explores a single branch at a time.
- Many states are expanded multiple times, but this does not worsen considerably the execution time.
- In general it is the favorite search strategy when the search space is very large.
- It can emulate the breadth-first search through repeated applications of the depth first search with an increasing depth limit.

E) UNIFORM COST STRATEGY

- Takes into account that path may have different cost
- It uses as evaluation function f(n) the path cost g(n)
- It explores first less expensive paths before moving to most expensive ones.
- It is still a BFS but it uses g(n) and not the depth “d” so indeed (since it is a BFS) it is complete, optimal and shares same computational costs of BFS so exponentially growing in both time and space.