Problem Set 1

Mark Xavier (xaviem01)

February 5, 2019

1. Problem 1

Task	T1	T2	T3	T4
Length	12	42	48	54

Processor	P1	P2
Speed	2	3

(a) Characterize tree structured state space problem.

- The **states** are characterized as as either the empty state (no tasks assigned to the any processors) or a partially filled state (some n tasks assigned to m processors). Technically the goal state (the state where all tasks are assigned some processor and the total time taken is less than deadline time D) is also a state.
- The **operators** (or operations) are defined as actions that assign a given task to a processor or remove a task from a processor. The goal is to maintain the inequality set up in the problem.
  In other words, assuming p_i to be some processor in the set of processors P, and t_j to be some task in the set of tasks T, and defining $t_j \in p_i$ as task t_j being assigned to p_i, the operation of assigning $t_j \in p_i$ is only allowed if the following inequality is maintained:

  $$\forall p \in P, \left(\sum_{t \in p} t.length/p.speed \right) < D$$

  Where D is the deadline time. If an unallowable assignment is made, breaking the above inequality, then a task must be removed from the processor in order to continue.
- The **branching factor** is 2, since at each step we assign a given task to one of the two processors. In general, the branching factor is the number of processors.
- The **depth of the goal node** is known initially in some cases where a solution exists. If a solution exists, then all tasks are assigned a processor, and therefore the depth of the goal node is 4, since there are 4 edges from the goal node to the root node. Of course if no solution exists, then the goal node does not exist and it's depth is undefined.

(b) State space with depth-first search:

```
0 --- T1 to P1 (6,0) --- T2 to P1 (27,0) --- T3 to P1 (51 - Fail)
                      |                   |
                      |                   |-- T3 to P2 (27,16) --- T4 to P1 (54,16 - Fail)
                      |                                         |
                      |                                         |-- T4 to P2 (27,34 - Fail)
                      |
                      |-- T2 to P2(6,14) --- T3 to P1(30,14) --- T4 to P1 (57,14 - Fail)
                                                              |
                                                              |-- T4 to P2 (30,32 - PASS)
```

(c) State space with breadth-first search:

```
0 --- T1 to P1 (6,0) --- T2 to P1 (27,0) --- T3 to P1 (51,0 - Fail) --- T4 to P1 - Fail
  |                  |                   |                          |
  |                  |                   |                          |-- T4 to P2 - Fail
  |                  |                   |
  |                  |                   |-- T3 to P2 (27,16) --- T4 to P1 (54,16 - Fail)
  |                  |                                        |
  |                  |                                        |-- T4 to P2 (27,34 - Fail)
  |                  |
  |                  |-- T2 to P2 (6,14) --- T3 to P1 (30,14) --- T4 to P1 (57,14 - Fail)
  |                                      |                    |
  |                                      |                    |-- T4 to P2 (30, 32 - PASS)
  |                                      |
  |                                      |-- T3 to P2 (6,30)
  |
  |-- T1 to P2 (0,4) --- T2 to P1 (14,4) --- T3 to P1 (38,4 - Fail)
                     |                    |
                     |                    |-- T3 to P2 (14,20)
                     |
                     |-- T2 to P2 (0,18) --- T3 to P1 (24,18)
                                          |
                                          |-- T3 to P2 (0,34 - Fail)
```

2. Problem 2

   (a) Characterize tree structured state space problem.

   - The **states** are either the empty state (no processors assigned any tasks), or a state that may be partially of completely filled such that some set of processors $\in P$ are assigned tasks $\in t$ so that for any given processor, the time taken is $< D$ the sum of lengths of the assigned $t \in T$ is > 0 (unless there is some case where a task has no length). Presumably if the sum of lengths of the assigned t is $\geq S$ then we are at a goal state.
   - The **operators** are to assign a task to a processor or to remove a task from a processor so that it may be assigned elsewhere or left unassigned. Assignments and re-assignments/unassignments follow from the inequality described above, so that

   $$\left(\forall p \in P, \left(\sum_{t \in p} t.length/p.speed < D\right)\right) \wedge \sum_{t \in P} \left(t.length < S\right)$$

     Where $t \in p$ denotes a task t that is assigned to processor p, $t \in P$ denotes a task t assigned to any processor p in P, and $\wedge$ denotes the propositional logic symbol for "and".
   - The **branching factor** here is 3, since it need to necessarily be the case that all tasks are assigned to processors, then at each step we determine whether we should assign the current task to a processor or not assign it at all. Then in general, the branching factor is the number of processors + 1.
   - In this case, the **depth of the goal node** is not initially known. Assuming that we have a random number of tasks and processers, because not all tasks need be assigned to the processors we have no way of knowing the depth of the goal node. We can perhaps make educated guesses in specific cases by analysing the lengths of the tasks and the value of S and D, but this would only give us an idea of where the goal node **could** be and may not be practical.

   (b) State space generated by depth-first search:

   (c) State space generated by breadth-first search: