MASSACHUSETTS INSTITUTE OF TECHNOLOGY
16.413/6.877 – PRINCIPLES OF AUTONOMY AND DECISION MAKING (GRADUATE VERSION)
FALL 2021

FINAL PROJECT

INTRODUCTION

In this final project, you will implement the techniques you have learned from 16.413/6.877, and apply them to a planning problem. Students will be allocated into groups of sizes 1-5 members. Each group focuses on an advanced topic from the list below. There can be multiple different groups working on the same topic.

- Planning (heuristic forward search AND sample-based motion planning)
- MDPs & HMMs
- Uncontrollable and Multi-agent Systems (Alpha-Beta Search, probabilistic planning)
- Constraint programming (constraint propagation, search & elimination; global constraints; MVRP)
- Logic and Model-based Reasoning (satisfiability & conflict-directed search; diagnosis)
- Constraint Applications: Bayes Nets
- Constraint Applications: Temporal Scheduling
- Mathematical Programming (LPs, simplex, MILPs, branch & bound; motion planning)

The goal of the final project is to gain insight into how each technique can be applied and beneficial for the real-world problem. Although guidelines will be provided in this document, this project is **flexible**, in that you will use your creativity to decide innovative ways to meet the capability, using what you have learned during the course.

GUIDELINES

For the final project, you need to demonstrate your ability to understand, extend, and implement the techniques covered in the course by implementing a planner which can provide a given capability. For this, we provide sample submissions on a variety of topics from previous years as reference. **You should present a detailed problem formulation in your chosen context and provide a modularized explanation, implementation, and demonstration of the code modules all in a Jupyter notebook**. Your final project should explore the topic of your interest instead of replicating the given examples.

IMPLEMENTATION

Please create a step-by-step, pedagogical implementation of the methods of your choice. Note that some of the topics can become too complex to implement and can go beyond a course project. As such, you are free to implement a reasonable subset of the methods, so long as you have a clear way to explain and test everything. You are also encouraged to use third-party libraries or implementations that support your implementation. Focus on implementing what will help explain and demonstrate the method. Please discuss any questions you have with the course staff. All of the implementations must be in Python.

* Dates are tentative and may be adjusted based on the course progress and other constraints.

SCHEDULE AND DELIVERABLES*

10/01/2021 Announcement of Topics and Guidelines
Course TAs send out topic assignments and schedules for all graduate students.

11/12/2021 Planning and Research
The goal of the first deliverable is to have completed the required reading on the topic area, as specified in the lecture notes. The group should provide a skeleton Jupyter notebook, outlining the main ideas of each topic in markup cells, as well as commented code cells which include function stubs for the capabilities to be demonstrated.
Deliverable: Skeleton Jupyter notebook

11/22/2021 Initial Draft Submissions
By Friday at 4pm, the group will submit a first draft of their submission, including the walkthroughs of the ideas involved in each topic, and working code complementing each section. This draft will be reviewed by your classmates from other groups and the course TAs.
Deliverable: Draft of Jupyter notebook:

1. Implementation of main functions
2. Explanations about the functions

11/24/2021 Initial Peer Review
Initial peer review of the assigned initial drafts from other groups needs to be submitted by 4pm.

11/30/2021 - Final Submission
A final draft incorporating feedback from the TAs is to be turned in by Friday at 4pm.
Deliverable: Final Jupyter notebook

12/03/2021 Final Peer Review
Final peer review of the assigned final drafts from other groups needs to be submitted by 4pm.

GRADING

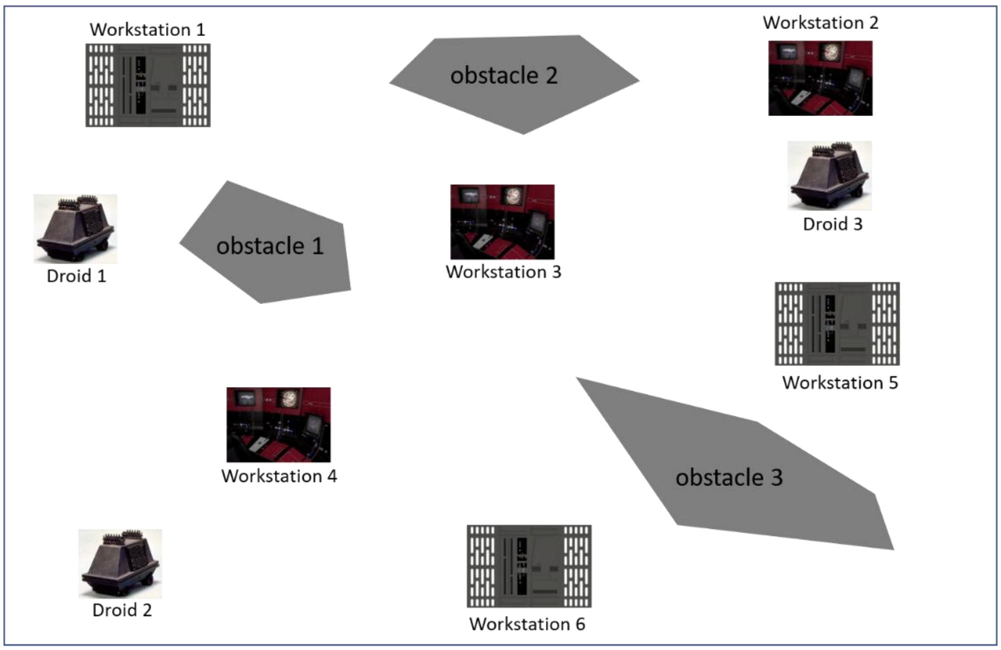
Your final project will be evaluated on the clarity in writing, clearness of the core materials, and the cleanness of modularized code implementation and demonstration. Your initial and final drafts will be peer-reviewed by other students from other groups as well as the TAs. For students who work as a group, you need to clearly indicate the work division and contribution of each group member.
Please always consult the course staffs whenever you have questions and concerns. Do not wait until the last minute!

* Dates are tentative and may be adjusted based on the course progress and other constraints.

Sample Problem Descriptions and Submissions

We provide several sample submissions from previous years. These submissions are based on the sample problems listed below. You may choose to work on one of these problems in your final project or devices your own problems. The relevant implementations/submissions can be found on Canvas.

Scenario



YOU are an engineer hired by the Generic Space Empire for a project. You have been assigned to a ~~moon~~ space station, the center of controversy throughout the galaxy.

Maintenance personnel have noticed that several work stations have been acting up, but have assured the engineering team that "uh, everything's perfectly all right now". You, however, suspect something sinister afoot. Could it be sabotage from the Underdog Rebels?

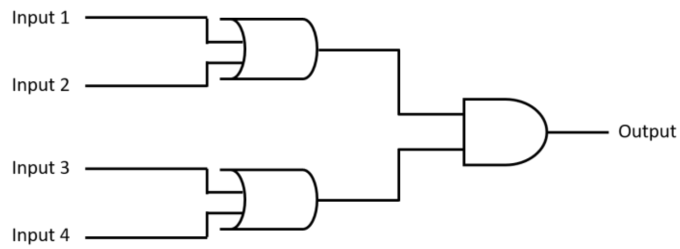
You know that a document for the space station was recently stolen (many Non-Copyrighted Alien spies died), in which several attacks taking advantage of design flaws have been documented. Each attack requires compromising a combination of components, distributed across the workstations.

While you cannot dismantle the workstations to directly inspect the components, you can send a team of diagnosis droids to make observations on the workstations to exactly determine the faulty components. As you perform diagnosis on each workstation, you can update the odds on which type of attack will occur. You must do this in a timely manner, so that you can escape before an attack if necessary, and warn the Big Bad Guy Wearing Black if time permits.

* Dates are tentative and may be adjusted based on the course progress and other constraints.

TASKS AND REQUIREMENTS

Task 1: Component Diagnosis
In this task, you need to perform diagnosis for a workstation to determine which components are faulty. You are given a series of diagnosis test results, including sets of corresponding inputs and outputs, as well as a logical model of the system. You will be asked to discover the list of possible faults given the observations. A simple example of the logical model is given below.

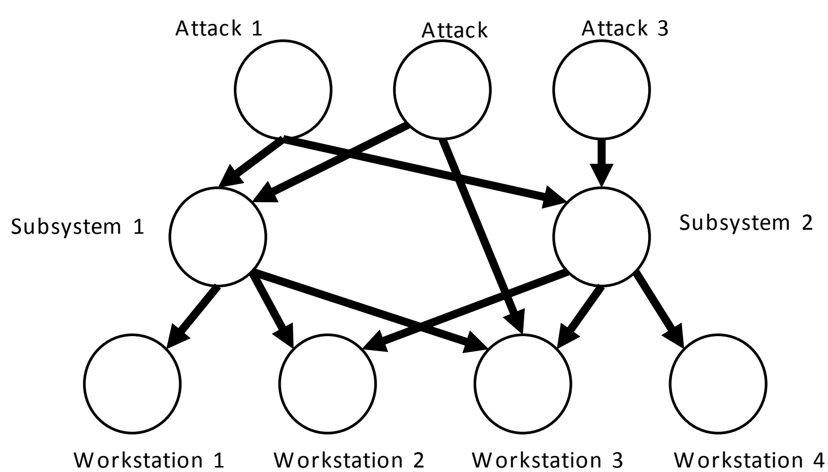


Example Model for Workstation 1

Group 1: Achieving task 1 using mode estimation
Inputs:
1) System model
2) Symptoms

Outputs:
1) Possible faults

Task 2: Attack Estimation
In this task, you are asked to estimate the probability of each attack, given reports of component failures at each workstation. Inference will be performed using a Bayes Net representing relating the probability of each component failing, conditioned on the anticipated attack. You will then be asked to infer the probability of each attack. A simple example of the Bayes Net is given below.



Group 2: Achieving task 2 using Bayesian network
Inputs:
1) Conditional Probability Distribution
2) Observation

* Dates are tentative and may be adjusted based on the course progress and other constraints.

Outputs:
1) Posterior Distribution
2) Most probable configuration (MAP configuration)

Task 3: Multiple Vehicle Routing
In this task, you will assign the order in which workstations are visited by your fleet of droids. You will be given the locations of a set of workstations which must be visited, and your current location. For each droid, you must assign the order in which they visit a subset of the workstations before returning to your location. You will be asked to find the minimum time plan, such that each workstation is visited by at least one droid.

Group 3: Achieving task 3 using constraint optimization
Inputs:
1) Location of depot
2) Goal locations
3) Travel times between locations

Outputs:
1) Sequence of location(s) for each agent

Task 4: Motion planning
In this task, you will be asked to provide trajectories for each droid, in between workstations. You will be given a qualitative state plan with sequences of workstations each robot must visit, as well as a more detailed description of the layout of the ~~moon~~ space station, including obstacles. You will be asked to output a set of waypoints and times of arrival at each waypoint.

Group 4: Achieving task 4 using search and mathematical programming
Inputs:
1) Initial location
2) Goal locations
3) Geometry of factory and obstacles
4) Vehicle speed limits

Outputs:
1) Waypoints
2) Times of arrival at each waypoint

Task 5: Temporal Planning and Reasoning
In this task, you will be asked to check whether you have time to warn the Big Bad Guy Wearing Black, or whether you should just try to escape. You will be given duration information for traversals, diagnosis, and possible additional actions you might take, as well as timing constraints you must meet in order to escape on a shuttle or provide warnings. You will be asked to check for the feasibility of alternative plans, providing a schedule for a feasible plan, and provide an explanation for each infeasible plan.

Group 5: Achieving task 5 using STNs
Inputs:
1) Set of possible actions
2) Set of events
3) Temporal constraints

Outputs:
1) Feasibility?
2) Conflict if infeasible
3) Schedule if feasible

* Dates are tentative and may be adjusted based on the course progress and other constraints.

4) Best temporally feasible plan

QUESTIONS?

If you have any questions, please be sure to ask them on Piazza, preferably earlier than later.

* Dates are tentative and may be adjusted based on the course progress and other constraints.