COMP 307 — *Introduction to AI*

Assignment 4: Planning and Scheduling

8% of Final Mark — Due: 23:59 Friday 8 June 2018

1 Objectives

The goal of this assignment is to help you understand the basic concepts of planning and scheduling, and simple algorithms for them. In particular, the following topics should be reviewed:

- Use Planning Domain Definition Language (PDDL) to represent a classical planning problem,
- Use state-space search algorithms to find a plan,
- Represent a schedule of job shop scheduling problem,
- Generate a schedule by a dispatching rule for a given job shop scheduling problem,
- Find a solution for a vehicle routing problem using the nearest-neighbor heuristic.

All the questions in this assignment can be answered without programming. You can simply write the answers to the questions in your report.

These topics are (to be) covered in lectures 19–22. The online materials can also be checked.

2 Question Description

Part 1: Classical Planning – Monkey-and-Bananas [35 marks]

In this part, you are required to represent the monkey-and-bananas problem using PDDL, and find a plan by forward state-space search.

Problem Description

In the monkey-and-bananas problem, a monkey, a box and some bananas are placed in a room. The bananas are hung from the ceiling and the monkey cannot reach them directly. The only way for the monkey to catch the bananas is to move the box under the bananas and climb up onto the box.

Assume that initially, the monkey is at location A, box at location B and bananas at location C. The monkey and box have height *Low*, i.e. they are on the ground. The bananas have height *High*. If the monkey climbs onto (down) the box, its height will be *High* (*Low*). The monkey cannot catch the bananas when its height is *Low*. The actions that the monkey can take include (1) *Go* from one place to another, (2) *Push* the box from one place to another (assume this action requires that the box and the monkey are at the same current location, and as a result the box and the monkey are at the same new location), (3) *ClimbUp* onto or *ClimbDown* from the box (the box and the monkey are at the same location before and after these actions), and (4) *Grasp* and *Ungrasp* the bananas. The result of a *Grasp* is that the monkey holds the bananas, if the monkey and the bananas are in the same place at the same height. The goal state is that the monkey holds a banana.

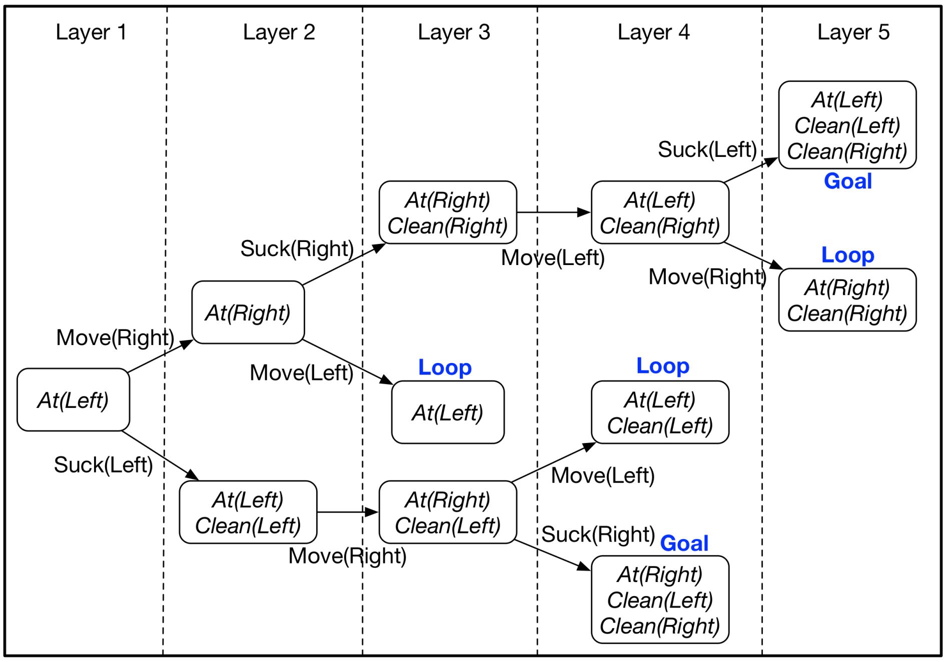
Questions

1. (5 marks) Write the description for the initial and goal states using PDDL.
2. (10 marks) Write all the action PDDL schemas (there are 6 actions in total). Each action should include a *name*, *variables*, *precondition* and *effect*. The precondition of each action should be set in a way that self-connection is avoided, i.e. no action connects any state to itself.

3. (15 marks) A plan to achieve the goal state from the initial state can be found by using forward state-space search. Based on the PDDL in Questions 1 and 2, draw the **first three layers** of the corresponding state-space search graph to demonstrate the search process.

   - Each node (state) in the graph is represented by a conjunction of fluents.
   - Each edge is associated with an action.
   - Each leaf node is either a goal state or a loop.

   Example: The figure below is the 5-layer forward state-space search graph of the vacuum cleaner's world (the $\wedge$ *between fluents are omitted).*



   If you find it hard to draw the graph using drawing software, you can simply draw the graph by hand and scan it as an electronic copy.

4. (5 marks) Write a plan to achieve the goal state from the initial state. The plan needs to be formatted as follows:

   - Initial state:
   - Action 1:
   - State 1:
   - Action 2:
   - ...
   - State k (goal state):

Part 2: Job Shop Scheduling [40 marks]

In this part, you are required to find solutions (schedules) for a job shop scheduling problem.

Problem Description

The table below gives a job shop schedule problem with 3 jobs and 2 machines.

Job	ArrivalTime	Operation	Machine	ProcTime
J_1	0	O_{11}	M_1	50
		O_{12}	M_2	25
J_2	10	O_{21}	M_2	30
		O_{22}	M_1	35
J_3	20	O_{31}	M_1	40
		O_{32}	M_2	20

- (Number of operations) Each job J_j has two operations O_{j1} and O_{j2}.
- (Order constraint) The operations strictly follow the order constraint. That is, O_{j2} $(j = 1, 2, 3)$ cannot be processed until O_{j1} has been completely processed.
- (Arrival time) Each job has an arrival time (ArrivalTime). For each job J_j, the first operation O_{j1} cannot be processed earlier than its arrival time.
- (Resource constraint) Each operation can only be process by a particular machine. For example, operation O_{11} can only be processed by machine M_1. Each machine can process at most one operation at a time.

Solution/Schedule Representation

A solution/schedule for a job shop scheduling problem is a sequence of actions. Each action is composed of the processed operation, the machine to process the operation, and the starting time. The finishing time of an action is the starting time plus the processing time of the processed operation. The actions are sorted in the increasing order of starting time, i.e. the former action starts no later than the latter one. In this assignment, the following format is adopted to represent a schedule:

$$Process(O_{11}, M_1, 0) \rightarrow Process(O_{21}, M_2, 10) \rightarrow \dots,$$

where $Process(o, m, t)$ stands for an action that processes the operation o with machine m and starts at time t.

Questions

1. (10 marks) Given a schedule whose action sequence is as follows: $Process(O_{11}, M_1, t_1) \rightarrow Process(O_{21}, M_2, t_2) \rightarrow Process(O_{31}, M_1, t_3) \rightarrow Process(O_{12}, M_2, t_4) \rightarrow Process(O_{22}, M_1, t_5) \rightarrow Process(O_{32}, M_2, t_6)$. Since the sequence is sorted in the increasing order of starting time, we know that $t_1 \leq t_2 \leq t_3 \leq t_4 \leq t_5 \leq t_6$. Calculate the **earliest starting time** (t_1 to t_6) of each action. You can draw gantt chart to help you think.

   Hint: the earliest starting time of an action is the later time between the earliest ready time of the operation and the earliest idle time of the machine.

2. (5 marks) For the solution given in Question 1, find the **completion time** of each job, which is the finishing time of its last operation. Then, calculate the **makespan** of the solution, which is defined as the maximum completion time of all the jobs.

3. (15 marks) Write the state from **step 1 to step 3**, and the **final solution** when applying the Shortest Processing time (SPT) dispatching rule to the problem. At each step, the representation of a state is composed of (1) a **partial solution**, (2) the **earliest idle time** of each machine and (3) the **earliest ready time** of each unprocessed operation. The initial state (step 0) is given below for your reference.

> **Step 0:**
>
> - Partial solution: (empty, no action is scheduled)
> - $earliestIdleTime(M_1) = 0$, $earliestIdleTime(M_2) = 0$
> - $earliestReadyTime(O_{11}) = 0$, $earliestReadyTime(O_{12}) = \infty$
> - $earliestReadyTime(O_{21}) = 10$, $earliestReadyTime(O_{22}) = \infty$
> - $earliestReadyTime(O_{31}) = 20$, $earliestReadyTime(O_{32}) = \infty$

4. (5 marks) For the solution obtained by the SPT rule, calculate the completion time of each job and the makespan. Compare the makespan between this solution with that obtained in Question 1 to find out which solution is better in makespan.

   Note: the solution in Question 1 is obtained by the First-Come-First-Serve (FCFS) rule.

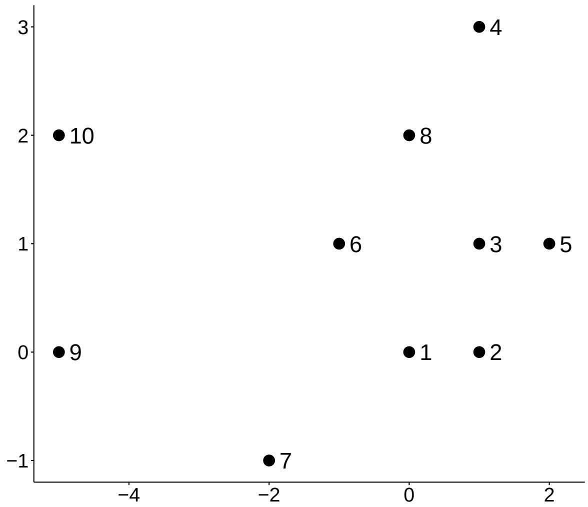
5. (5 marks) The two compared solutions are obtained by the SPT and FCFS rules, respectively. If one solution is better then the other, does it mean that the rule that generates the better solution is better than the other rule? Why?

Part 3: Vehicle Routing [25 marks]

In this part, you are required to find a solution using the nearest neighbor heuristic and calculate its total cost for the given vehicle routing problem.

Problem Description

The graph below gives a vehicle routing problem.



The location and demand of each node is given as follows. Node 1 is the depot. Each node except the depot has a demand of 1. The capacity is 3.

Node	x-coordination	y-coordination	Demand
1 (depot)	0	0	0
2	1	0	1
3	1	1	1
4	1	3	1
5	2	1	1
6	-1	1	1
7	-2	-1	1
8	0	2	1
9	-5	0	1
10	-5	2	1

The cost of each edge is the Euclidean distance between the two end-nodes. The Euclidean distance matrix is given below.

$$D = \begin{array}{cc} & \begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array} & \begin{pmatrix} 0 & 1.00 & 1.41 & 3.16 & 2.24 & 1.41 & 2.23 & 2.00 & 5.00 & 5.39 \\ 1.00 & 0 & 1.00 & 3.00 & 1.41 & 2.24 & 3.16 & 2.24 & 6.00 & 6.32 \\ 1.41 & 1.00 & 0 & 2.00 & 1.00 & 2.00 & 3.61 & 1.41 & 6.08 & 6.08 \\ 3.16 & 3.00 & 2.00 & 0 & 2.24 & 2.83 & 5.00 & 1.41 & 6.71 & 6.08 \\ 2.24 & 1.41 & 1.00 & 2.24 & 0 & 3.00 & 4.47 & 2.24 & 7.07 & 7.07 \\ 1.41 & 2.24 & 2.00 & 2.83 & 3.00 & 0 & 2.24 & 1.41 & 4.12 & 4.12 \\ 2.23 & 3.16 & 3.61 & 5.00 & 4.47 & 2.24 & 0 & 3.61 & 3.16 & 4.24 \\ 2.00 & 2.24 & 1.41 & 1.41 & 2.24 & 1.41 & 3.61 & 0 & 5.39 & 5.00 \\ 5.00 & 6.00 & 6.08 & 6.71 & 7.07 & 4.12 & 3.16 & 5.39 & 0 & 2.00 \\ 5.39 & 6.32 & 6.08 & 6.08 & 7.07 & 4.12 & 4.24 & 5.00 & 2.00 & 0 \end{pmatrix} \end{array}$$

Questions

1. (15 marks) Find a solution by the nearest neighbor heuristic. Write the solution as a set of node sequences starting and ending at the depot node (node 1). It should look like $R_1 = (1, \ldots, 1)$, $R_2 = (1, \ldots, 1)$, $\ldots$.

2. (10 marks) Calculate the total length (Euclidean distance) for the obtained solution.

3 Notes

During the time between the assignment handout and submission, the tutor(s) will run a number of helpdesks to provide assistance.

4 Submission Guidelines

4.1 Submission Requirements

A document consisting of the answers of all the questions. The document can be written in PDF, text or the DOC format. You need to submit this document in both *soft copy* and *hard copy*.

If you write the answers on the paper, you should scan the paper and submit the scanned PDF as the soft copy.

4.2 Submission Method

The PDF/Text/DOC version of the document should be submitted through web submission system from the COMP307 course web site **by the due time**.

The hard copy of the document is required to submit to the COMP 307 handin box in the 2nd floor corridor of the Cotton building by the due time.

4.3 Late Penalties

All assignments must be submitted on time unless you have made a prior arrangement with the lecturer or have a valid medical excuse (for minor illnesses it is sufficient to discuss this with the lecturer.) The penalty for assignments that are handed in late without prior arrangement is one grade reduction per day. Assignments that are more than one week late will not be marked.